ORIGINAL ARTICLE

Enhancing fluency and productivity in human-robot collaboration through online scaling of dynamic safety zones

Lorenzo Scalera[1] · Andrea Giusti[2] · Renato Vidoni[3] · Alessandro Gasparetto[1]

Abstract

Industrial collaborative robotics is promising for manufacturing activities where the presence of a robot alongside a human operator can improve operator's working conditions, flexibility, and productivity. A collaborative robotic application has to guarantee not only safety of the human operator, but also fluency in the collaboration, as well as performance in terms of productivity and task time. In this paper, we present an approach to enhance fluency and productivity in human-robot collaboration through online scaling of dynamic safety zones. A supervisory controller runs online safety checks between bounding volumes enclosing robot and human to identify possible collision dangers. To optimize the sizes of safety zones enclosing the manipulator, the method minimizes the time of potential stop trajectories considering the robot dynamics and its torque constraints, and leverages the directed speed of the robot parts with respect to the human. Simulations and experimental tests on a seven-degree-of-freedom robotic arm verify the effectiveness of the proposed approach, and collaborative fluency metrics show the benefits of the method with respect to existing approaches.

Keywords Industry 4.0 · Collaborative robotics · Human-robot interaction · Trajectory planning · Dynamic safety zones · Collaborative fluency metrics

1 Introduction

Nowadays, collaborative robotics is one of the enabling technologies of Industry 4.0 [1]. This paradigm allows a manipulator to work side-by-side with a human operator and to realize a safe sharing of workspace during manufacturing activities. The need of collaborative robotics application in modern manufacturing is driven by the request of flexibility of tasks and automation settings, and of rapidly variable productivity in lot sizes and time to market [2, 3]. Furthermore, the introduction of collaborative robotics can reduce the incidence of occupational risks among the employees, and thus increase their working conditions, while enhancing the performance of the production line [4].

Implementing safety in a human-robot collaborative application is fundamental, and several strategies to design safe collaborative robotic workcells have been proposed in the literature [5, 6]. These strategies mainly rely on the minimization of injury when human and robot in motion come in contact, or on schemes that preemptively avoid a collision.

The minimization of injury can be obtained by leveraging compliant joints with nonlinear stiffness [7], sensitive covers for robot links [8], or pre-loaded structures that dissipate contact energy [9]. On the contrary, approaches for avoiding a potential collision between operator and robot include, for instance, works that consider probabilistic predictions of the worker's motion [10], or that model robot and operator as capsules and alter on-the-fly the nominal path of the manipulator to prevent collisions with the co-worker [11]. The authors in [12] implemented an optimal path planning approach for avoiding collisions

✉ Lorenzo Scalera
lorenzo.scalera@uniud.it

Andrea Giusti
andrea.giusti@fraunhofer.it

Renato Vidoni
renato.vidoni@unibz.it

Alessandro Gasparetto
alessandro.gasparetto@uniud.it

[1] Polytechnic Department of Engineering and Architecture, University of Udine, 33100 Udine, Italy

[2] Fraunhofer Italia Research, 39100 Bolzano, Italy

[3] Faculty of Science and Technology, Free University of Bozen-Bolzano, 39100 Bolzano, Italy

by predicting the volume occupied by an operator during the collaboration with the robot. Furthermore, a control strategy for obstacle avoidance is presented in [13], in which the predefined robot motion is modified online by means of an interpolation with Bézier curves.

Four methods of safe operation for collaborative robotic applications are introduced by the standard ISO 10218: hand guiding, safety-rated monitored stop, speed and separation monitoring, power and force limiting [14, 15]. In addition, the technical specification ISO/TS 15066 supplements and supports the industrial robot safety standard, and provides additional guidance on the identified operational functions for collaborative robotics [16]. The technical specification also depicts a guideline to compute a protective separation distance.

