IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

A control barrier function approach for maximizing performance while fulfilling to ISO/TS 15066 regulations

Federica Ferraguti[1], Mattia Bertuletti[1], Chiara Talignani Landi[1], Marcello Bonfè[2]
Cesare Fantuzzi[1] and Cristian Secchi[1]

***Abstract*— ISO/TS 15066 is globally recognized as the guideline for designing safe collaborative robotic cells, where human and robot collaborate in order to fulfill a common job. Current approaches for implementing the ISO/TS 15066 guidelines lead to a conservative behavior (e.g. low velocity) of the robot and, consequently, to poor performance of the collaborative cell. In this paper, we propose an approach based on control barrier functions that allows to maximize the performance of a robot acting in a collaborative cell while satisfying the ISO/TS 15066 regulations. The proposed approach has been successfully validated both in simulation and through experiments.**

I. Introduction

In collaborative robotics, humans and robots interact in industrial scenarios without traditional safeguarding, under specific conditions. As a consequence of the spreading of human-robot collaboration technologies in the industry, great importance has been attributed to robot safety standards, which have been updated to address new co-working scenarios [1]. In particular, International ISO 10218-1 and ISO 10218-2 [2], [3] safety standards have identified specific applications and criteria where collaborative operations can occur. More recently, the technical specification ISO/TS 15066 [4] has been introduced to specify safety requirements for collaborative industrial robot systems and the work environment and to supplement the requirements and guidance on collaborative industrial robot operations given in ISO 10218-1 and ISO 10218-2. The guidelines defined in the technical specification ISO/TS 15066 identify four *collaborative operations*, which can be used either individually or in combination, depending on the requirement of the application and the design of the robot system: Safety-rated Monitored Stop (SMS), Speed and Separation Monitoring (SSM), Power and Force Limiting (PFL) and Hand Guiding (HG). In particular, SSM and PFL can be adopted when humans and robots perform their tasks in a shared workspace. In SSM, the speed of the robot is monitored and adjusted based on the position and, possibly, speed of the operator in the safeguarded space, considering as a primary objective the avoidance of any human-robot collision. In particular, the robot must always move at a speed allowing it to reach a complete stop before getting in touch with the operator, so that the distance of the latter is relevant in the adaptive computation of the safe robot speed (i.e. depending on the deceleration capability of the robot). On the other side, in PFL the physical contact between a robot and an operator is allowed and the necessary safety of humans is achieved by limiting the power and force to values at which the risk of injuries is not expected. To this aim, the technical specification ISO/TS 15066 describes biomechanical load maximum values, which must not be exceeded if the robot collide with body parts, and how to compute the robot speed limits guaranteeing safe human-robot contacts. In common scenarios, SSM is usually adopted, while PFL is used in tight collaborative operations setting a low collaborative robot speed. In the first case, the result is a frequently stopped robot, especially when dealing with small shared spaces, since collisions are not allowed. In the second case, instead, the result is a conservative system, where the worst collision case is always considered and the robot is moved very slowly.

The objective of this work is to provide a control method overcoming the limitations of both SSM and PFL operations, by embedding in an optimization-based controller the requirements of ISO/TS 15066 on safe human-robot contacts and by allowing the robot to move at higher speeds when the risk of such contacts is reduced, due to the human-robot separation distance. This objective will be obtained exploiting the theoretical framework of Control Barrier Functions [5], that provides a formal way to constrain the behavior of a controlled system into any properly specified safe set.

Related works on industrial human-robot collaborations have already addressed the definition of metrics and control methods based on separation distance and robot velocity with respect to a human operator [6]. In particular, [7] describes a trajectory generation algorithm for safe human-robot collaboration based on the solution of an optimization problem and the calculation of human predicted occupancy, a volume in which the human is expected to move within a given time horizon. However, the primary objective of these works is the avoidance of any human-robot collisions. The ISO/TS 15066 PFL requirements are instead explicitly addressed for robot motion planning or, better, re-planning in [8], however the properties of the re-planning method are not formally proved, and PFL requirements are also addressed in [9], whose aim is to define a software architecture that can abruptly switch (on demand) from SSM to PFL, rather than overcoming the limits of both modes. Finally, an analysis tool is presented in [10], whose objective is to analyze the robot dynamics from a task-dependent point of view, to evaluate the safety performance of the robot design. However, the

[1]Department of Science and Methods for Engineering, University of Modena and Reggio Emilia, Italy. E-mail: {federica.ferraguti, mattia.bertuletti, chiara.talignanilandi, cristian.secchi, cesare.fantuzzi}@unimore.it
[2]Engineering Department, University of Ferrara, Italy. E-mail: marcello.bonfe@unife.it

proposed tool does not suggest any methodology for robot control design.

The contribution of this paper, instead, is the definition of a control design framework based on the concepts of Zeroing Control Barrier Functions (ZCBFs, [5]), to enforce ISO/TS 15066 PFL velocity bounds, while obtaining faster operations when human-robot distance allows to relax the conservative constraint of the ISO technical specification. To the best of our knowledge, our proposal is the first application of ZCBFs as a design tool for safe human-robot collaboration compatibly with the requirements of ISO/TS 15066. In particular, the contributions of this paper are:

- the real-time computation of the safe set of the system on the basis of the current distance and relative velocity between robot and human, embedding the ISO/TS 15066 PFL concepts in a non conservative way;
- the application of ZCBF-based optimization as a safety filter that can constrain the behavior of the robotic system in the mentioned safe set, whatever is the nominal and potentially unsafe controller adopted for robot trajectory tracking;
- the practical validation of the ZCBF-based control framework on an experimental setup, including a Universal Robot UR5 manipulator and depth camera system for human body tracking.

