Safety in human-robot collaborative manufacturing environments: metrics and control

Andrea Maria Zanchettin, *Member, IEEE,* Nicola Maria Ceriani, *Student Member, IEEE,*
Paolo Rocco, *Member, IEEE,* Hao Ding, *Member, IEEE,* and Björn Matthias

***Abstract*—New paradigms in industrial robotics no longer require physical separation between robotic manipulators and humans. Moreover, in order to optimize production, humans and robots are expected to collaborate to some extent. In this scenario, involving a shared environment between humans and robots, common motion generation algorithms might turn out to be inadequate for this purpose.**

This paper proposes a kinematic control strategy which enforces safety, while maintaining the maximum level of productivity of the robot. The resulting motion of the (possibly redundant) robot is obtained as an output of an optimization-based real-time algorithm in which safety is regarded as a hard constraint to be satisfied. The methodology is experimentally validated on a dual-arm concept robot with 7-DOF per arm performing a manipulation task.

***Note to practitioners*—This paper discusses a way to handle safety and production requirements in human-robot collaborative working environments. A simply tractable set of constraints on the robot velocity is derived in order to meet the regulations imposed by the minimum separation distance criterion. This constraint can be used in real-time to limit the robot velocity, depending on its distance with respect to the human, as perceived by the safeguarding system. Delays in both the perception and the reaction layers as well as stopping time of the robot can systematically be taken into account.**

***Index Terms*—Industrial robots; Safety standards; Human-robot collaboration; Motion planning**

I. Introduction

SMall-medium enterprises (SMEs) are expected to adapt quickly and flexibly to production changes: robotic manipulators represent the best candidates for this requirement. Moreover, for enhanced flexibility, next generation robots are expected to share the same environment with humans and to collaborate with them, to some extent. This however introduces some safety issues that need to be carefully addressed. Requirements for safe human-robot interaction were introduced for industrial robots in [1] and intensively investigated in [2], [3], [4]. Unfortunately, according to the available standards, see [5] for a review, severe restrictions are applied to the robot motion during collaboration with humans. These restrictions are usually transformed into conservative speed limitations (e.g. tool velocity to be less than 250 mm/s) or bounds on driving torques. One of the most intuitive ways to enforce safety during human-machine cooperation is represented by the minimum separation distance, [6], [7], [8]. As the distance between the robot and the human becomes smaller, the velocity of the robot should be reduced accordingly. On the other hand, even in case of a reduced separation distance, the robot should continue its task if its velocity is oriented so that the distance with the human worker may increase.

In this scenario, safety plays the role of a hard constraint, in which respect production could be somehow maximized. The robot trajectory must obey the following constraint[1]

$$\text{distance} \geq \text{velocity } t_b \tag{1}$$

where t_b is the braking time, possibly depending on the robot payload, [6]. This way, provided that all the task and safety constraints are satisfied, the robot motion should result from a compromise between production (maximum or nominal speed) and safety (guarded speed).

In order to allow for safe collaboration, robot controllers have to be equipped with sophisticated sensors as well as with advanced reactive motion planning capabilities. In this scenario, the availability of some additional degrees of freedom in the task specification can be exploited to allow the robot to react to unforeseen events or to implement a safe behavior. In the literature of safe human-robot collaboration (HRC) many reactive control strategies have been implemented, e.g. based on the minimization of some danger-related metrics [10], [11], [12].

The idea of reactive motion planning in dynamic environments has been first introduced in [13] with the so-called velocity obstacles. Particularly interesting is the concept of dynamic envelope developed in [14], which is a region, around the obstacle surface, whose size depends on both robot/obstacle relative position and velocity and is guaranteed to be collision free within a certain prediction horizon. A similar concept has been also presented in [15] as well as with the so-called inevitable collision states presented in [16]. In [17] the so-called distance-controlled velocity has been proposed, the key idea being to adapt the robot velocity on an assigned path based on the perceived distance with respect to environment obstacles. This concept has been further elaborated in [18].

In the human-robot interaction community, it is widely agreed that robot controllers should embed the knowledge on the severity of the injury they might cause to the nearby humans,

A.M. Zanchettin and P. Rocco are with Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria, Piazza L. Da Vinci 32, 20133, Milano, Italy (email: andreamaria.zanchettin@polimi.it; paolo.rocco@polimi.it). N.M. Ceriani is with Tenaris Dalmine, Piazza Caduti 6 Luglio 1944, 1, 24044 Dalmine (email: nceriani@tenaris.com), H. Ding and B. Matthias are with ABB Corporate Research Center, Wallstadter Strasse 59, 68526, Ladenburg, Germany (email: hao.ding@de.abb.com; bjoern.matthias@de.abb.com).

[1] Ongoing standardization process will end up with the new standard ISO/CDTS 15066 [9], currently in draft stage.

[19]. Unfortunately, to the best of the authors' knowledge, none or just few of the aforementioned approaches have been explicitly designed to be consistent with production requirements or aim at maximizing the overall throughput. In fact, in manufacturing environments, the given path cannot be arbitrarily relaxed or modified without violating some of the production constraints. The use of redundancy in the mapping from the task space to the actuation space, see e.g. [20], [21], can be exploited to keep the productivity at a maximum level while performing some kind of safety interventions.

The first work describing the overall picture of a collaborative manufacturing application is [22]. Another interesting work describing the different strategies to account for safety within a bin-picking application was reported in [23], together with a detailed description of the software architecture. A more focused work on collaborative assembly stations can be found, e.g., in [24].

This paper presents a methodology to generate a feasible and optimal motion in presence of both production and safety constraints. In particular, an optimization technique for generating velocity commands is introduced, which maximizes the productivity of the robot while guaranteeing safety. An important feature of the proposed contribution is the use of collision avoidance strategies directly in the actuation (joint) space, thus avoiding the limitations, due to time-consuming computations, of other approaches defined in the 3D Cartesian space [25]. Another novel contribution of the proposed framework is the possibility to consider simultaneously the redundancy of the robot as well as a speed reduction on the desired path, as available safety procedures.

Preliminary versions of these ideas have been originally reported in [26] and [27]. The present paper extends the previous work by fully exploiting the methodology in case of redundant degrees of freedom and presents new and more involved case studies.

The remainder of this paper is structured as follows. In Section II, the concept of safety in terms of motion constraints and inspired by available safety standards is introduced and discussed. Section III describes the optimization-based real-time motion planner adopted to generate safe motions while maximizing the productivity of the robot, consistently with the assigned task. Sections IV and V describe the control architecture and the experimental verifications, respectively. Finally, considerations about the proposed metrics as compared to existing approaches are discussed in Section VI.

II. Safety constraints for collaborative robots

In the following we discuss a condition, briefly discussed in [26], to check whether the current state of motion (i.e. position and velocity) of a given robot manipulator can be regarded as safe or not with respect to the given position of the human/obstacle. The basic idea is to constrain the velocity of the robot to fulfill the safety requirement in (1), which can be represented as shown in Fig. 1. The unsafe region is a combination of velocity/distance for which the worst case braking time[2], see e.g. [29], is not sufficient to avoid collisions, cfr. [17].

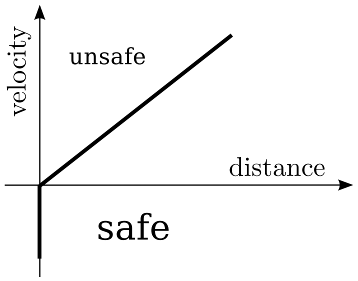


Fig. 1. Graphical representation of the minimum distance separation criterion in (1)

Consider preliminarily a point robot at location $\boldsymbol{r}$ with velocity $\boldsymbol{v}$ and stopping time t_b. For a given obstacle (e.g. a part of the human body) detected at position $\boldsymbol{r_{obst}}$, the minimum separation distance criterion can be written as follows:

$$\|\boldsymbol{r_{obst}} - \boldsymbol{r}\| \geq t_b \frac{(\boldsymbol{r_{obst}} - \boldsymbol{r})^T}{\|\boldsymbol{r_{obst}} - \boldsymbol{r}\|} \boldsymbol{v} \tag{2}$$

where $\left((\boldsymbol{r_{obst}} - \boldsymbol{r})^T / \|\boldsymbol{r_{obst}} - \boldsymbol{r}\|\right) \boldsymbol{v}$ represents the projection of $\boldsymbol{v}$ onto the normalized segment connecting $\boldsymbol{r}$ to $\boldsymbol{r_{obst}}$. The inequality (2) can be rewritten as follows

$$\|\boldsymbol{r_{obst}} - \boldsymbol{r}\|^2 - t_b \left(\boldsymbol{r_{obst}} - \boldsymbol{r}\right)^T \boldsymbol{v} \geq 0$$

The given inequality has a twofold interpretation. First, it defines the set of points $\boldsymbol{r_{obst}}$ around the point robot $\boldsymbol{r}$ to be kept free of any kind of obstacles. This region can be used e.g. within a collision detection algorithm to effectively represent the worst case swept volume. Then, being affine in the velocity $\boldsymbol{v}$, it can be further exploited in a motion planning algorithm as a motion constraint. The second property will be extensively discussed and exploited in Section III.