The speed and separation monitoring (SSM) criterion is applied in [17], where the speed of the manipulator is adapted with respect to the relative motion between robot and human. In [18], a collision preventing controller predicts possible unwanted contacts with the operator and modifies the robot speed accordingly to the collision danger. In [19], bounding volumes as sphere swept lines (SSLs) are implemented to compute online proximity checks between the robot and its human co-worker. More recently, the SSLs are adopted by the authors in [20] to guarantee safety by considering path-consistent trajectories for every possible human behavior. In [21], the probability of human hand intrusion in the robot workspace is estimated using interference theory applying the SSM safety method. Furthermore, in [22] dynamically scaled safety zones implemented as SSLs are adopted to ensure a safe collaboration according to the ISO/TS 15066 and to compute stop trajectories based on a set of tentative stop times. However, the main drawback of that work is that the size of the SSLs is not continuous over time and can enlarge abruptly, affecting the collaboration negatively. Alternatively, control barrier functions are considered in [23] to ensure safety without unnecessarily limiting the manipulator speed. In [24], the authors introduce a scheme for solving the performance/safety trade-off in collaborative workspace, which accounts for safety of a predefined set of points of interests on the robot based on the interaction/collision model of [25]. More recently, in [26], an integrated mixed-reality system for safety-aware human-robot collaboration based on a deep learning approach and on the generation of a digital twin is adopted to compute the minimum safety distance in real time. Furthermore, in [27] a computationally efficient control scheme for safe human-robot interaction is introduced. That approach is based on real-time constraints control and focuses on high dynamic environments where a manipulator moves in close proximity with a human operator and needs to avoid the unpredictable human motions.

Based on the references reported above, it emerges that in modern manufacturing, the design of collaborative robotic applications has not only to guarantee safety of the human operator, but also fluency in the collaboration, as well as performance in terms of productivity and task time [28]. Beyond considering only safety, it is therefore important to design schemes that allow the manipulator to react smartly, and achieve a smooth and successful collaboration between human and robot, otherwise the advantage of introducing a robot into the production line diminishes. A recent example can be found in [29], where efficiency in human-robot collaboration is enhanced by predicting the operator behavior with the hidden semi-Markov model in an assembly scenario.

In the context of human-robot collaboration, the performance of the coordinated meshing of shared activities between human and robot can be evaluated by means of collaborative fluency metrics. Indeed, when robots are introduced in a shared environment, it is suitable to evaluate how fluently robotic teammates could perform with their human counterparts. Collaborative fluency metrics have been introduced in [30], in the context of an anticipatory controller for shared-workspace decision processes. Subsequent works proposed several objective indexes to assess aspects of human-robot collaboration. In [31], human-robot cross-training was modeled and evaluated through different fluency metrics, such as *concurrent motion*, *human idle time*, *robot idle time*, and *human-robot distance*. The authors in [32] measured fluency in a coffee-making task, planned with different motion approaches, and applied objective metrics such as *coordination time*, *total task time*, and *concurrent motion time*. A codification of subjective and objective fluency metrics can be found in [33], where the author provides an analytical model of four objective metrics (*human idle time*, *robot idle time*, *concurrent activity*, and *functional delay*). Finally, in [34] a method for estimating online the quality of interaction between robot and operator is presented.

In this paper, we present an approach to guarantee safety of a human operator during the collaboration with a manipulator and, at the same time, enhance fluency and productivity in human-robot collaboration through online scaling of dynamic safety zones by building on [22, 35]. This paper includes novel contributions that allow us to: (i) extend the theoretical formulation accounting for the direction of the robot towards a collision to minimize the size of the safety zones enclosing the manipulator; (ii) compare the method with other available approaches with extensive simulations; (iii) demonstrate the practical benefit of the proposed approach by considering fluency metrics for human-robot collaboration; (iv) introduce a novel task performance index named *concurrent activity in the robot workspace* (C-ACT-WS); and (v) perform experiments using a 7-DOF manipulator on a real collaborative

scenario. To summarize, our proposed approach considers: (a) the online optimization of potential smooth stop trajectories to be traveled if a collision danger is identified; (b) the exploitation of the dynamics of the robot to minimize the stop time, and, consequently, the size of the dynamic safety zones enclosing the manipulator, while meeting joint torque limits, and accounting for the directed speed of the robot parts with respect to the human.

The paper is structured as follows: we describe in Sect. 2 the addressed problem, whereas the proposed approach is detailed in Sect. 3. The simulation results are illustrated in Sect. 4, and the experiments in Sect. 5. Finally, the conclusions are given in Sect. 6.