In Section II we will first specify the relevant ISO/TS 15066 PFL provisions, then in Sections III and IV we will present the proposed control approach to address them. Finally, Section V will describe the practical validation and the results.

II. Power and Force Limiting According to ISO/TS 15066

In the technical specification ISO/TS 15066, the Power and Force Limiting (PFL) collaborative operation allows a physical contact between the robot system and an operator, either intentionally or unintentionally, under certain circumstances. In particular, the ISO/TS 15066 provides a guidance on how to assess safety of collaborative robots in PFL operations, suggesting calculations based on force limits on the basis of the pain sensitivity thresholds when contacts between the human and the robots occur. The maximum permissible force is based on the lowest energy transfer criteria that could result in a minor injury, such as a bruise, equivalent to a severity of 1 on the Abbreviated Injury Scale (AIS) established by the Association for the Advancement of Automotive Medicine[1]. Adherence to the limits will prevent the occurrence of skin or soft tissue penetrations that are accompanied by bloody wounds, fractures or other skeletal damages. The force limits are used, in the Annex A of the ISO/TS 15066, to establish the related bounds that have to be set on the robot velocity for movements inside the collaborative workspace, depending on the transfer of energy that results from an hypothetical human-robot contact. In particular, the maximum permissible energy transfer can be calculated as a function of the maximum contact force value, assuming that the human body absorbs the energy of an impact as a linear spring, as follows [4]:

$$E_{max} = \frac{{F_{max}}^2}{2k} \tag{1}$$

where F_{max} and k are the maximum contact force and the stiffness constant of the body region interested by the hypothetical contact with the robot. Indeed, both the force limits and the stiffness constants vary depending on the different body areas that are involved and their values are detailed and tabulated in the Annex A of the ISO/TS 15066, for each body region.

Once the energy transfer limit value for the contact scenario is computed according to (1), it can be used to identify the maximum velocity at which the robot would be able to move through the collaborative workspace, while maintaining the force value below the threshold limit. Let us define v_{rel} as the relative velocity between the robot and the interested human body region. It is important to remark that in the ISO/TS 15066 and, therefore, in the rest of the paper, by focusing on said relative velocity the analysis of robot behavior is reduced to a single (translational) degree of freedom, namely the local axis connecting the human body part of interest to the closest part of the robot. In order to derive a relationship between v_{rel} and the contact force F, the ISO/TS 15066 assumes that, in case of fully inelastic contact, the total kinetic energy involved in the hypothetical impact is transferred to the human body area according to its stiffness. Thus, it results the following energy balance:

$$E = \frac{F^2}{2k} = \frac{1}{2}\mu {v_{rel}}^2 \tag{2}$$

where μ is the reduced mass of the two-body system (i.e. the system composed by the robot and the human body area), that can be computed as:

$$\mu = \left(m_h^{-1} + m_r^{-1}\right)^{-1} \tag{3}$$

being m_h the effective mass of the human body area and m_r the effective mass of the robot. To compute the latter, the ISO/TS 15066 suggests the following simplified formula:

$$m_r = (M/2) + m_l \tag{4}$$

where M is the total mass of the moving parts of the robot and m_l is the effective payload of the robot system, including the tool and workpiece. Equations (2) and (3) are obtained under the assumption that at the impact event the two bodies of human and robot colliding parts are coupled, from which the formulation of the reduced mass μ is derived, and the total kinetic energy is completely transformed into the energy related to the deformation of human body tissue, modeled as a pure spring with stiffness k. Thus, from (1) and (2) it results that human safety in PFL operations enforces a constraint on the relative velocity between the robot and a given human body part. Recalling that such a relative velocity is positive when the separation distance is increasing (i.e. there is no risk of collision) and negative when the distance

[1]https://www.aaam.org/abbreviated-injury-scale-ais/

is decreasing, the constraint applies only in the latter case and can be stated as follows:

$$|v_{rel}| \leq v_{PFL} = \frac{F_{max}}{\sqrt{\mu k}} \tag{5}$$

where we have denoted with the term v_{PFL} the maximum permissible value on v_{rel} in PFL operations, according to the ISO/TS 15066.

III. Definition of an Adaptive Safe Operating Region

The objective of the proposed control system is enforce the safety constraints on human-robot PFL operations by guaranteeing that the robot behavior, described in terms of distance and relative velocity with respect to a selected human body area, is confined into an adaptive safe operating region. In particular, we aim to reduce the robot velocity along the direction of motion pointing to the human body, depending on the distance between the robot itself and such a human body, until the limit of v_{PFL} is reached and maintained. However, we do not want to enforce v_{PFL} all the time. Indeed, if the human-robot distance is sufficient to apply a deceleration along the robot to human direction and reach a relative velocity lower than v_{PFL} before the human-robot contact is really imminent, than the system would still be safe. In the following, we will first describe how to map these concepts into the definition of an adaptive safe operating region, while in the following section we will show how to apply geometrical considerations to formalize a Zeroing Control Barrier Function and a related optimization problem for the computation of safe robot velocities.

A. Definition of the safe operating region

As suggested by the ISO/TS 15066, the safety constraints on PFL can be analyzed considering a single translational degree of freedom: the distance between two points of interest on the human body and on the robot structure, respectively, and the relative velocity between the same points. Therefore, we will define the safe operating region in terms of the relationship between these two variables, that we will denote with d and v_{rel} respectively.