Consider now a link represented as a beam, as shown in Fig. 2. The position $\boldsymbol{r_s}$ of each point of the link and its velocity $\boldsymbol{v_s}$ can be written in terms of position and velocity of the two end points as follows

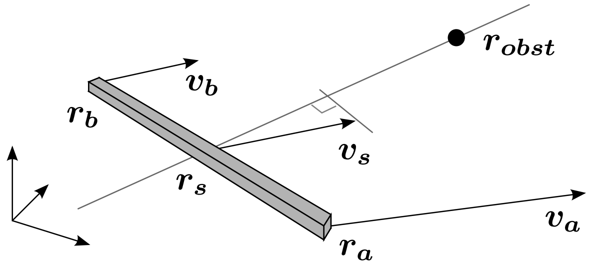


Fig. 2. A rigid beam representing one link

$$\boldsymbol{r_s} = \boldsymbol{r_a} + s\left(\boldsymbol{r_b} - \boldsymbol{r_a}\right) \quad \boldsymbol{v_s} = \boldsymbol{v_a} + s\left(\boldsymbol{v_b} - \boldsymbol{v_a}\right)$$

[2] Unless otherwise specified, with braking time we will refer to the Category 1 (programmed stop) stopping time as defined in [28]. For the sake of completeness, Category 0 (emergency stop) stopping time is in turn measured by allowing the activation of the electromechanical brakes of the robot.

where $s \in [0,1]$. The inequality presented for a point robot has now to be satisfied for every point $\boldsymbol{r_s}$ on the link, hence for all $s \in [0,1]$. Therefore, for a single link, the minimum separation distance can be represented through the following constraints

$$\left\|\boldsymbol{r_{obst}} - \boldsymbol{r_s}\right\|^2 - t_b \left(\boldsymbol{r_{obst}} - \boldsymbol{r_s}\right)^T \boldsymbol{v_s} \geq 0, \forall s \in [0,1]$$

Assuming the link to be rigid[3] we further obtain

$$\beta s + \gamma + \left\|\boldsymbol{r_{obst}} - \boldsymbol{r_s}\right\|^2 \geq 0 \tag{3}$$

where

$$\begin{aligned} \beta &= t_b \left(\boldsymbol{r_b} - \boldsymbol{r_a}\right)^T \boldsymbol{v_a} - t_b \left(\boldsymbol{r_{obst}} - \boldsymbol{r_a}\right)^T \left(\boldsymbol{v_b} - \boldsymbol{v_a}\right) \\ \gamma &= -t_b \left(\boldsymbol{r_{obst}} - \boldsymbol{r_a}\right)^T \boldsymbol{v_a} \end{aligned} \tag{4}$$

Notice that (3) represents an infinite number of inequalities, namely one for each value of $s \in [0,1]$. A sufficient condition for (3) to be satisfied for all $s \in [0,1]$ is the following one

$$\beta s + \gamma + \min_s \left\|\boldsymbol{r_{obst}} - \boldsymbol{r_s}\right\|^2 \geq 0 \tag{5}$$

Notice that the first two terms in the left-hand side of (5) represent a linear, hence monotonic, function in s while the last term is a constant parameter with respect to s, representing the squared distance between the beam and the obstacle.
A necessary and sufficient condition for (5) to be satisfied for all $s \in [0,1]$ is that the two extrema, i.e. $s = 0$ and $s = 1$, of the linear function are consistent with the inequality, i.e.

$$\begin{cases} \gamma + \min_s \left\|\boldsymbol{r_{obst}} - \boldsymbol{r_s}\right\|^2 \geq 0 \\ \beta + \gamma + \min_s \left\|\boldsymbol{r_{obst}} - \boldsymbol{r_s}\right\|^2 \geq 0 \end{cases}$$

Finally, by considering all the n robot links and noticing that β and γ are affine in the robot joint velocity $\dot{\boldsymbol{q}}$, the minimum separation distance criterion can be written as follows

$$t_{b,\rightharpoondown i} \boldsymbol{E_i} \dot{\boldsymbol{q}} \leq \boldsymbol{f_i}, \forall i = 1, \ldots, n \tag{6}$$

where $t_{b,\rightharpoondown i}$ is the maximum joint braking time up to joint i, namely $t_{b,\rightharpoondown i} = \max_{j=1,\ldots,i} t_{b,j}$, and

$$\begin{aligned} \boldsymbol{E_i} &= \begin{bmatrix} \left(\boldsymbol{r_{obst}} - \boldsymbol{r_a}\right)^T \boldsymbol{J_a} \\ \left(\boldsymbol{r_{obst}} - \boldsymbol{r_a}\right)^T \boldsymbol{J_b} - \left(\boldsymbol{r_b} - \boldsymbol{r_a}\right)^T \boldsymbol{J_a} \end{bmatrix} \\ \boldsymbol{f_i} &= \min_s \left\|\boldsymbol{r_{obst}} - \boldsymbol{r_s}\right\|^2 \begin{bmatrix} 1 \\ 1 \end{bmatrix} \end{aligned} \tag{7}$$

while $\boldsymbol{J_a}$ and $\boldsymbol{J_b}$ denote the position Jacobians of points $\boldsymbol{r_a}$ and $\boldsymbol{r_b}$, respectively.

Remark The set of inequalities in (6) represents a straightforward tool to check whether the current robot state of motion satisfies the minimum separation distance or, in other terms, whether the current robot velocity is sufficiently low to allow the robot to stop before a collision occurs. A conservative property has been applied in order to decouple the computation of the minimum distance from the evaluation of the velocities of the end-points of each link. This way, while the inequality in (3) conceptually consists of an infinite number of constraints, a simpler and more conservative version, consisting in only two inequalities for each link, has been obtained for faster evaluation.

As an example, Fig. 3 shows the regions to be avoided around an industrial robot at different positions and with different speeds. In case an obstacle, or part of the human body, $\boldsymbol{r_{obst}}$ lies within the highlighted region, the stopping time might be not sufficient to avoid collisions.
Notice that, as an interesting property of the proposed method for safety evaluation, the volume of the region around the robot to be monitored decreases as the velocity of the robot itself is reduced. Moreover, the volume of the unsafe region around the robot can be shrunk to zero by stopping the robot.

III. Safety-oriented motion planning

So far, we have introduced a metrics to evaluate whether a given configuration/velocity of the robot can be regarded as safe with respect to given points in the 3D workspace. Also, we have shown that whenever the velocity of the robot is not fully specified or constrained (e.g. it can be reduced on a given path), this region can be shrunk by reducing the velocity of the manipulator. In this Section, we introduce an algorithm to efficiently solve the safety-oriented constrained motion planning problem by suitably scaling the trajectory in time and by possibly taking advantage of redundant degrees of freedom.
Let $\boldsymbol{x}$ be a set of variables describing the task. Its relation in terms of joint variables $\boldsymbol{q}$ is the direct kinematic function, i.e. $\boldsymbol{x} = \boldsymbol{k}\left(\boldsymbol{q}\right)$. Its derivative with respect to time allows to specify the robot velocity $\dot{\boldsymbol{q}}$ and its relationship with the task velocity $\dot{\boldsymbol{x}}$ by means of the task Jacobian (whose expression is assumed to be known)

$$\dot{\boldsymbol{x}} = \frac{\partial \boldsymbol{k}}{\partial \boldsymbol{q}} \dot{\boldsymbol{q}} = \boldsymbol{J}\left(\boldsymbol{q}\right) \dot{\boldsymbol{q}} \tag{8}$$