2 Problem statement

In this work, we address the problem of minimizing the size of the dynamic safety zones enclosing a manipulator, subjected to torque constraints, while performing an operation in coexistence with a human. As it will be shown below in Sect. 4, the minimization of the size of the dynamic safety zones results in a more fluent collaboration between human and robot, which, in turn, reduces the total task time of a given collaborative task and, therefore, increases productivity. During the collaborative task, the supervisory controller regularly checks online the collision danger between the manipulator links and the operator. In this regard, during robot motion, the robot system must never get closer to the operator than a protective separation distance S_p, which is computed online as in [16]. The contributions of S_p consider the space S_r traveled by the robot during the reaction time t_r, the robotic system stop distance S_s during the stop time t_s, the operator's change in location S_h:

$$S_p = S_h + S_r + S_s + \xi \tag{1}$$

where the term ξ includes several error sources, i.e., the measurement tolerance of the sensing system, the uncertainty in the robot position, and the portion of the human that can intrude the sensing field before being perceived. Disturbances and noise on the estimation of human position can also be included in the term ξ. As stated by the technical specification [16], the terms in (1) can be considered either as a constant worst-case value, or as a function of the current speed of the human or the manipulator. More in detail, the space S_{r_k} traveled by the k-th link of the robot arm from the initial time of the safety check t_0 (present or current time) to $t_0 + t_r$ is given by:

$$S_{r_k} = \int_{t_0}^{t_0+t_r} v_{r_k}(t) \, dt \tag{2}$$

where v_{r_k} is the speed of the k-th link during t_r. Once t_s is defined, as it will be described below in Sect. 3, the distance S_{s_k} traveled by the k-th link to come to stop is calculated as:
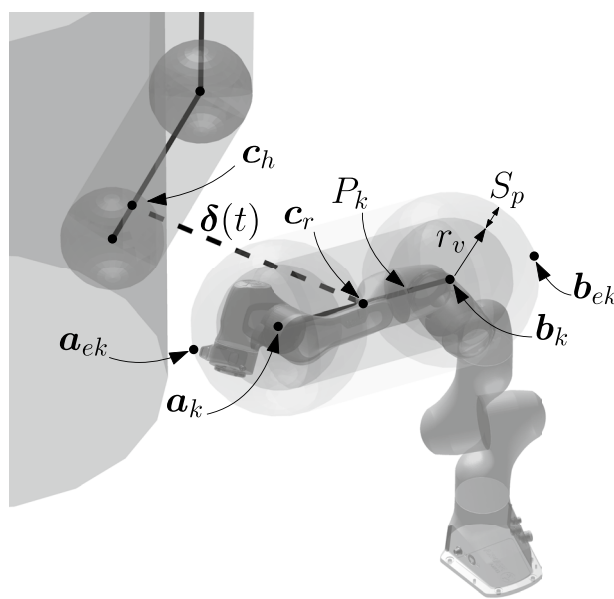

Fig. 1 The k-th part of a manipulator enclosed in a SSL and in its corresponding safety zone (light gray). On the left, a human bounding volume

$$S_{s_k} = \int_{t_0+t_r}^{t_0+t_r+t_s} v_{\delta_k}(t) \, dt \tag{3}$$

being v_{δ_k} the speed of the k-th link being stopped, directed on the minimum-distance line segment with respect to the human. Finally, the term S_h can be computed only once by considering a worst case, as:

$$S_h = v_h(t_r + t_s) \tag{4}$$

where $v_h = 1.6 \ m/s$ is the maximum velocity that may be considered for the human by the ISO/TS 15066 [16].

In this work, we choose SSLs as bounding volumes surrounding both robot and human to verify possible unwanted contacts between them. SSLs are given by the Minkowski sum of a sphere S and a line segment P. The SSL for the k-th link is defined by the end-points $\boldsymbol{a}_k$ and $\boldsymbol{b}_k$ of segment P_k, and the radius r_v, which is the minimum radius needed to enclose the geometry of the link, as shown in Fig. 1.