To embed the concepts of the ISO/TS 15066, we start by considering that when the human is far from the robot or human and robot are moving away from each other (i.e. the relative velocity is positive), the robot can safely move at a high velocity. Hence, the upper bound of the relative velocity can be defined as the maximum robot velocity in the Cartesian space, that we denote as v_{max}. This bound depends on the capabilities of the mechanical structure. When the human approaches and the relative velocity is negative, instead, the robot velocity has to be modified, so that its component along the direction of motion pointing towards the human satisfies the safety bound of (5) (i.e. v_{PFL}) before a contact could happen. This means that we have to analyze the deceleration capability of the robot, again in the direction of interest for the computation of the relative velocity. In general, assuming that the robot can constantly apply a maximum acceleration/deceleration of a_{max} in that direction of motion, the distance required to reduce the relative velocity from its current value to a given v^* (satisfying $0 \leq v^* < v_{PFL}$) is:

$$d = \frac{1}{2}\frac{(v_rel - v^*)^2}{a_{max}} \tag{6}$$

It can be noted that the previous equation also describes a quadratic curve on a plane with d on the abscissa and v_{rel} on the ordinate. With this in mind, we can consider that a safety-related deceleration aiming to reduce the absolute value of the relative velocity from v_{max} to v^*, that is a transition on the d/v_{rel} plane along the curve defined by (6), could end when a value of v_{PFL} is reached. Therefore, if we draw on the d/v_{rel} plane the quadratic curve (6), an horizontal line described by a constant v_{PFL} at any distance d and an horizontal line at constant v_{max}, we obtain a pictorial view of the safe operating region for ISO/TS 15066 compatible PFL operations. Such a view, focused on negative values of the relative velocity, which are those of interest for human safety, is given in Fig. 1 for a generic and abstract case in which $v^* = 0$, the value chosen for simulations and experiments in this paper. Higher values of v^* would enlarge the safe set, but also reduce robustness against the inherent uncertainties of practical applications. In the figure, the overall safe area is highlighted in yellow. The robot motion can result in a velocity towards the human up to v_{max} (red line) when the distance is large. If the distance becomes smaller than the one at which the curve at constant maximum deceleration (blue line) intersects the line at constant v_{max}, the upper bound on the relative velocity must be enforced by the blue line itself. Finally, if the distance is such that the v_{PFL} can be enforced without accelerating away from the human, the bound of (5) (i.e. the magenta line on the d/v_{rel} plane) is actually applied on v_{rel}.

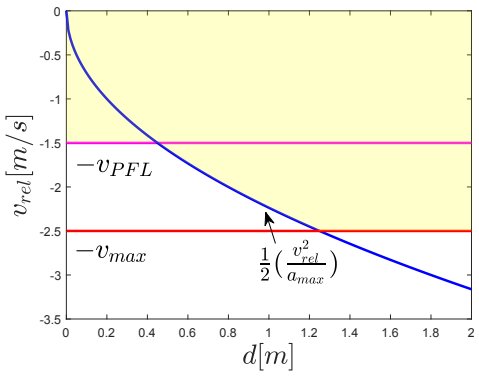


Fig. 1. Definition of the safe operating region (yellow area). If the robot moves with a velocity that guarantees that the relative velocity with respect to the human lies inside the yellow area, the trajectory is safe.

B. Human-Robot distance and relative velocity evaluation

To precisely compute distance and relative velocity between a human and a robot, we will consider in the following the case of point-like objects, that is sufficient to describe the

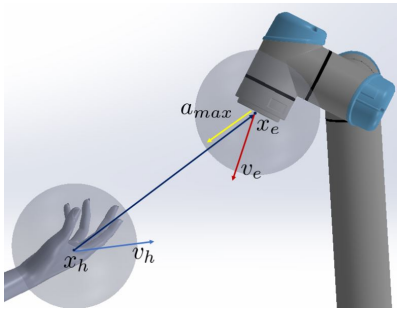


Fig. 2. End-effector and human body part positions and velocities.

collisions between the robot end-effector and a specific part of the human body. Therefore, we will refer to Fig. 2 as the case of study and we will denote with $x_h \in \mathbb{R}^3$ the human body part position and with $x_e \in \mathbb{R}^3$ the position of the robot end-effector. The following computations could be extended, to achieve whole-body safety, to all the pairs composed by a robot link and a human link as proposed in [11].

Hence, we define the distance between the end-effector position and the human hand position and as:

$$d = \|x_e - x_h\| \tag{7}$$

and the relative velocity as:

$$v_{rel} = \dot{d} = \frac{(x_e - x_h)^T (v_e - v_h)}{\|x_e - x_h\|} \tag{8}$$

It is worth to note that, without loss of generality, we could also subtract from (7) the (constant) radii of two spheres, centered respectively in x_e and x_h, to have an additional margin on the safety constraint.

Considering again Fig. 2, we also compute acceleration and velocity limits for the robot with respect to the direction of motion between the human body part and the robot end-effector, which can be defined as the following unit vector:

$$u_s = \frac{(x_e - x_h)}{\|x_e - x_h\|} \tag{9}$$

We recall that the velocity and acceleration in the operational space of a robot with n degrees of freedom depend on its joint velocities and accelerations, $\dot{q} \in \mathbb{R}^n$ and $\ddot{q} \in \mathbb{R}^n$ respectively, as follows:

$$\dot{x}_e = J_t \dot{q} \tag{10}$$

$$\ddot{x}_e = \dot{J}_t \dot{q} + J_t \ddot{q} \tag{11}$$

where $J_t \in \mathbb{R}^{3\times n}$ is the translational part of the robot's Jacobian. Then, we assume that the dynamic capabilities of the robot are expressed in joint space by symmetrical (i.e. minimum values are the opposite of the maximum ones) bounds, that we denote as $\dot{q}_{max} \in \mathbb{R}^n$ and $\ddot{q}_{max} \in \mathbb{R}^n$. As a consequence, robot velocity and acceleration bounds along the direction of u_s, corresponding to the values of v_{max} and a_{max} appearing in Fig. 1, can be computed as follows:

$$v_{max} = |u_s^T J_t| \dot{q}_{max} \tag{12}$$

$$a_{max} = u_s^T \dot{J}_t \dot{q} + |u_s^T J_t| \ddot{q}_{max} \tag{13}$$

where $|\cdot|$ is the element-wise absolute value of a vector.