We here assume the well-known path/velocity parametrization of the task

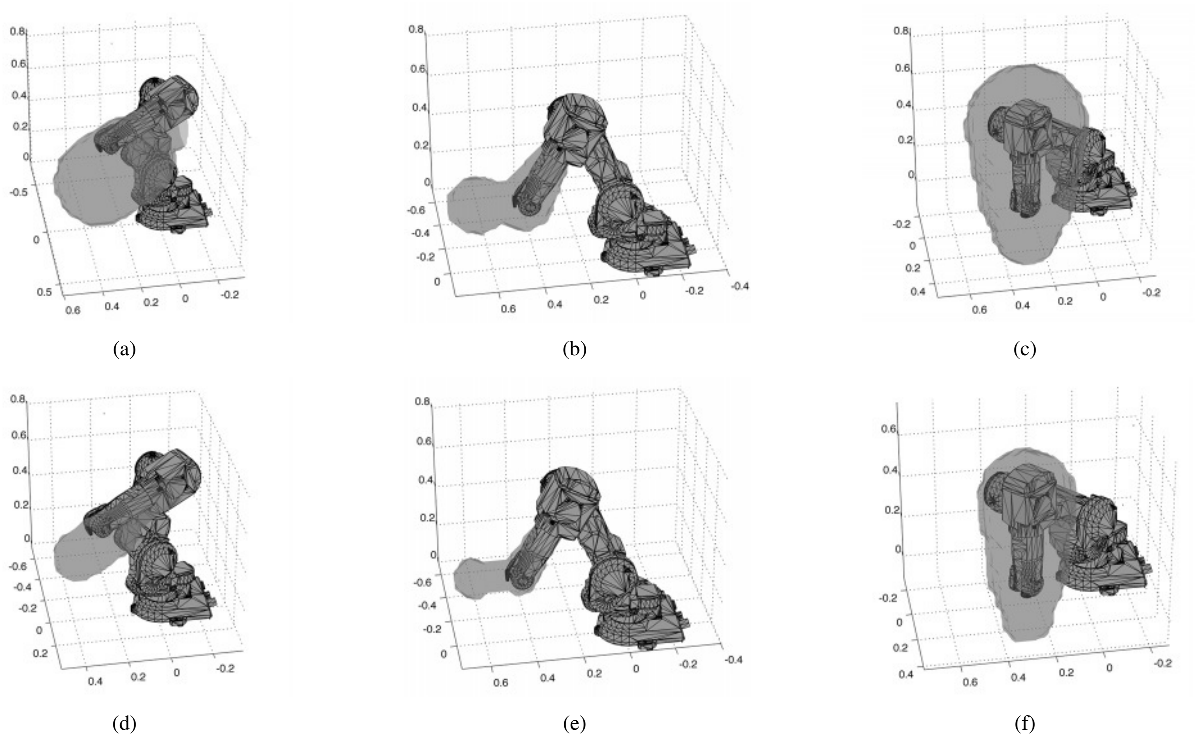
$$\boldsymbol{x}\left(\tau\right) \quad \boldsymbol{x}'\left(\tau\right) = \frac{\partial \boldsymbol{x}}{\partial \tau}$$

where $\boldsymbol{x}\left(\cdot\right)$ is a differentiable function specifying the desired trajectory and assumed to be accessible, at least instantaneously, but not necessarily a priori known. Let $\delta \in [0,1]$ be a scalar quantity adopted to kinematically scale the trajectory in time, see e.g. [30]. The value $\delta = 1$ corresponds to the nominal trajectory, i.e. the one executed at programmed speed, while $\delta = 0$ forces the robot to stop. This way the task velocity along the given path can be simply computed as $\dot{\boldsymbol{x}} = \delta \boldsymbol{x}'\left(\tau\right)$.
In the following, an algorithm to transform the task velocity $\dot{\boldsymbol{x}}$ into joint velocity commands is introduced. To avoid drifts at position level, the differential constraints in (8) are stabilized with a closed-loop inverse kinematics (CLIK) algorithm, [31]. In order to exploit the sensor-based trajectory scaling, we introduce a Linear Programming (LP) optimization problem. The decision variables for the LP problem are the joint velocity commands $\dot{\boldsymbol{q}}_{k+1}$ as well as the velocity scaling factor δ_k which has to be maximized[4]

$$\max_{\delta_k, \dot{\boldsymbol{q}}_{k+1}} \delta_k \tag{9a}$$

[3] Meaning that $\left\|\boldsymbol{r_b} - \boldsymbol{r_a}\right\|$ is constant, hence its derivative is zero.

[4] From now on, subscript k indicates a discrete time instant.



(a) (b) (c)

(d) (e) (f)

Fig. 3. Regions to be avoided around an industrial robot in motion with nominal speed (top) and half speed (bottom). From left to right: with nominal velocity of 150 deg $/s$ on the first axis, moving forward with nominal TCP velocity of 200 mm/s, and moving upward with nominal TCP velocity of 200 mm/s

$$t_{b,\rightharpoonup i}\boldsymbol{E_i}\left(\boldsymbol{q}_k\right)\dot{\boldsymbol{q}}_{k+1} \leq \boldsymbol{f_i}\left(\boldsymbol{q}_k\right), \forall i = 0, \ldots, n-1 \tag{9b}$$

$$\boldsymbol{J}\left(\boldsymbol{q}_k\right)\dot{\boldsymbol{q}}_{k+1} = \delta_k \boldsymbol{x}'\left(\tau_k\right) + \boldsymbol{K\epsilon}\left(\tau_k, \boldsymbol{q}_k\right) \tag{9c}$$

$$-\ddot{\boldsymbol{q}}^{\boldsymbol{max}}\Delta t \leq \dot{\boldsymbol{q}}_{k+1} - \dot{\boldsymbol{q}}_k \leq \ddot{\boldsymbol{q}}^{\boldsymbol{max}}\Delta t \tag{9d}$$

$$-\dot{\boldsymbol{q}}^{\boldsymbol{max}} \leq \dot{\boldsymbol{q}}_{k+1} \leq \dot{\boldsymbol{q}}^{\boldsymbol{max}} \tag{9e}$$

$$\boldsymbol{q_{min}} \leq \boldsymbol{q}_k + \dot{\boldsymbol{q}}_{k+1}\Delta t \leq \boldsymbol{q}^{\boldsymbol{max}} \tag{9f}$$

$$0 \leq \delta_k \leq 1 \tag{9g}$$

where Δt is the discrete time step, $\boldsymbol{\epsilon}$ is the kinematic error, e.g. $\boldsymbol{\epsilon}\left(\tau_k, \boldsymbol{q}_k\right) = \boldsymbol{x}\left(\tau_k\right) - \boldsymbol{k}\left(\boldsymbol{q}_k\right)$, and $\boldsymbol{K} = \boldsymbol{K}^T > 0$ is a control weight. The constraints appearing in (9) have been introduced to account for safety as well as for task (production) constraints and joint position, velocity and acceleration limits. At the discrete time step k, matrices in (9) are evaluated before solving the optimization problem, while the following update law is evaluated to compute the next reference value of τ_{k+1} and of the joint positions $\boldsymbol{q}_{k+1}$

$$\tau_{k+1} = \tau_k + \Delta t \delta_k \quad \boldsymbol{q}_{k+1} = \boldsymbol{q}_k + \Delta t \dot{\boldsymbol{q}}_{k+1}$$

where δ_k and $\dot{\boldsymbol{q}}_{k+1}$ are the outputs of (9).

Remark The LP problem in (9) is always feasible and its solution is always bounded. In fact, provided that $\boldsymbol{\epsilon} = \boldsymbol{0}$ (which happens asymptotically), since $t_{b,\rightharpoonup i}$ is the time required to stop the the first i links of the manipulator and $\boldsymbol{f}_i \geq 0$, it follows that $\dot{\boldsymbol{q}}_{k+1} = \boldsymbol{0}, \delta_k = 0$ is always a solution, though not optimal. Moreover both $\dot{\boldsymbol{q}}_{k+1}$ and δ_k are always lower and upper bounded. It follows that the optimization algorithm needed to solve the problem in (9) will be always able to find a bounded, feasible and optimal solution. Compared to alternative solutions based on dynamic programming, see e.g. [32], [33], adopted to optmize the velocity of the robot along a specified path, the approach described here, though being slightly conservative, does not need neither the knowledge of the full path nor the adoption of a full dynamic model of the manipulator. For this reason, it can be easily applied to significant real-world situations. Moreover, so far none of the alternative approaches has able to efficiently deal with redundant degrees of freedom.

A. Measurements and model uncertainty

The possibility to cope with uncertainties of various nature is crucial for the developed methodology to be applied in real-world scenarios. Two major sources of uncertainty will be discussed. The former is represented by measurement noise or inaccuracies, while the latter takes into account approximations on the model for both the robot geometry and for the motion of the obstacle(s).

The easiest way to account for both types of uncertainty is to introduce a clearance parameter to allow the robot to stop at a certain distance (strictly greater than zero) from the obstacle. In other words, equation (1) can be rewritten as follows:

$$\max\left(0,\ \text{distance} - \Delta\right) \geq \text{velocity } t_b \tag{10}$$

where Δ represents a design parameter. When $\Delta > 0$ is selected, the velocity of the robot is further reduced in order to maintain a clearance distance between the (possibly moving) obstacle and the robot itself. From a geometrical point of view, reducing the point-to-segment distance of a quantity Δ is equivalent to either computing the distance between the segment representing the link and a sphere of radius Δ centered in $\boldsymbol{r_{obst}}$ or to evaluating the distance between point $\boldsymbol{r_{obst}}$ and a capsule of radius Δ built around the robot link, see Fig. 4. Therefore, the parameter Δ can simultaneously be used to:

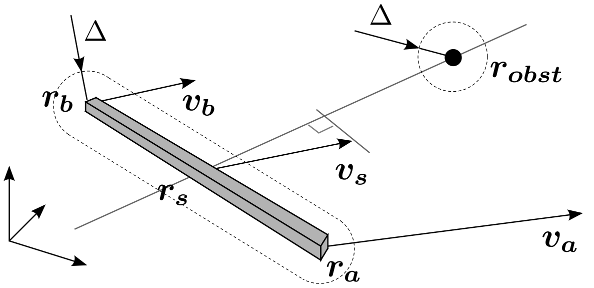


Fig. 4. Dual interpretation of the clearance parameter Δ

- take into account the size of the links of the robot;
- further denote an actual clearance distance;
- account for measurement uncertainty;
- take into account the velocity of the obstacle.