We define safety zones as SSLs that also account for the protective separation distance S_p, and that over-approximate the space the robot needs to stop starting from its current kinematic state. Accordingly, the radius r_{SZ} of a SSL that represents a safety zone surrounding a link is given by:

$$r_{SZ} = r_v + S_p. \tag{5}$$

After the calculation of r_{SZ}, the possible collision between each safety zone of the robot and the SSLs enclosing the human is verified. For this purpose, the distance between
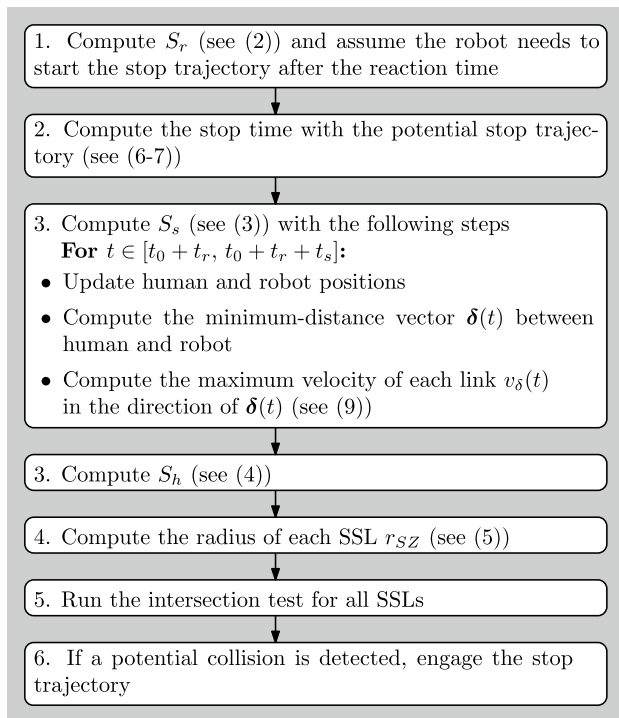
1. Compute S_r (see (2)) and assume the robot needs to start the stop trajectory after the reaction time

2. Compute the stop time with the potential stop trajectory (see (6-7))

3. Compute S_s (see (3)) with the following steps
   For $t \in [t_0 + t_r, t_0 + t_r + t_s]$:
   - Update human and robot positions
   - Compute the minimum-distance vector $\delta(t)$ between human and robot
   - Compute the maximum velocity of each link $v_\delta(t)$ in the direction of $\delta(t)$ (see (9))

3. Compute S_h (see (4))

4. Compute the radius of each SSL r_{SZ} (see (5))

5. Run the intersection test for all SSLs

6. If a potential collision is detected, engage the stop trajectory

Fig. 2 Overview of the proposed approach

each couple of line segments is computed online, and checked against the radii of the two spheres corresponding to the closest points on the line segments of the human and the robot. The SSM criterion is satisfied, if the safety distance between human and robot is guaranteed, or if the manipulator is stopped.

3 Proposed approach

In this work, the optimization of the size of the safety zones around the robot is based on the online minimization of the stop time t_s during which the manipulator is steered to stop without exceeding its torque limits nor activating the mechanical brakes. The core of the approach is the online optimization of t_s that aims at computing the stop trajectory that makes the safety regions around the robot as small as possible. We compare the proposed approach with a direct implementation of the technical specification ISO/TS 15066 [16], which guarantees safety through static safety zones, and with the method in [22].

Figure 2 reports a graphical diagram of the approach presented in this work. As it can be seen from the figure, the contribution S_r is first computed using (2) and the nominal trajectory from the current time t_0 to $t_0 + t_r$. We assume that it will be required to stop the manipulator right after the reaction time of the supervisory controller. Therefore, the potential stop trajectory defined from $t_0 + t_r$ to $t_0 + t_r + t_s$ is planned by

optimizing the stop time t_s and considering the joint torque limits of the robot as nonlinear constraints. After the definition of t_s, the algorithm computes the term S_s by considering the minimum-distance vector $\delta(t)$, as explained below. After that, the contribution S_h and the radius of each safety zone enclosing the manipulator are computed. It has to be noted that the contribution S_h includes both the space covered by the human during the stop time, and the distance traveled during the reaction time, which represents the receding horizon of the collision avoidance approach.

Finally, intersection tests are run for all the safety zones of the robot against the SSLs enclosing the human. If a potential collision is identified, the stop trajectory is engaged right after the reaction time. Otherwise, the robot keeps tracking the nominal trajectory. Our method follows the indications of the ISO/TS 15066 [16] for handling the risk of collision when the robot moves. On the other hand, the technical specification does not consider collisions when the robot stops. In the following, we describe the proposed approach for optimizing the stop trajectory online and computing the contribution S_s.