It is important to remark that all of the previous results depend on the current robot configuration. Therefore, the safe operating region described in Fig. 1 is inherently time-varying and must be computed adaptively in real-time, according to the robot kinematics. Moreover, one should note that the minimum velocity value is actually $-v_{max}$, while the minimum acceleration is obtained by changing the sign of the second term in (13). Theoretically, even the value of a_{max} could result negative. However, this can hardly happen for collaborative robots, since their joint velocities are generally limited. Indeed, we never detected this condition during our experiments.

IV. A ZCBF-BASED FRAMEWORK FOR HUMAN-ROBOT PFL OPERATIONS

To guarantee that robot velocity never exceeds the safety boundaries, we exploit Control Barrier Functions, in particular the Zeroing Control Barrier Functions (ZCBFs). In Section IV-A we will provide the fundamentals of ZCBFs (for more details, the reader is addressed to [12]). Then, in Sections IV-B and IV-C we will exploit the ZCBFs to formalize a framework to enforce ISO/TS 15066 PFL velocity bounds.

A. Background on Zeroing Control Barrier Functions

Let us consider a control affine, nonlinear system as:

$$\dot{\chi} = f(\chi) + g(\chi)u \tag{14}$$

where $\chi \in \mathbb{R}^n$ is the system state and $u \in \mathcal{U} \subset \mathbb{R}^m$ is the control input, with $\mathcal{U}$ the set of admissible control values. Functions f and g are locally Lipschitz continuous. In our previous work [11], we exploited the Reciprocal Barrier Functions (RBF), where the barrier function was defined as a non-negative function that grows to infinity for states that approach to safety constraints, while it decreases moving away from them. However, unbounded functions values may be not desirable in practical applications, to avoid numerical issues. For this reason, in this paper we replaced the RBF with the Zeroing Barrier Functions, where the barrier function vanishes on the set boundary.

Formally, a smooth function $h(\chi)$ is defined in order to find an appropriate Control Barrier Function (CBF) that constraints the system state to lay inside the safe region, i.e. $h(\chi) > 0$, whereas $h(\chi) \leq 0$ indicates a violation of the constraint. Thus, the safety region and its boundaries are defined by the set of admissible states $\mathscr{C}$:

$$\begin{aligned} \mathscr{C} &= \{\chi \in \mathbb{R}^n \mid h(\chi) > 0\}, \quad h : \mathbb{R}^n \to \mathbb{R} \\ \partial\mathscr{C} &= \{\chi \in \mathbb{R}^n \mid h(\chi) = 0\} \end{aligned} \tag{15}$$

Then, the goal becomes to design a proper control law u that allows to guarantee the forward invariance of the set $\mathscr{C}$ for all future times, i.e. if $\chi(0) \in \mathscr{C}$ then $\chi(t) \in \mathscr{C}$ for all $t \geq 0$. This objective can be reached exploiting the formulation of the CBFs and relating the system control input with the function $h(\chi)$, in order to enforce a proper constraint

on the time derivative of the CBF. More precisely, a Zeroing Control Barrier Function is defined as follows.

Definition 4.1: ([5], *Def. 5*)
Given a set $\mathscr{C} \subset \mathbb{R}^n$ defined by (15) for a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$, the function h is called a Zeroing Control Barrier Function defined on set $\mathscr{D}$ with $\mathscr{C} \subseteq \mathscr{D} \in \mathbb{R}^n$, if there exists an extended class $\mathcal{K}$ function [2] α such that

$$\sup_{u \in U} \left[\mathcal{L}_f h(\chi) + \mathcal{L}_g h(\chi) u + \alpha(h(\chi))\right] \geq 0, \quad \forall \chi \in \mathscr{D} \tag{16}$$

The ZCBF h is said to be locally Lipschitz continuous if α and the derivative of h are both locally Lipschitz continuous.

It is worth to note that the previous definition of ZCBF forces:

$$\dot{h}(\chi) \geq -\alpha(h(\chi)) \tag{17}$$

Once we define an appropriate ZCBF, a relationship with the control value u is derived (i.e. provided that $\mathcal{L}_g h(\chi) \neq 0$, i.e. the ZCBF has relative degree of one with respect to the control input, within the admissible set $\mathscr{C}$), yielding to the set of admissible control values $\mathcal{K}_h(\chi)$:

$$\mathcal{K}_h(\chi) = \{u \in \mathcal{U} \mid \mathcal{L}_f h(\chi) + \mathcal{L}_g h(\chi) u - \alpha(h(\chi))\} \geq 0\} \tag{18}$$

By applying a control input $u \in \mathcal{K}_h(\chi)$, we guarantee that the set $\mathscr{C}$ is forward invariant.

Multiple barrier functions can also be composed with logical "AND/OR" operators as described in [13], introducing the concept of Piecewise Barrier Function (PBF) and relying on the definitions of piecewise smooth functions and of the so-called B-derivative to compute the Lie derivatives in (16).