While for the first three cases, the selection of the corresponding values is an easy task, in order to account for the motion of the human worker, a simple empirical rule can be found in safety standards yielding

$$\Delta = K \max_i t_{b,i} \tag{11}$$

where K represents the typical walking speed of humans, as reported in safety regulations, see e.g. [34] which suggests $K = 1.6\ ms^{-1}$ when the minimum distance is larger than $500\ mm$, and $2.0\ ms^{-1}$ otherwise.

In the remainder of this work, we will assume the following definition of the right hand side of the inequalities (7) characterizing the safety constraints:

$$\boldsymbol{f_i} = \left[\max\left(0,\ \min_s \|\boldsymbol{r_{obst}} - \boldsymbol{r_s}\| - \Delta\right)\right]^2 \begin{bmatrix}1\\1\end{bmatrix} \tag{12}$$

B. Kinematic or task redundancy

In case of (task) redundancy the solution of (9) might not be unique, meaning that, the optimal value of δ_k corresponds to several velocity vectors $\dot{\boldsymbol{q}}_{k+1}$ all satisfying the given set of constraints. In this situation, a further optimization step can be performed to optimize the self-motion of the robot in a hierarchical fashion, i.e. having lower priority with respect to the main task. As a redundancy resolution criterion, we assume the self-motion of the robot to be as similar as possible to a reference velocity $\dot{\boldsymbol{q}}^0_{k+1}$. To this end the following LP problem is introduced

$$\min_{\dot{\boldsymbol{q}}_{k+1}} \left\|\dot{\boldsymbol{q}}_{k+1} - \dot{\boldsymbol{q}}^0_{k+1}\right\|_\infty \tag{13a}$$

$$t_{b,\rightharpoondown i}\boldsymbol{E_i}\left(\boldsymbol{q}_k\right)\dot{\boldsymbol{q}}_{k+1} \le \boldsymbol{f_i}\left(\boldsymbol{q}_k\right), \forall i = 0, \ldots, n-1 \tag{13b}$$

$$\boldsymbol{J}\left(\boldsymbol{q}_k\right)\dot{\boldsymbol{q}}_{k+1} = \delta^*_k \boldsymbol{x}'\left(\tau_k\right) + \boldsymbol{K}\boldsymbol{\epsilon}\left(\tau_k, \boldsymbol{q}_k\right) \tag{13c}$$

$$-\ddot{\boldsymbol{q}}^{\boldsymbol{max}}\Delta t \le \dot{\boldsymbol{q}}_{k+1} - \dot{\boldsymbol{q}}_k \le \ddot{\boldsymbol{q}}^{\boldsymbol{max}}\Delta t \tag{13d}$$

$$-\dot{\boldsymbol{q}}^{\boldsymbol{max}} \le \dot{\boldsymbol{q}}_{k+1} \le \dot{\boldsymbol{q}}^{\boldsymbol{max}} \tag{13e}$$

$$\boldsymbol{q_{min}} \le \boldsymbol{q}_k + \dot{\boldsymbol{q}}_{k+1}\Delta t \le \boldsymbol{q}^{\boldsymbol{max}} \tag{13f}$$

where δ^*_k is the optimal value of the cost function in (9), hence a fixed value for this LP problem. Notice that in case $\dot{\boldsymbol{q}}^0_{k+1} = 0$, the minimum norm solution, i.e. the one that minimizes the norm of the joint velocities $\dot{\boldsymbol{q}}_{k+1}$, has been selected for task execution.

The LPs in (9) and (13) result in a bilevel linear programming for which optimized solvers exist. Those solvers are able to speed up the resolution of the whole linear problem, rather than adopting a (naive) cascade of the two LPs.

C. Non-redundant robot

In case $\boldsymbol{J}\left(\boldsymbol{q}_k\right)$ is a square non singular matrix, which happens in 6-DOF industrial manipulators away from singular configurations, the kinematic constraints in (9c) can be solved[5] with respect to $\dot{\boldsymbol{q}}_{k+1}$

$$\dot{\boldsymbol{q}}_{k+1} = \boldsymbol{J}\left(\boldsymbol{q}_k\right)^{-1}\boldsymbol{x}'\left(\tau_k\right) \tag{14}$$

yielding the following simplified LP optimization algorithm

$$\max_{\delta_k} \delta_k \tag{15a}$$

$$\delta_k t_{b,\rightharpoondown i}\boldsymbol{E_i}\left(\boldsymbol{q}_k\right)\dot{\boldsymbol{q}}_{k+1} \le \boldsymbol{f_i}\left(\boldsymbol{q}_k\right), \forall i = 0, \ldots, n-1 \tag{15b}$$

$$0 \le \delta_k \le 1 \tag{15c}$$

In this case, some of the constraints in (9) have been removed, assuming that the joint velocities $\dot{\boldsymbol{q}}_{k+1}$ and the corresponding nominal trajectory are already consistent with the joint position, velocity and acceleration limits. This assumption is reasonable, since in case of a non-redundant robot, the consistency of the end-effector trajectory, as well as of its joint-level counterpart, can be checked with offline programming tools. Furthermore, notice that the quantity in (15b) can be simply expressed as

$$\boldsymbol{E_i}\left(\boldsymbol{q}_k\right)\dot{\boldsymbol{q}}_{k+1} = \begin{bmatrix} \left(\boldsymbol{r_{obst}} - \boldsymbol{r_i}\right)^T \boldsymbol{v_i} \\ \left(\boldsymbol{r_{obst}} - \boldsymbol{r_i}\right)^T \boldsymbol{v_{i+1}} - \left(\boldsymbol{r_{i+1}} - \boldsymbol{r_i}\right)^T \boldsymbol{v_i} \end{bmatrix}$$

where $\boldsymbol{v_i}$ and $\boldsymbol{v_{i+1}}$ are the velocities of the end points of link i that can be easily computed from the robot velocity $\dot{\boldsymbol{q}}_{k+1}$, through appropriate Jacobians.

Finally, notice that the LP problem in (15), having a unique decision variable δ_k, can be actually solved *without* using an LP-solver. Each inequality can be processed independently (and possibly in parallel) to provide a lower or upper bound value $\delta^{(i)}_k$. Then, a simple voting algorithm can return the maximum within $\delta^{(i)}_k$'s, i.e. $\delta_k = \max_i \delta^{(i)}_k$. This is also the case of multiple humans or multiple points along the human body to be monitored. The set of safety-related inequalities consists, in fact, of two rows (inequalities) per each link-point pair, see Fig. 2, thus being highly parallelizable.

[5] Since the asymptotic stability of the CLIK algorithm is a well-known result, we here assume a negligible kinematic error, i.e. $\boldsymbol{\epsilon} = 0$.

D. Closed-ended controllers and communication delays

Modern industrial controllers have dedicated functionalities to implement kinematic scaling of pre-planned trajectories in real-time. In this case, the methodology proposed so far in this Section can be adapted in order to completely exploit the available functionalities (i.e. kinematic inversion and online trajectory scaling) of the proprietary industrial controller. Assuming that joint positions and velocities are available through a real-time interface, an additional PC can solve the LP problem in (15) where the optimal (maximum) value δ_k is forwarded to the robotic controller to be interpreted as a trajectory scaling command (the corresponding implementation will be further discussed in the next Section). Differently from the scenario described in the previous Sections, the LP algorithm is no longer responsible for sending reference values to the low-level controller. Instead, the proprietary controller will interpret the value of δ_k and reduce the speed accordingly.

Finally, notice that in case the algorithm is split into two separate computers, namely the industrial robot controller and the external PC, a communication delay can be expected. Therefore in (9)-(13) or (15), one can adopt the quantity $t_{b,\rightharpoonup i} + t_{delay}$ in place of the pure braking time $t_{b,\rightharpoonup i}$, where t_{delay} is the reaction time of the safeguarding system (typically needed e.g. to handle interrupts and possibly comprising an estimate of the communication delay).

IV. Overall control architecture

This Section describes the overall control architecture for safe human-robot collaboration. The first part discusses the data flow within the proposed control scheme, whilst a higher level picture of the control architecture is given later on in this Section. A preliminary sketch of the control architecture can be found in [27].