We consider the stop trajectory $\boldsymbol{q}_s(t)$ as a five-degree polynomial planned from the current joint variables at time $t_0 + t_r$, i.e., initial position $\boldsymbol{q}_{s,i}$, velocity $\dot{\boldsymbol{q}}_{s,i}$, and acceleration $\ddot{\boldsymbol{q}}_{s,i}$, to the final joint state, chosen as $\boldsymbol{q}_{s,f} = \boldsymbol{q}_{s,i}$, $\dot{\boldsymbol{q}}_{s,f} = 0$, and $\ddot{\boldsymbol{q}}_{s,f} = 0$. Five-degree polynomial trajectories allow us to steer the robot from the current kinematic state to the safe stop with smooth acceleration and limited jerk. Please note that the stop trajectory needs to be computed in discrete values in a real implementation.

The optimal stop trajectory $\boldsymbol{q}_s(t)$ is computed by means of a numerical optimization, which is solved online within the reaction time t_r. The optimization problem is defined as:

$$\min_{t_s} \quad w_0 \, t_s + w_1 \, |t_s - t_{s,prev}| \tag{6}$$

subject to

$$|\tau_i(\boldsymbol{q}_s(t), \dot{\boldsymbol{q}}_s(t), \ddot{\boldsymbol{q}}_s(t))| \le \tau_{i,max}$$
$$t \in [t_0 + t_r, \, t_0 + t_r + t_s], \quad i = 1, ..., N \tag{7}$$

where w_0 and w_1 are positive weights, $t_{s,prev}$ is the stop time found at the previous iteration, τ_i and $\tau_{i,max}$ are the torque at the i-th joint and the maximum torque at the same joint, respectively, and N is the number of the robot DOFs. The term $w_1 \, |t_s - t_{s,prev}|$ allows us to penalize large discontinuities between successive iterations that would result in sudden widening of the safety zones. If no available solution of (6) with (7) is found within t_r, the worst-case stop time $t_{s,wc}$ is selected, which can be obtained by considering the robot specification. $t_{s,wc}$ also initializes $t_{s,prev}$ at the beginning.

The dynamics of the robot is considered to verify the torque values during the stop trajectory. More in detail, the torques $\boldsymbol{\tau}$ during $\boldsymbol{q}_s(t)$ can be computed as follows:

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q}_s)\ddot{\boldsymbol{q}}_s + \boldsymbol{C}(\boldsymbol{q}_s, \dot{\boldsymbol{q}}_s)\dot{\boldsymbol{q}}_s + \boldsymbol{F}_v\dot{\boldsymbol{q}}_s + \boldsymbol{f}_c\mathrm{sign}(\dot{\boldsymbol{q}}_s) + \boldsymbol{g}(\boldsymbol{q}_s) \quad (8)$$

where $\boldsymbol{M}(\boldsymbol{q}_s)$ represents the mass matrix of the robot, and $\boldsymbol{C}(\boldsymbol{q}_s, \dot{\boldsymbol{q}}_s)\dot{\boldsymbol{q}}_s$ considers the Coriolis and centrifugal terms. $\boldsymbol{F}_v$ and $\boldsymbol{f}_c$ are the viscous and Coulomb friction terms, respectively, whereas $\boldsymbol{g}(\boldsymbol{q}_s)$ is the gravity vector. In this work, the dynamic model of the manipulator is considered to be well known[1]. However, if the available robot dynamics is uncertain or inaccurate, an alternative approach that allows one to compute online forces and torques bounds from uncertain dynamics can be adopted, e.g., using a recursive Newton-Euler algorithm based on interval arithmetic, as in [36]. In that work, it is shown that even with uncertainties, a resolution of one step of inverse dynamics is in the order of 10^{-5} s on a standard computer.