B. Definition of CBF candidates

In Section III-A we defined a safe operating region in terms of a sub-region of the distance-relative velocity plane. The safe operating region can be expressed as a safe set within the ZCBF framework. In particular, we need to find a CBF candidate $h : \mathbb{R}^2 \to \mathbb{R}$, such that $h > 0$ inside the safe region. Then, we will plug the geometrical expressions of distance and relative velocity obtained in Section III-B into the CBF candidate, to obtain the constraint on robot control input that keeps the system inside the safe region by guaranteeing a safe robot velocity. It is worth to note that a mandatory theoretical assumption to guarantee safe robot operations (i.e. the forward invariance of the previously described safe set) would require that the robot motion belongs to the safe set in its initial state. In practice, satisfying this assumption is not particularly challenging (e.g. the robot generally starts its task from rest, when the human is outside of its workspace).

To mathematically describe the safe set and be able to define CBFs over it, we first define the system state χ in terms of distance and relative velocity, as follows:

$$\chi = \begin{bmatrix} d & v_{rel} \end{bmatrix}^T, \quad \dot{\chi} = \begin{bmatrix} v_{rel} & \dot{v}_{rel} \end{bmatrix}^T \tag{19}$$

[2] A continuous function $\alpha : [-b, a) \to [-\inf, \inf)$ is said to belong to extended class $\mathcal{K}$ for some $a, b > 0$ if it is strictly increasing and $\alpha(0) = 0$.

According to the definition of relative velocity in (8), its time derivative becomes:

$$\dot{v}_{rel} = \frac{(x_e - x_h)^T (\dot{v}_e - \dot{v}_h)}{\|x_e - x_h\|} + \frac{\|v_e - v_h\|^2}{\|x_e - x_h\|} + - \frac{\left[(x_e - x_h)^T (v_e - v_h)\right]^2}{\|x_e - x_h\|^3} \tag{20}$$

Now, we can observe that if we assume that the control system is able to act on the robot acceleration as a control input, i.e. $u = \dot{v}_e = \ddot{x}_e$, the dynamics of the state χ can be easily rewritten in the general form of (14) (we omit this passage since it is not necessary in the following computations). Moreover, this assumption also implies that the specification of a CBF as a function of the state χ would already satisfy the requirement on the relative degree between such a function and the controllable input of the system.

To obtain a CBF with a tractable expression in terms of the state vector χ and satisfying the properties of (15), especially considering the particular shape of the safe set, we first approximate the safe region described in Section III-A. In particular, we choose to cover the entire yellow area of Fig. 1 with an arbitrary number of intersecting ellipses. Ellipses are chosen because they can be also viewed as the base (i.e. the zero level) of a paraboloid. Then, paraboloids with an elliptic base are good candidates for ZCBFs: they are positive inside the ellipse and zero on the boundary; their quadratic form and the related derivatives can be easily manipulated in real-time (i.e. the safe region of Fig. 1 has to be adapted online, at each control cycle). Each ellipse is specified in the d/v_{rel} plane by the coordinates of its center and by its widths, so that its equation is the following:

$$\frac{(d - D_i)^2}{a_i^2} - \frac{(v_{rel} - V_i)^2}{b_i^2} = 1 \tag{21}$$

where i is the index of the ellipse, D_i and V_i are the coordinates of its center and a_i and b_i are its half-widths.

The ellipse-based approximation of the safe set is depicted in Fig. 3.

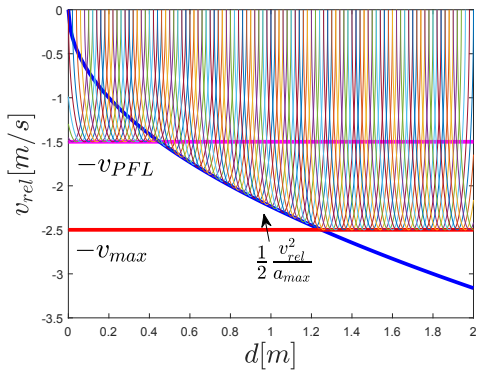


Fig. 3. Approximation of the safe set through a set of ellipses.

It is important to note that the number of ellipses used to compute this approximation is relevant. Of course, the highest is number of ellipses the more accurate is the

approximation. On the other hand, the number should not be too large to affect the real-time adaptive computation of the safe set and the ZCBF-based constraints. Moreover, we should also recall that such ellipses necessarily cover both negative and positive values of the v_{rel} axis. For the reasons remarked in Section III-A, only the negative ones are of interest. Theoretically, the upper bound on v_{rel} in the opposite quadrant of the one shown in Figs. 1 and 3 (i.e. human and robot moving away from each other) is irrelevant. However, a (large) constant value for this bound has to be defined, to properly compute the ellipse-based approximation of the safe set.

Now, the formulation of the final ZCBF is obtained. From each ellipse in the set approximating the safe region, we build a corresponding ZCBF as follows:

$$h_i(\chi) = \begin{cases} 1 - \frac{(d-D_i)^2}{a_i^2} - \frac{(v_{rel}-V_i)^2}{b_i^2} & \text{if } \chi \text{ inside i-th ellipse} \\ 0 & \text{otherwise} \end{cases}$$

where i, D_i, V_i, a_i and b_i are the same as in (21).

As a result, the 3D view of the paraboloids associated to safe set approximation of Fig. 1 can be depicted as shown in Fig. 4. Notice that in this plot both positive and negative values of v_{rel} are shown, being the latter those affected by the safety constraint (i.e. positive relative velocities are bounded by the v_{max} of the robot).