A. Data flow

All sensor signals (from both the robot and the exteroceptive safeguarding system) are interpreted within the controller and, based on these interpretations, appropriate actions for the robot are selected. Figures 5 and 6 show the data flow implemented within such a controller in the case of an open- and a closed-ended robot controller, respectively. Both the algorithms consist of three function blocks, namely data collection, data evaluation and interpretation, and behavior selection. The first block, named *data collection*, is responsible for gathering all available information on current state of the robot and its environment. The block labeled *data evaluation and interpretation* examines whether the data are valid and usable or they must be discarded. Based on the data interpretation and on the knowledge about the task to be executed, the block named *behavior selection* selects an appropriate behavior to be executed by the robot.

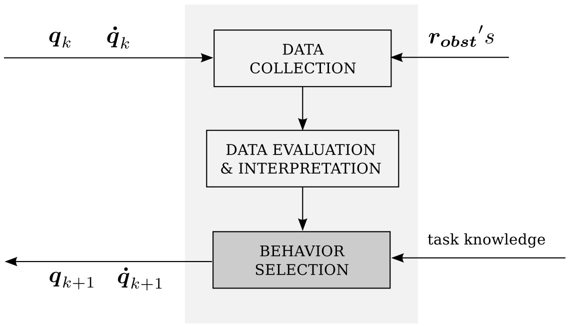


Fig. 5. Data flow within the control algorithm for safe human-robot collaboration

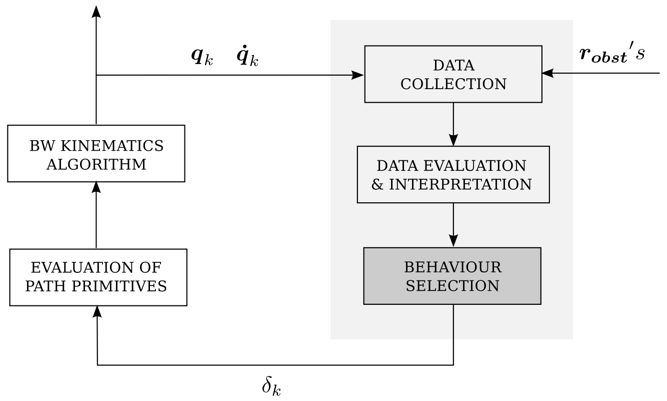


Fig. 6. Data flow within the control algorithm for safe human-robot collaboration in case of closed-ended controller

In the proposed approach, the block named *behavior selection* implements the algorithm in (9)-(13) previously detailed. Notice that such a component will be also responsible for composing the demanded safety constraints with task/production constraints, which depend on the particular application the robot is adopted for. In other words, the block *behavior selection* represents the point where safety and production constraints are composed in order to form motion commands for subsequent execution. The natural way to compose constraints of different nature and to exploit the availability of multiple feasible solutions is an optimization framework, which is actually implemented within this component.

B. Supervision and state machines

In order to sketch an overall picture of the control architecture, we here exploit and complement some previously developed concepts, see e.g. [23], [27]. The state machine in Fig. 7 describes the possible behavior of the system and is responsible of coordinating all the adopted software components. When no human is present in the workspace of the robot, the robot motion can be maintained at maximum (programmed) speed, all actions needed within this behavior are taken inside the state labeled *autonomous behavior*. As a human enters the working area of the robot and is detected by the safeguarding system, the state machine activates the macro-state *collaborative behavior*. The distance of the human fellow co-worker from the robot arm is monitored, in the state *guarded distance*. Here, the kinematic redundancy, if available, is exploited to keep the throughput of the robot at maximum level by selecting an appropriate (and human-aware) null-space velocity in order to guarantee the satisfaction of safety constraints with programmed speed as long as possible.

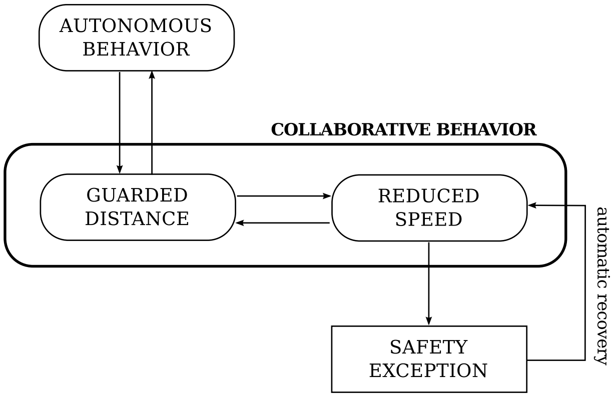


Fig. 7. Functional mode of the robot controller

As the human comes closer and the safety constraints cannot be guaranteed while maintaining the production at maximum level, the state machine changes its state in *reduced speed* and the behavior of the robot is modified accordingly, by reducing its speed and again by taking advantage of the kinematic redundancy, if any.

If the robot comes almost in contact with the human, which must happen at zero speed, the production is temporarily suspended and an execution exception is thrown. The state machine enters the state *safety exception*. Differently from other kinds of exceptions (e.g. emergency stop or other exception not modeled here for brevity), this one can be automatically recovered when the human leaves the workspace of the robot or her/his distance can be considered safe to resume the motion.

The approach can also be extended to deal with multiple human-robot collaboration, by e.g. applying the composition rules of the state machines [35].

V. Case studies

In this Section we describe two relevant case studies which might benefit from the proposed approach. Both the experiments are based on the prototype robot ABB Dual-Arm Concept Robot, also referred to as FRIDA, [36], a 14-DOF dual arm lightweight robot[6].

The first experimental validation implements a manipulation task and exploits all the potentialities of the algorithm, i.e. speed reduction as well as redundancy resolution, in a hard real-time architecture. In turn, the second experiment describes how the approach can be effectively applied in case of a closed-ended control architecture by exploiting the embedded functionalities of a modern industrial controller.

The human can access the workspace of the robot from both the front, the right and the left sides. In both the experiments, the motion of the human is tracked through a surveillance depth camera, namely the Microsoft Kinect. OpenNI drivers have been selected[7]. The uncertainty of sensor measurements, being of few centimeters at most, has been taken into account in the control algorithm, as described in Section III-A.

A. First validation scenario

In this scenario, the robotic task is a manipulation of chemical instruments. The experimental setup is sketched in Fig. 8. For the computation of distances in (5), each link of the robot is regarded as a segment, while the human worker is regarded as a set of capsules (cylinders with hemispherical extremities). The LP algorithms in (9) and (13), implemented using the IBM CPLEX Optimization Studio[8], the evaluation of distances and the computation of the optimum value of δ, as well as of the optimal joint velocities, have been coded on an external real-time Linux Xenomai PC, which is interfaced to the ABB IRC 5 industrial robot controller with cycle time of $\Delta t = 4\ ms$, see [37]. The industrial controller takes $\boldsymbol{q}_{k+1}$ and $\dot{\boldsymbol{q}}_{k+1}$ as joint references and transforms them into torque commands through common position/velocity axis controllers.

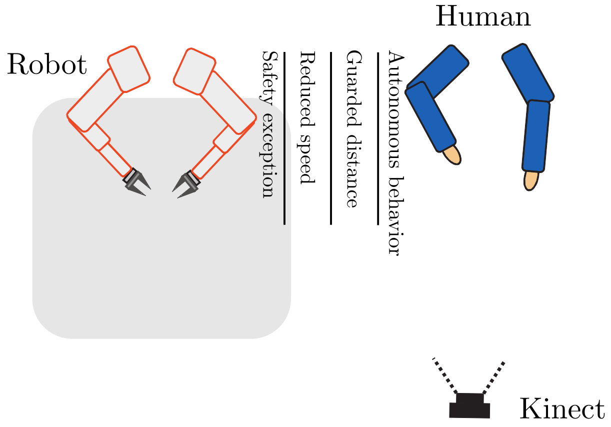


Fig. 8. Experimental setup for HRC in manipulation of chemical instruments

As a verification experiment, the human enters the workspace of the robot and gets closer to it. In the following, we report and discuss the different behaviors of the robot corresponding to the different collaboration modes, as sketched in Fig. 7. The computation time of the LP algorithm in (9) is reported in Tab. I. The adopted LP solver allows the user to specify an upper bound on the computation time. If this bound is about to be exceeded, the solver returns the best feasible solution currently available. On the other hand, as one can see from Tab. I, the maximum computation time is less than 10% of the sampling time Δt, hence guaranteeing hard real-time performance, without the adoption of the any-time feature of the solver.

TABLE I
LP COMPUTATION TIME

Average	95th percentile	Max
245.7 μs	308.0 μs	370.0 μs

Time histories of δ, the minimum distance $\min_i \sqrt{\boldsymbol{f}_i}$, the elbow angle α, see [38], as well as the norm of the TCP

[6] Though the case studies are applied on a redundant manipulator in order to show all the possibilities discussed in the previous Sections, the same algorithm can be clearly adopted, without substantial modifications, for traditional 6-DOF arms.