The distance S_{s_k} traveled by the k-th link of the robot being stopped is computed as in (3), where we account for the maximum linear velocity of a link directed in the direction of the vector $\boldsymbol{\delta}(t)$, representing the minimum-distance line segment between human and robot over the stop time. Since the proposed approach is based on the intersection of bounding volumes considered as capsules, the minimum distance between human and robot is computed between each possible couple of line segments belonging to the human and the robot, as described in [37]. The algorithm for computing the distance of two line segments is efficient and suitable for online computation, since it is based on an analytical formulation. Figure 1 shows that $\boldsymbol{c}_h$ and $\boldsymbol{c}_r$ are the closest points on the generic line segments of the human and the robot, respectively. During the stop trajectory, the robot is being stopped, and also the human is considered to be in motion. Therefore, both $\boldsymbol{c}_h$ and $\boldsymbol{c}_r$, and the direction of $\boldsymbol{\delta}(t)$ change over the stop time, as shown in Fig. 3. During that time interval, i.e., when $t \in [t_0 + t_r, t_0 + t_r + t_s]$, the position of the human is updated as if it is moving at the full speed v_h towards the manipulator along $\boldsymbol{\delta}(t)$.

The maximum linear velocity $v_{\delta_k}(t)$ of a link directed in the direction of $\boldsymbol{\delta}(t)$ is obtained by considering the velocity of the spherical end-caps $\dot{\boldsymbol{a}}_{e_k}$ and $\dot{\boldsymbol{b}}_{e_k}$. As shown in Fig. 1, $\boldsymbol{a}_{e_k}$ and $\boldsymbol{b}_{e_k}$ are the extreme points of the spherical end-caps that over-approximate the k-th link of the robot. To compute $\boldsymbol{a}_{e_k}$ and $\boldsymbol{b}_{e_k}$, the potential stop trajectory $\boldsymbol{q}_s(t)$ and its velocity $\dot{\boldsymbol{q}}_s(t)$ are calculated for $t \in [t_0 + t_r, t_0 + t_r + t_s]$. The maximum linear velocity v_{δ_k} of a link, directed on the minimum-distance line segment with respect to the human, for the same time interval t is given by:

$$v_{\delta_k}(t) = \max(|\dot{\boldsymbol{a}}_{e_k}(t) \cdot \boldsymbol{\delta}(t)|, |\dot{\boldsymbol{b}}_{e_k}(t) \cdot \boldsymbol{\delta}(t)|) \quad (9)$$

---

[1] Please note that a kinematics and dynamics model description of the robot to be used within a recursive Newton-Euler algorithm for efficient numerical computations is an equivalent requirement.
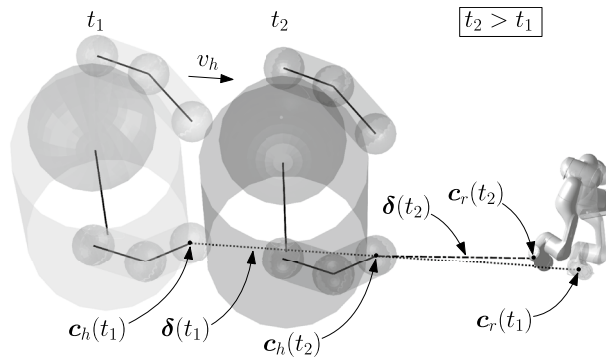

Fig. 3 Updating of the human position towards the robot along $\boldsymbol{\delta}(t)$ during one step of the stop trajectory. The lighter gray color refers to a previous time instant of robot and human. For clarity, the robot motion and the human shift are amplified

where the velocities of $\boldsymbol{a}_{e_k}$ and $\boldsymbol{b}_{e_k}$ are computed using forward differential kinematics, as in [35].

4 Comparison results

In this section, the simulated test cases and the fluency metrics for human-robot collaboration are first described. Then, the simulation test bed is illustrated. Finally, the practical benefit of the proposed approach is demonstrated through the comparison results.

4.1 Simulated test cases

To evaluate the feasibility of the proposed method, the following approaches to plan the potential stop trajectory within the reaction time t_r are implemented and compared:

- Approach (1): online scaling of dynamic safety zones;
- Approach (2): a method based on the selection of the best stop trajectory among n tentative stop times, as in [22];
- Approach (3): an implementation of the technical specification ISO/TS 15066 [16], resulting in static safety zones.