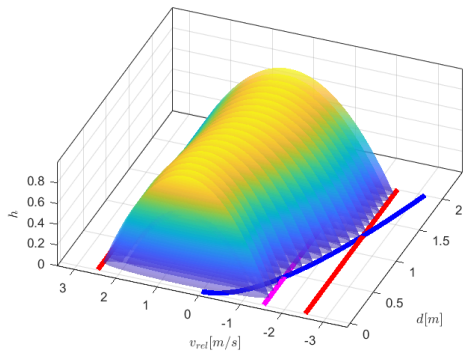


Fig. 4. Paraboloid-shaped CBFs defined on the set of ellipses that approximate the safe set.

Each barrier function would guarantee that the control input keeps the system state inside of it. However, since the overall safe set is approximated by the intersection of the zero levels of paraboloid-shaped barrier functions, the overall ZCBF must be defined as the piecewise smooth logical "OR" of such functions, as described in [13]:

$$h(\chi) = \sum_{i=1}^{n} h_i(\chi) \tag{22}$$

where n is the total number of ellipses used in the approximation. We recall that the derivative of such a ZCBF is meant to be the B-derivative defined in [13]. However, in the following we will keep the standard notation of time derivatives, for the sake of simplicity.

To adopt the previous function as a ZCBF, we must be able to guarantee, by means of the control input design, that the constraint given by (17) is always satisfied. This means that we must compute the time derivative of the barrier function and enforce such a constraint on the control input computation. Considering the time derivative of the i-th paraboloid (if χ is inside the i-th ellipse):

$$\begin{aligned} \dot{h}_i(\chi) &= -2\frac{(d-D_i)\,\dot{d}}{a_i^2} - 2\frac{(v_{rel}-V_i)\,\dot{v}_{rel}}{b_i^2} \\ &= -2\frac{(d-D_i)\,v_{rel}}{a_i^2} - 2\frac{(v_{rel}-V_i)\dot{v}_{rel}}{b_i^2} \end{aligned} \tag{23}$$

and rewriting the time derivative in terms of robot/human position, velocity and acceleration, we obtain:

$$\begin{aligned} &\dot{h}_i(\chi) = -2\frac{(d-D_i)\,v_{rel}}{a_i^2} - 2\frac{(v_{rel}-V_i)}{b_i^2}\cdot \\ &\cdot\left[\frac{(x_e-x_h)^T(\dot{v}_e-\dot{v}_h)}{\|x_e-x_h\|} + \frac{\|v_e-v_h\|^2}{\|x_e-x_h\|} - \frac{\left[(x_e-x_h)^T(v_e-v_h)\right]^2}{\|x_e-x_h\|^3}\right] \end{aligned} \tag{24}$$

Since we consider as input the robot acceleration (i.e. $u = \dot{v}_e = \ddot{x}_e$), we can separate its contribution in (24) as $\dot{h}_i(\chi) = \beta_i(\chi) + \varphi_i(\chi)u$ with:

$$\begin{aligned} &\beta_i(\chi) = -2\frac{(d-D_i)\,v_{rel}}{a_i^2} - 2\frac{(v_{rel}-V_i)}{b_i^2}\cdot \\ &\cdot\left[-\frac{(x_e-x_h)^T\dot{v}_h}{\|x_e-x_h\|} + \frac{\|v_e-v_h\|^2}{\|x_e-x_h\|} - \frac{\left[(x_e-x_h)^T(v_e-v_h)\right]^2}{\|x_e-x_h\|^3}\right] \end{aligned} \tag{25}$$

$$\varphi_i(\chi) = -2\frac{(v_{rel}-V_i)}{b_i^2}\frac{(x_e-x_h)^T}{\|x_e-x_h\|} \tag{26}$$

Finally, computing:

$$\varphi(\chi) = \sum_{1}^{n}\varphi_i(\chi) \quad \beta(\chi) = \sum_{1}^{n}\beta_i(\chi) \tag{27}$$

the set of admissible control values (18) for the robot can be stated as the one satisfying:

$$\varphi(\chi)u \geq -\beta(\chi) - \alpha(h(\chi)) \tag{28}$$

Recalling that $u = \ddot{x}_e$ and substituting the expression of the robot Cartesian acceleration in terms of joint velocities and accelerations previously recalled, i.e. (11), in (28), we can rewrite the constraint that keeps the end-effector inside the safe set in terms of joint space variables, which will become useful for the practical implementation of the control law:

$$\varphi(\chi)J_t\ddot{q} \geq -\varphi(\chi)\dot{J}_t\dot{q} - \beta(\chi) - \alpha(h(\chi)) \tag{29}$$

In the next section, we finally describe how to obtain the safe control input at joint level, assuming that a nominal controller (e.g. PD for trajectory tracking) computes a desired, but potentially unsafe, robot acceleration $\ddot{x}_e^{des}$.

C. Optimization problem

The objective of the Control Barrier Functions is to act as a safety filter, by modifying the nominal control input only when safety is threatened. Hence, the resulting controller is minimally invasive: it allows to precisely follow the desired input when the system velocity is under the ISO/TS limit, while it modifies the behavior when the limit is not respected.