[7] http://www.openni.org/.

[8] http://www-03.ibm.com/software/products/it/it/ibmilogcple/.

(Tool Center Point) velocity v will be reported to support the discussion.

Figure 9 reports the behavior named *guarded distance*. For the proactive behavior to be activated prior to a speed reduction, we selected a null-space velocity $\dot{\boldsymbol{q}}^0_{k+1}$ in (13) in order to steer the elbow of the robot close to its down-most position, i.e. $\alpha = 20$ deg. As soon as the human worker entered the working area of the robot, which happened at time instant $t = 14\ s$, its elbow retracted towards the commanded position. This behavior is meant to take advantage of the kinematic redundancy of the robot. In fact, no speed reduction was commanded and the robot was able to continue its task with maximum productivity, i.e. with $\delta = 100$ %.

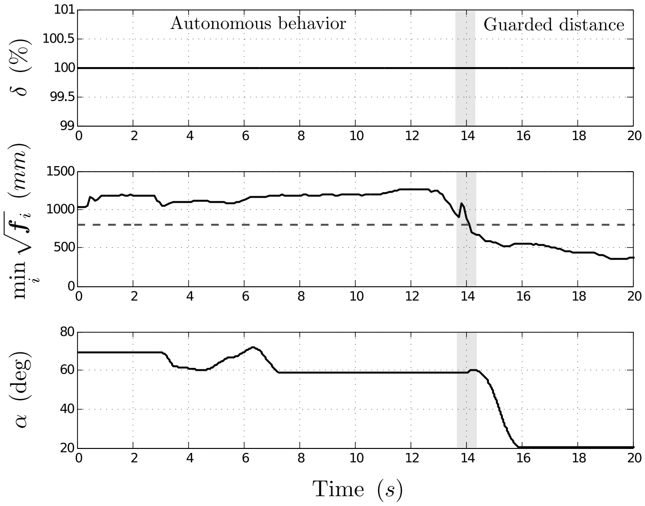


Fig. 9. Proactive elbow-down motion triggered by human vicinity

Figure 10 shows the behavior activated within the state *reduced speed*. At a certain point, namely in between $t = 58\ s$ and $t = 60\ s$ and right before $t = 66\ s$, the velocity of the robot had to be reduced from the programmed one. The elbow was also retracted from its commanded position ($\alpha = 20$ deg). However, due to joint limits, the elbow angle, having a lower priority, no longer recovered its nominal position within the time instants from $t = 60\ s$ until $t = 66\ s$.

Finally, Fig. 11 reports the behavior named *safety exception*. In particular, the minimum distance between the robot and the human worker abruptly dropped to near-zero values. In this case, the robot could not continue the task and was gently stopped by the algorithm right after time instant $t = 32\ s$. As soon as a relative (and safe) distance of $400\ mm$ was (re)established, the robot resumed the task by fully exploiting its acceleration capabilities to regain the programmed speed, corresponding to $\delta = 100$ %, which happened right after time instant $t = 38\ s$. We remind that, contrarily to other kind of exceptions, the nominal behavior of the robot was automatically recovered, without any acknowledgment from the human worker.

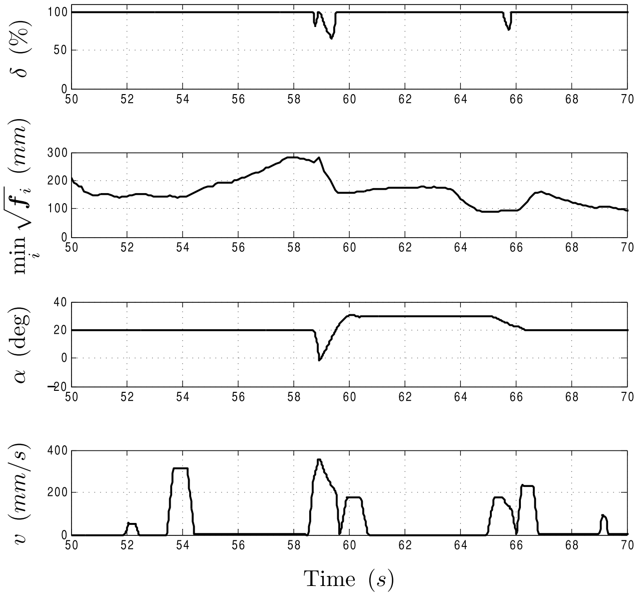


Fig. 10. Slow down behavior and elbow motion

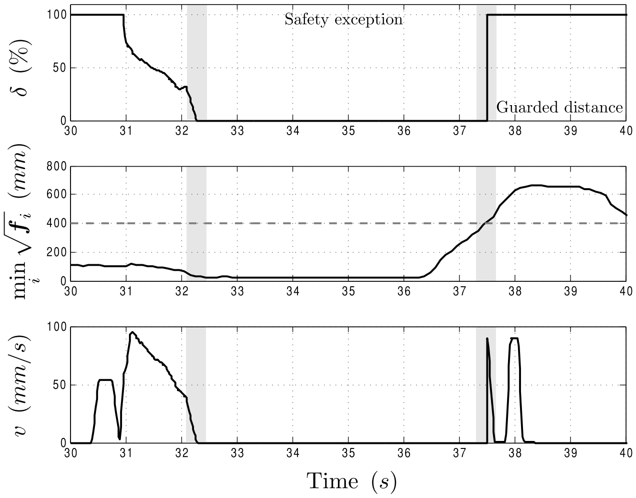


Fig. 11. Safety exception and automatic task resumption

B. Second validation scenario

This second validation scenario describes how the methodology developed in this paper can be effectively applied in case of a closed-ended industrial controller not capable of handling a real-time communication with an external PC. Figure 12 shows the scheme of the HRC assembly station. The main field of interest here is the small-part assembly in an industrial environment.

In this setup, the assembly scenario is with two Kinect cameras for workspace supervision. These cameras detect the workers' position and send this information to the application control instance running on an external PC. Two interaction zones in which direct contact between the robot and the human might occur are designated in this example scheme. One zone is located between the robot and the human standing next to each other, the other is located between the robot and a human standing across the workbench.

Similarly to the previous experimental demonstration, the *guarded distance* behavior, which corresponds to the proactive utilization of the redundant degree of freedom, is activated based on the relative distance between the robot and the human worker operating next to the robot itself. Differently from the previous demonstration, the other two behaviors *reduced speed* and *safety exception* are in turn triggered by means of properly defined distance thresholds. Distance thresholds governing the activation of the different mode, see Fig. 12, are computed by applying the general rule in (1). In particular, they can be made adaptive to the programmed speed or kept fixed. When the human, standing in front of the robot, crosses the first threshold (or the one next to the robot crosses the second threshold), the *reduced speed* mode of the corresponding robot arm is triggered. In case the human further approaches to the robot and crosses the next threshold, a *safety exception* is thrown and the standstill mode will be triggered. As in the previous experimental verification, the normal operation can be automatically recovered when the human intervention is gone. Fig. 13 shows the activation or the deactivation of the corresponding functional mode during the multiple human-robot collaboration. The images of the Kinect cameras (front and side) are displayed in a self-developed graphical user interface.

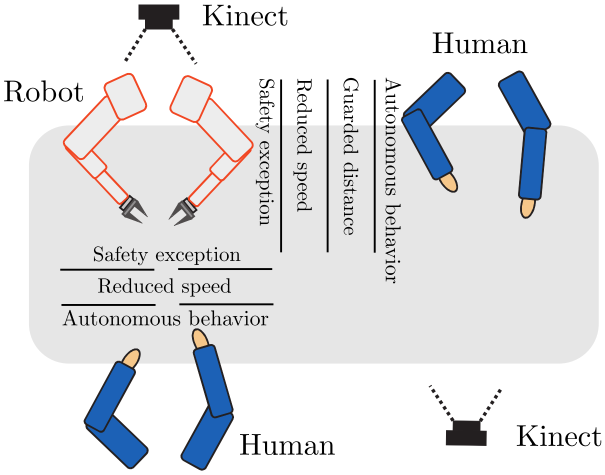


Fig. 12. Experimental setup for multiple HRC in industrial assembly

VI. Discussion

In this Section, we briefly discuss possible relations of our approach with pre-existing solutions, either available in the literature or within commercial products.