Approach (1) consists of the optimization scheme described in Sect. 3, which is the strategy proposed in this paper. Approach (2) implements the strategy that selects the best smooth stop trajectory by considering n linearly spaced tentative stop times $\boldsymbol{t}_s = [t_{s,1}, ..., t_{s,n}]$, with $t_{s,j} < t_{s,j+1}$ for $j = 1, ..., n$, and $t_{s,n} \leq t_{s,wc}$, as in [22]. The number of tentative stop times is limited to $n = 10$, to cope with the available reaction time t_r. Each stop trajectory is planned within t_r, and the first feasible trajectory that meets the torque limits $\boldsymbol{\tau}_{max}$ is chosen. Even in this case, five-degree polynomial motion laws are adopted to guarantee smoothness

and limited jerk. Such a fall-back variant with respect to the proposed approach, however, is more conservative since it does not allow the robot to exploit the full range of available torques and may lead to safety zones that can enlarge abruptly over time. Finally, Approach (3) corresponds to a simple implementation of the technical specification ISO/TS 15066 [16].

4.2 Fluency metrics for human-robot collaboration

The practical benefit of the proposed method is shown by considering fluency metrics for human-robot collaboration. We adopt from [33] the objective metrics of *total task time* and *robot idle time*. Furthermore, a novel task performance index named *concurrent activity in the robot workspace* is introduced in this paper. We also account for the *number of robot stops* during the prescribed task. These metrics quantitatively estimate the degree of fluency in a given human-robot interaction, and are defined as follows:

- *Total task time* (T-TIME): this metric consists of the total time needed for the robot to complete its motion by keeping possible safety stops and restarts into account. T-TIME is directly related to productivity.
- *Robot idle time* (R-IDLE): this measure corresponds to the percentage of the total task time during which the robot is not perceivably active. The inactivity of the robot is measured as the time during which the norm of the joint velocities is less than 10^{-3}. Therefore, R-IDLE corresponds to the time during which the robot is being stopped or waits for a safe restart. Low values of R-IDLE indicate an efficient use of the robot and reflect positively on the collaborative fluency. In our simulated test case, described in Sect. 4.3, where the human is always active, R-IDLE results the percentage complement of the concurrent activity (C-ACT), defined as the percentage of the time out of the total task time during which robot and human are concurrently active [33].
- *Concurrent activity in the robot workspace* (C-ACT-WS): this novel measure is related to the percentage of the time during which robot and human are concurrently active *inside the robot workspace*, out of the time the human enters the robot workspace at least partially.
- *Number of robot stops* (R-STOPS): this measure accounts for the total number of times the robot is stopped by the supervisory controller during the prescribed task.

The novel metric C-ACT-WS has been introduced in this paper since C-ACT did not highlight the presence of the human in the robot workspace. If there were sporadic presences with most of the task performed outside the workspace, C-ACT could hide impact on the fluency perceived by the operator for the intervals of time she/he collaborates

very close to the robot. High levels of C-ACT-WS can be related to a high sense of fluency and work balance in the collaborative task.

4.3 Simulation test bed

The proposed approach is validated on the 7-DOF Panda robot by Franka Emika GmbH and adopting the dynamic model identified in [38]. The simulated test bed is implemented in Simulink®, using a laptop running Windows 10 Pro with an Intel i7-8565U CPU and 16 GB of RAM. The optimization problem (6) with (7) is solved online using the constrained nonlinear Matlab^{TM} function *fmincon* and the sequential quadratic programming iterative scheme. We set to 10 the maximum number of iterations to meet real-time requirements. If the optimization function does not find a feasible solution within the reaction time t_r, the worst-case scenario stop time $t_{s,wc}$ is selected. We consider $t_r = 5$ *ms*, and a sampling time of 1 *ms* for the dynamic model of the manipulator and its tracking controller.