To achieve this behavior, the safety filter is deployed as an optimization problem described as:

$$\begin{aligned} \underset{\ddot{q}}{\text{minimize}} \quad & \|\ddot{x}_e^{des} - \dot{J}_t\dot{q} - J_t\ddot{q}\|^2 \\ \text{subject to} \quad & -\varphi(\chi)J_t\ddot{q} \leq \varphi(\chi)\dot{J}_t\dot{q} + \beta(\chi) + \alpha(h(\chi)) \\ & \|\ddot{q}\|_\infty < \bar{\ddot{q}} \\ & \|\dot{q} + \ddot{q}\cdot\Delta t\|_\infty < \bar{\dot{q}} \end{aligned} \tag{30}$$

where $\ddot{x}_e^{des}$ is the desired (or nominal) control input, $\ddot{q}$ is the commanded robot acceleration in the joint space and $\dot{q}$ is the actual joint velocity. To make the motion feasible, joint acceleration and velocity bounds $\bar{\dot{q}}$ and $\bar{\ddot{q}}$ are added to the constraints

V. Experiments

The validation of the proposed approach has been performed considering a Universal Robots UR5 manipulator, both for simulations and practical experiments. For the latter, the setup is completed by an Intel Realsense D415 depth camera and a human tracking software based on the Nuitrack library. The control software architecture is implemented using OROCOS and ROS frameworks and is similar to the one presented in [11]. The major differences are related to the CBF-based optimization problem, which is the novel contribution of this paper. The optimization problem is solved online using C++ code generated by CVXGEN software.

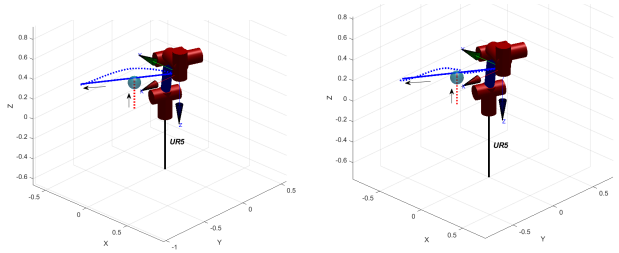
In both simulations and experiments, the parameters in Table I were considered for the PD feedback controller (i.e. computing $\ddot{x}_e^{des}$), for the calculation of the ISO/TS 15066 PFL values and for the ZCBF-based optimization.

TABLE I
PARAMETERS SETTING FOR SIMULATIONS AND EXPERIMENTS

PD Controller gain matrices:	$\mathbf{K_P} = diag\,(60, 60, 60)$ $\mathbf{K_D} = diag\,(20, 20, 20)$
ISO/TS 15066 PFL:	
F_{max} (human arm):	60 N
k (human arm):	40000 N/m
m_h (human arm):	2 Kg
m_r (robot):	10 Kg
$\dot{q}_{max}$	$(1.8, 1.8, 1.8, 1.8, 1.8, 1.8)$ rad/s
$\ddot{q}_{max}$	$(40, 40, 40, 40, 40, 40)$ rad/s^2
ZCBF-based Optimizer:	
n (approximating ellipses)	100
$\bar{\dot{q}}$:	1.8 rad/s
$\bar{\ddot{q}}$:	40 rad/s^2
$\alpha(h(\chi))$:	$\gamma \cdot h(\chi)$

A. Simulations

The analysis of the results obtained in the Matlab/Simulink environment allowed to evaluate the effects of parameters tuning on the trajectories executed by the robot, in the ideal case of a pure kinematic system (i.e. it is assumed that joint velocity commands are instantaneously executed by the manipulator). Moreover, the trajectory of an obstacle emulating the human arm is deterministic and repeatable, while the real tracking of a human body by means of a depth camera is much more noisy and less reliable, even with state-of-the-art technologies. In particular, we focused on the influence of the parameter γ (in the function $\alpha(h(\chi)) = \gamma \cdot h(\chi)$ of our ZCBF) on the behavior of the system inside the safe set. Indeed, the ZCBF constraint expressed by (17) implies that with smaller values of γ the state of the controlled systems is kept at larger distances from the boundary of the safe set $\mathscr{C}$ and vice versa. In our case, staying away from the boundary of the safe set would also mean that the nominal trajectory, which is expected to achieve high velocities in absence human obstacles, would be executed quite differently after the ZCBF optimization. Conversely, larger values of γ would allow the robot to track more accurately the nominal trajectory, while avoiding impacts with the human body only if they are not safe from a PFL perspective. The previous considerations are confirmed by the 3D plots shown in Fig. 5(a) and 5(b), obtained with values of $\gamma = 1$ and $\gamma = 15$, respectively. In both figures, the robot is drawn in its initial position, while the obstacle in its final one, and arrows depict the direction of motion from start to end of the trajectories. As said before, the value of $\gamma = 1$ provides higher deviations from the nominal trajectory, while with $\gamma = 15$ the robot can almost reach the boundary of the safe set and, therefore, can touch the obstacle with a relative velocity close to v_{PFL}.

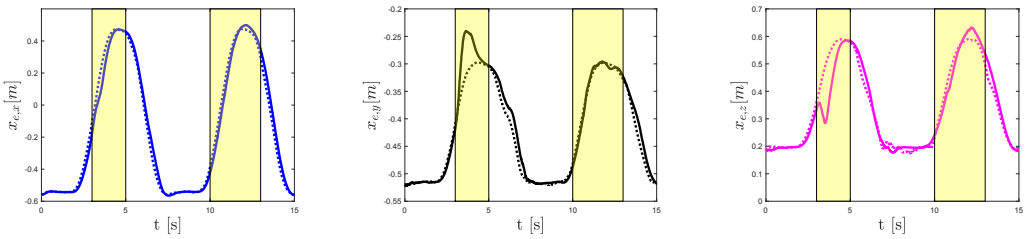


(a) Simulation with $\gamma = 1$ (b) Simulation with $\gamma = 15$

Fig. 5. Robot trajectory (nominal in solid blue, executed in dashed blue) and obstacle trajectory (dashed red).