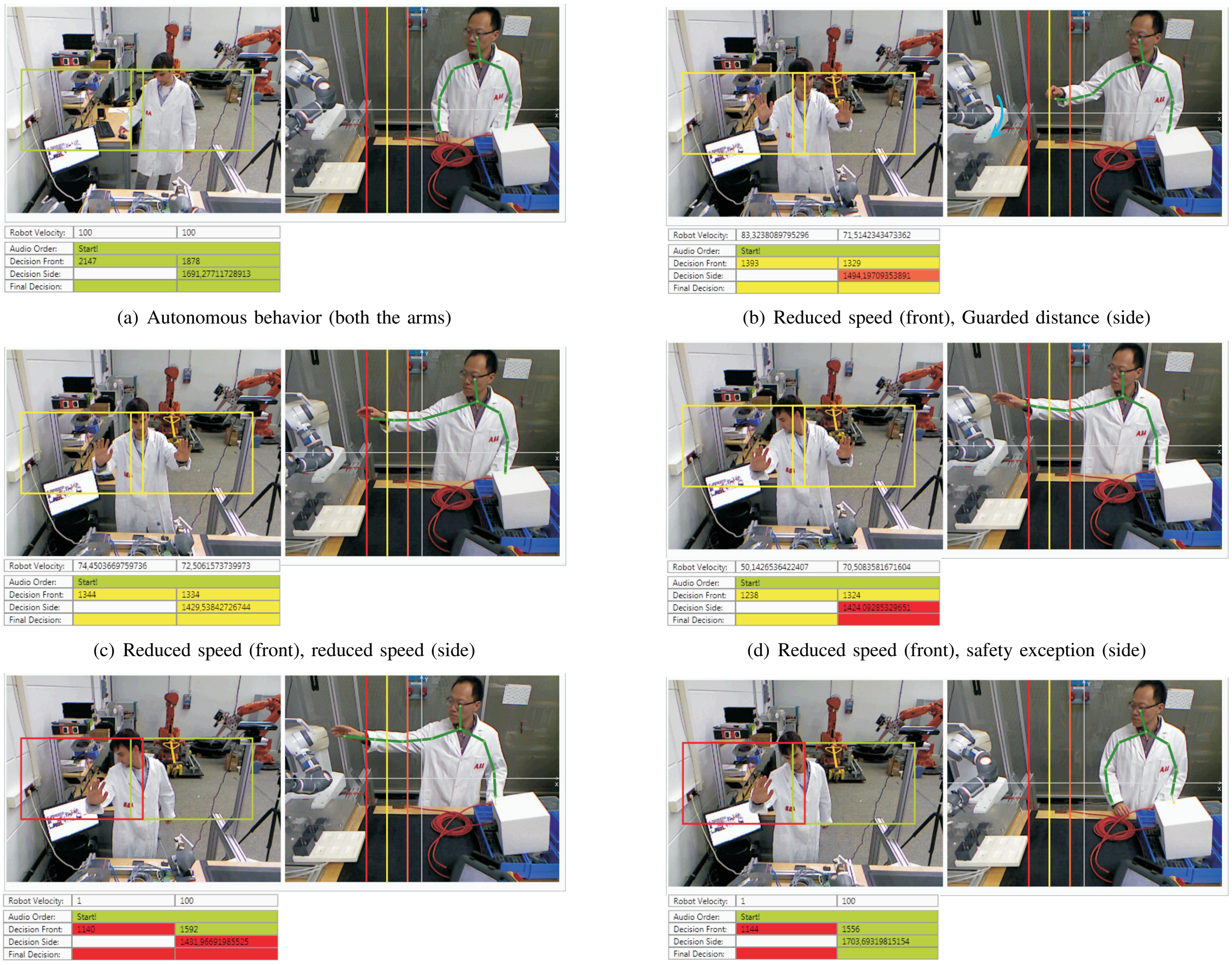
As discussed in the introduction, many authors have already proposed methodologies in which the speed of the robot is reduced along a given path, depending on certain metrics. The selection of these metrics are crucial for achieving the best trade-off between the throughput of the robot and the safety of the human worker(s). The aim of this Section is then to compare existing approaches with the one developed herein.

An approach quite similar to the one proposed in this work can be found in [8]. The speed and the relative distance are monitored, but the adopted metrics generates a considerably discretized decision (robot stopped, 50% or 100% of programmed speed), hence being suboptimal with respect to the approach described in this paper which adopts a continously modulated speed scaling, rather than a coarse discrete set of values. Moreover, only the TCP velocity, rather than the full robot, has been taken into account.

As mentioned in the introduction, the approach developed in this paper is somehow related to the concept of inevitable collision states originally presented in [16]. However, the concept proposed in Fraichard et al. has been mainly adopted in path planning for car-like robots within an RRT-like planner. In turn, our approach systematically includes the mapping between the joint and the Cartesian space by means of proper Jacobians matrices in order to make the approach suitable for articulated industrial manipulators, with strict TCP path constraints. Finally, in our paper collisions are not necessarily avoided, they are only guaranteed to eventually happen with the robot still, as prescribed by safety regulations. Furthermore, the extension to redundant degrees of freedom represents a novelty of this work, with respect to the state of the art.

In [30], [23], Haddadin et al. proposed a trajectory scaling algorithm originally based on torque monitoring, and then also adapted to distance. While the approach is very similar to the one proposed herein, to the best of the authors' knowledge the scaling factor was selected by means of heuristics, rather than being optimized for productivity. While this fact can be tolerated, since the robot was proved to be able to safely interact with its human fellow co-worker, from a production perspective, there is no guarantee that the mentioned method could achieve the *best* trade-off between safety and productivity. Moreover, in our approach safety is considered as a hard constraint to be satisfied, rather than a cost function to be somehow optimized, as traditionally conceived within the framework of artificial potential fields, see e.g. [11], [12], [39]. Finally, all the approaches based on potential function suffer from the major drawback of being sensitive to tuning parameters. In turn, in our approach, the sole parameters to be tuned are the braking time, which is usually known, and possibly the clearance distance, for which general guidelines from safety standards can be adopted.

The approach discussed by Kuhn et al. in [18] consists in generating an augmented silhouette around the robot whose dimension depends on the velocity of the robot itself. This region is generated directly in the frames of a ceiling-mounted surveillance camera. While the approach is comparable to the one presented in this paper, there are three major differences. First, the approach developed here allows to systematically define the relationship between the volume around the robot to be monitored and the velocity (magnitude and direction) of the robot itself, also accounting for possible delays in the perception/action toolchain. Second, no assumption has been made here on the safeguarding system. In particular, the KINECT camera has been adopted within the experimental verifications, but, in principle, any other kind of surveillance sensor can be used. Finally, the property of our metrics to be



Robot Velocity:	100	100
Audio Order:	Start!	
Decision Front:	2147	1878
Decision Side:		1691,27711728913
Final Decision:	🟩	🟩

(a) Autonomous behavior (both the arms)

Robot Velocity:	83,3238089795296	71,5142343473362
Audio Order:	Start!	
Decision Front:	1393	1329
Decision Side:		1494,19709353891
Final Decision:	🟨	🟨

(b) Reduced speed (front), Guarded distance (side)

Robot Velocity:	74,4503669759736	72,5061573739973
Audio Order:	Start!	
Decision Front:	1344	1334
Decision Side:		1429,53842726744
Final Decision:	🟨	🟨

(c) Reduced speed (front), reduced speed (side)

Robot Velocity:	50,1426536422407	70,5083581671604
Audio Order:	Start!	
Decision Front:	1238	1324
Decision Side:		1424,09285329651
Final Decision:	🟨	🟥

(d) Reduced speed (front), safety exception (side)

Robot Velocity:	1	100
Audio Order:	Start!	
Decision Front:	1140	1592
Decision Side:		1431,96691985525
Final Decision:	🟥	🟥

(e) Safety exception (front), safety exception (side)

Robot Velocity:	1	100
Audio Order:	Start!	
Decision Front:	1144	1556
Decision Side:		1703,69319815154
Final Decision:	🟥	🟩

(f) Safety exception (front), autonomous behavior (side)

Fig. 13. Activation or deactivation of the corresponding functional mode(s)

affine in the robot control input (the vector of joint velocities) allows for an easier deployment on an industrial controller.

As for commercially available solutions of HRC in industrial environments, ABB SafeMove[9] enables human workers and robots to work together without compromising on safety. It uses geometrical and speed restrictions, maintaining automatic operation. Similar safety-rated control solutions of industrial robots are also available on the market like Safe Production[10], Safe Operation[11], Dual Check Safety[12], etc. But redundant DOF and dynamic motion planning are normally not considered, which is different from the approach proposed in the paper.

Furthermore, human workers, collaborating with conventional industrial robots, are usually perceived by safety-rated devices like safety laser scanner, SafetyEYE[13] for 3D zone monitoring, etc. However, the 3D coordinates cannot be detected in a safety-rated way by existing commercial solutions. In the HRC applications considered in this paper, the non-safety Kinect cameras can be used because of the inherently safety design of the robot and the proper risk assessment.

Different from manual assembly, robotic automation and fixed automation, HRC has its own benefit while considering the production volume and the unit cost, like the manufacturing paradigm proposed in [40]. However, a drawback of present HRC implementations particularly in industrial environments is that the reaction in the event of an impending risk to a human worker is to stop robot motion, thus interrupting the application and curtailing productivity. The proposed approach is to maximize the productivity during the operation with respect to safety constraints. Experiments described above have shown improvement of the uptime of the collaborative applications, especially in the examples by moving down the elbow in case that the worker approaches. Speed reduction or standstill is mostly sufficient, which can be automatically recovered to normal operation from exception. As a result, this

[9] http://www.abb.com/.
[10] http://www.reisrobotics.com/.
[11] http://www.kuka.com/.
[12] http://www.fanucrobotics.com/.
[13] http://www.pilz.com/.

reduces the frequency of unintended contacts between worker and robot, and upholds the productivity.

VII. Conclusions

This paper contributes in introducing a metrics for safety evaluation in human-robot collaborative manufacturing environments. This metrics depends on the relative distance between the human worker and the robot, as well as on the robot itself. A control strategy, modulating the velocity of the robot on an assigned path, has been proposed to enforce a safe HRC. Experiments have been discussed both in case of adoption of an open robot controller, and in case of a traditional, closed-ended, scenario.

Acknowledgments

A.M. Zanchettin, N.M. Ceriani and P. Rocco would like to thank Prof. Renato Casagrandi for the chemical instruments needed for the experiments in the first validation scenario. H. Ding and B. Matthias thank Jokab Heyn, Junjie Zhang and Malte Schipper for their contributions to the setup of the multiple HRC assembly station.

References

[1] ISO TC184/SC2, *ISO 10218 - Robots for industrial environments – Safety requirements – Part 1: Robot*, 2006, (This version is no longer in effect.).

[2] M. Zinn, O. Khatib, and B. Roth, "A new actuation approach for human friendly robot design," in *IEEE International Conference on Robotics and Automation, ICRA*, 2004, pp. 249–254.

[3] M. Zinn, O. Khatib, B. Roth, and J. Salisbury, "Playing it safe [human-friendly robots]," *IEEE Robotics & Automation Magazine*, vol. 11, no. 2, pp. 12–21, 2004.

[4] S. Haddadin, A. Albu-Schaeffer, and G. Hirzinger, "Requirements for safe robots: measurements, analysis and new insights," *International Journal of Robotics Research*, vol. 28, pp. 1507–1527, 2008.

[5] J. Fryman and B. Matthias, "Safety of industrial robots: From conventional to collaborative applications," in *German Conference on Robotics, ROBOTIK*, 2012.

[6] ANSI/RIA, *R15 06 - "Safety requirements for industrial robots and robot systems"*, 1999.

[7] ISO TC184/SC2, *ISO 10218 - Robots for industrial environments – Safety requirements – Part 1: Robot*, 2011.

[8] J. Marvel, "Performance metrics of speed and separation monitoring in shared workspaces," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 2, pp. 405–414, 2013.

[9] ISO TC184/SC2, *ISO/CD 15066 Robots and robotic devices – Safety requirements for industrial robots – Collaborative operation*, 2013.

[10] D. Kulić and E. Croft, "Real-time safety for human–robot interaction," *Robotics and Autonomous Systems*, vol. 54, no. 1, pp. 1–12, 2006.

[11] B. Lacevic, P. Rocco, and A. M. Zanchettin, "Safety assessment and control of robotic manipulators using danger field," *IEEE Transactions on Robotics*, vol. 29, pp. 1257–1270, 2013.

[12] A. M. Zanchettin, B. Lacevic, and P. Rocco, "A novel passivity-based control law for safe human-robot coexistence," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2012.

[13] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[14] R. Vatcha and J. Xiao, "Perceiving guaranteed continuously collision-free robot trajectories in an unknown and unpredictable environment," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2009.

[15] D. Wilkie, J. van den Berg, and D. Manocha, "Generalized velocity obstacles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2009.

[16] T. Fraichard and H. Asama, "Inevitable collision states âĂŤ a step towards safer robots?" *Advanced Robotics*, vol. 18, no. 10, pp. 1001–1024, 2004.

[17] D. Henrich and S. Kuhn, "Modeling intuitive behavior for safe human/robot coexistence cooperation," in *IEEE International Conference on Robotics and Automation, ICRA*, 2006.

[18] S. Kuhn and D. Henrich, "Fast vision-based minimum distance determination between known and unkown objects," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2007.

[19] S. Haddadin, S. Haddadin, A. Khoury, T. Rokahr, S. Parusel, R. Burgkart, A. Bicchi, and A. Albu-Schaeffer, "On making robots understand safety: Embedding injury knowledge into control," *International Journal of Robotics Research*, vol. 31, no. 13, pp. 1578–1602, 2012.

[20] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *International Journal of Robotics Research*, vol. 6, no. 2, pp. 3–15, 1987.

[21] A. M. Zanchettin and P. Rocco, "A general user-oriented framework for holonomic redundancy resolution in robotic manipulators using task augmentation," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 514–521, 2012.

[22] J. T. C. Tan, F. Duan, R. Kato, and T. Arai, "Safety strategy for human–robot collaboration: Design and development in cellular manufacturing," *Advanced Robotics*, vol. 24, no. 5-6, pp. 839–860, 2010.

[23] S. Haddadin, M. Suppa, S. Fuchs, T. Bodenmüller, A. Albu-Schäffer, and G. Hirzinger, "Towards the robotic co-worker," in *International Symposium on Robotics Research, ISRR*. Springer, 2011, pp. 261–282.

[24] B. Matthias, S. Kock, H. Jerregard, M. Kallman, I. Lundberg, and R. Mellander, "Safety of collaborative industrial robots: Certification possibilities for a collaborative assembly robot concept," in *IEEE International Symposium on Assembly and Manufacturing, ISAM*, 2011.

[25] P. Jiménez, F. Thomas, and C. Torras, "3d collision detection: a survey," *Computers & Graphics*, vol. 25, no. 2, pp. 269–285, 2001.

[26] A. M. Zanchettin and P. Rocco, "Path-consistent safety in mixed human-robot collaborative manufacturing environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2013.

[27] H. Ding, J. Heyn, B. Matthias, and H. Staab, "Structured collaborative behavior of industrial robots in mixed human-robot environments," in *IEEE International Conference on Automation Science and Engineering, CASE*, 2013.

[28] IEC, *EN 60204 - "Safety of machinery. Electrical equipment of machines. General requirements"*, 2006.

[29] T. Dietz and A. Verl, "Simulation of the stopping behavior of industrial robots using a complementarity-based approach," in *Advanced Intelligent Mechatronics (AIM), 2011 IEEE/ASME International Conference on*, 2011.

[30] S. Haddadin, A. Albu-Schaeffer, A. De Luca, and G. Hirzinger, "Collision detection and reaction: A contribution to safe physical human-robot interaction," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2008.

[31] B. Siciliano, "A closed-loop inverse kinematic scheme for on-line joint-based robot control," *Robotica*, vol. 8, no. 3, pp. 231–243, 1990.

[32] J. E. Bobrow, S. Dubowsky, and J. Gibson, "Time-optimal control of robotic manipulators along specified paths," *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 3–17, 1985.

[33] K. Shin and N. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Transactions on Automatic Control*, vol. 30, no. 6, pp. 531–541, 1985.

[34] *ISO 13855 - Safety of machinery âĂŤ Positioning of safeguards with respect to the approach speeds of parts of the human body*, 2010.

[35] H. Ding, M. Schipper, and B. Matthias, "Collaborative behavior design of industrial robots for multiple human-robot collaboration," in *International Symposium on Robotics, ISR*, 2013.

[36] S. Kock, T. Vittor, B. Matthias, H. Jerregard, M. Kallman, I. Lundberg, R. Mellander, and M. Hedelind, "Robot concept for scalable, flexible assembly automation: A technology study on a harmless dual-armed robot," in *IEEE International Symposium on Assembly and Manufacturing, ISAM*, 2011.

[37] A. Blomdell, G. Bolmsjo, T. Brogardh, P. Cederberg, M. Isaksson, R. Johansson, M. Haage, K. Nilsson, M. Olsson, T. Olsson, *et al.*, "Extending an industrial robot controller: implementation and applications of a fast open sensor interface," *IEEE Robotics & Automation Magazine*, vol. 12, no. 3, pp. 85–94, 2005.

[38] K. Kreutz-Delgado, M. Long, and H. Seraji, "Kinematic analysis of 7 dof anthropomorphic arms," in *IEEE International Conference on Robotics and Automation, ICRA*, 1990.

[39] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The international journal of robotics research*, vol. 5, no. 1, pp. 90–98, 1986.

[40] H. Ding and B. Matthias, “Safe human-robot collaboration combines expertise and precision in manufacturing,” *ATP Edition*, vol. 10, pp. 22–25, 2013.