B. Experiments

Experiments with a real UR5 and a depth camera for human tracking reflected the features observed from simulation tests. The accompanying video clip shows the experiments described in the following, obtained with the $\gamma = 1$ setup. The key aspects that distinguish the practical application are related to the limited update frequency of the control system, mainly due to the frame rate of the camera system (i.e. 60 Hz), and to the joint velocity tracking performance of the low-level controller of the UR5. However, the latter turned to be quite satisfactory, provided that the maximum joint acceleration prescribed by the manufacturer (i.e. 40 rad/s^2) is set. The plots in Fig. 6 show the robot end-effector position, in the three Cartesian coordinates, during two consecutive repetitions of a cyclic trajectory. The dashed lines show the positions tracked in absence of obstacles, while the solid lines are those obtained in presence of a human arm along the robot trajectory. Yellow boxes highlight



(a) x-axis (b) y-axis (c) z-axis

Fig. 6. Cartesian robot positions: nominal trajectory (dashed line) and ZCBF constrained trajectory (solid line).

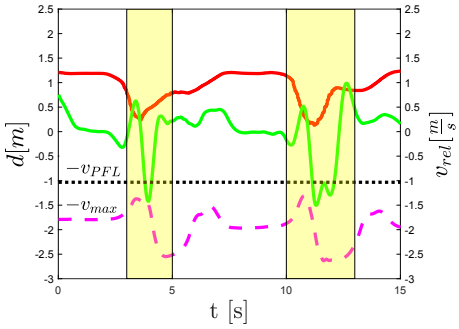


Fig. 7. Distance (red solid line), relative velocity (green solid line), PFL relative velocity bound (black dotted line) and maximum robot velocity (magenta dashed line).

the time periods in which the human operator was closer to the robot. In the first cycle, it can be noted that the presence of the operator causes larger deviations from the nominal trajectory, compared to the second cycle. The plot in Fig. 7 shows the distance between the robot end-effector and the human arm, their relative velocity, the v_{PFL} and the v_{max} bounds, being the latter adaptively computed on the basis of the robot configuration and, therefore, time-varying. From Fig. 7, it can be noted that the relative velocity changes more abruptly in the first cycle, which is due to the fact that the human arm acceleration, estimated by a Kalman filter as described in [11], was higher than in the second cycle. Since this acceleration directly affects the constraint of (28), this fact justifies also the previously mentioned higher deviation from the nominal trajectory. As can be seen in the accompanying video, the relative velocity returns to positive values mainly because the human retracts the hand. Finally, it is worth to note that the relative velocity overcomes the v_{PFL} during some transients, but in such transients the system is still confined in the safe set. On the other hand, a static graphical view of the latter property is difficult to obtain, since the safe set itself is time-varying.

VI. Conclusions

In this paper we have modeled the ISO/TS 15066 as a dynamic safe set and we have exploited a control barrier function approach for enforcing safety without unnecessarily limiting the velocity of the robot. The proposed approach has been successfully validated by simulations and by experiments. Future work aims at introducing human gestures prediction in the definition of the dynamic safe set and of the control barrier function in order to plan the best safe behavior for the collaborative cell. Moreover, the proposed ZCBF framework will be integrated with a Model Predictive Control (MPC) in order to consider the optimization problem in the long term.

References

[1] F. Ferraguti, A. Pertosa, C. Secchi, C. Fantuzzi, and M. Bonfè, "A methodology for comparative analysis of collaborative robots for industry 4.0," in *2019 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2019, pp. 1070–1075.

[2] ISO 10218-1-2011. (since July 1, 2011) Robots and robotic devices – Safety requirements for industrial robots. Part 1: Robots. [Online]. Available: http://www.iso.org

[3] ISO 10218-2-2011. (since July 1, 2011) Robots and robotic devices – Safety requirements for industrial robots. Part 2: Robot systems and integration. [Online]. Available: http://www.iso.org

[4] ISO TS 15066:2016. (since February 15, 2016) Robots and robotic devices – Collaborative robots. [Online]. Available: http://www.iso.org

[5] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

[6] A. M. Zanchettin, N. M. Ceriani, P. Rocco, H. Ding, and B. Matthias, "Safety in human-robot collaborative manufacturing environments: Metrics and control," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 882–893, April 2016.

[7] M. Ragaglia, A. M. Zanchettin, and P. Rocco, "Trajectory generation algorithm for safe human-robot collaboration based on multiple depth sensor measurements," *Mechatronics*, vol. 55, pp. 267 – 281, 2018.

[8] P. Švarný, M. Tesar, J. Behrens, and M. Hoffmann, "Safe physical hri: Toward a unified treatment of speed and separation monitoring together with power and force limiting," 11 2019, pp. 7580–7587.

[9] A. Vysockỳ, H. Wada, J. Kinugawa, and K. Kosuge, "Motion planning analysis according to iso/ts 15066 in human–robot collaboration environment," in *2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2019, pp. 151–156.

[10] N. Mansfeld, M. Hamad, M. Becker, A. G. Marin, and S. Haddadin, "Safety map: A robotics safety evaluation and safe robot design," Workshop on Autonomous Robot Design at the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, may 2018.

[11] F. Ferraguti, C. T. Landi, S. Costi, M. Bonfè, S. Farsoni, C. Secchi, and C. Fantuzzi, "Safety barrier functions and multi-camera tracking for human–robot shared environment," *Robotics and Autonomous Systems*, vol. 124, p. 103388, 2020.

[12] M. Rauscher, M. Kimmel, and S. Hirche, "Constrained robot control using control barrier functions," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 279–285.

[13] L. Wang, A. D. Ames, and M. Egerstedt, "Multi-objective compositions for collision-free connectivity maintenance in teams of mobile robots," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 2659–2664.