For each of the three compared approaches 100 simulations are run. In these tests, the supervisory controller verifies online the separation distance between the robot performing random trajectories and a human moving around the manipulator and cyclically entering its workspace. For each simulation, five random points are defined in the joint space of the robot, and a five-degree polynomial trajectory $q_d(t)$ is planned between each couple of way points. The starting and ending velocities and accelerations are imposed to be equal to zero in each $q_d(t)$. To evaluate the method in challenging conditions for the robot, the sequence of points is chosen in order to bring at least one joint to maximum velocity, while meeting the torque limits τ_{max} of the manipulator. The travel time to reach each way point is set equal to $t_{wp} = 2$ *s*, leading to a nominal total time of 10 *s*, without considering the possibility of stopping due to human intrusion. In case a safety stop of the robot occurs, a restart trajectory is engaged to steer the robot from the current kinematic state to the next way point. The restart trajectory is defined as a joint-space five-degree polynomial motion law with zero initial and final velocities and accelerations. The total time for the restart trajectory is computed analytically as the maximum between the time needed for that motion law to steer at least one joint to maximum velocity, and the time needed to steer at least one joint to maximum acceleration.

We choose four bounding volumes to enclose all the links and joints of the considered 7-DOF manipulator. However, choosing four bounding volumes instead of seven represents a reasonable approximation to speed up online calculations without introducing excessive conservativeness in the case considered in this work. Furthermore, a self-collision avoidance algorithm is implemented based on the same four bounding volumes that tightly over-approximate the robot.
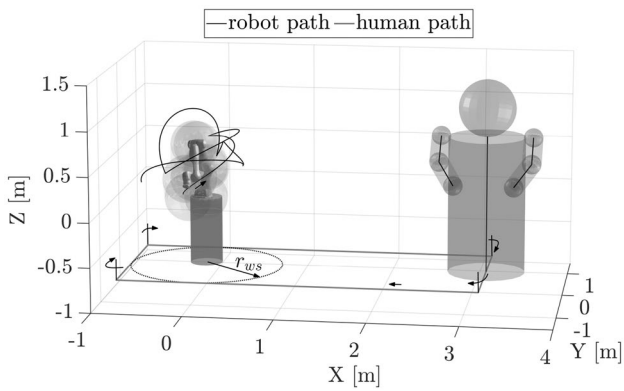
Fig. 4 Simulation test bed. The path of the robot end effector corresponds to the test trajectory of Fig. 7

The evaluation of the proposed method is carried out first of all with simulated humans, i.e., by realizing a deterministic and replicable motion model for the human that brings its speed to the limits that may be considered according to [16]. In this manner, different approaches for safe human-robot interaction can be evaluated and compared using exactly the same human motion over time. The simulated human is modeled as a simplified kinematic tree with a chest, a head, and two arms with 4-DOF, as shown in Fig. 4. For the sake of simplicity, the wrist and hand movement have been excluded, as in [39]. Six bounding volumes have been defined to encapsulate the human, whose dimensions are chosen according to the anthropometric estimates of the 95-th percentile of British adults aged 19–65 years [40]. In this way, at each iteration of the supervisory controller, the minimum distance between human and robot is found online as the minimum of the 24 distances between the six human line segments and the four of the robot.

The human is moved along an arbitrarily chosen rectangular path with a length of 4.00 m and a width of $2\,r_{ws}$, where $r_{ws} = 0.855\ m$ is the maximum horizontal reachability of the arm from the robot base reference frame. The geometry of that path is chosen in order to simulate cyclical intrusions of the human in the workspace of the robot during the random motion of the manipulator. In this way, the compared approaches are tested with a high number of random

intrusions in which the relative motion between human and robot is not defined a priori. At each corner of the path, the human model rotates $\pi/2\ rad$. The motion law for the human is defined by means of four consecutive trapezoidal speed profiles that bring its speed v_h to 1.6 m/s. The total time for the human to complete the rectangular path is set equal to 10 s. However, in the numerical tests where at least one safety stop of the robot occurs, the human keeps moving on its path and cyclically entering the robot workspace until the robot end effector has not reached the last way point.

Figure 5 shows the architecture of the robot supervisory controller. In the figure it can be seen that the hypothesis on the basis of this work considers the knowledge of the desired trajectory prescribed to the robot, the robot kinematic and dynamic parameters, its torque limits, and the human position and bounding volumes. In the block diagram, the intersection test verifies if the desired trajectory q_d should be traveled by the manipulator ($\beta = 1$), or if the optimal stop trajectory q_s should be engaged ($\beta = 0$). Finally, the tracking controller supplies the robot with the torques values τ for all the seven joints on the basis of the measured joint positions q_m and velocities $\dot{q}_m$.

4.4 Results of comparative simulations

Figure 6 reports exemplary frames of one of the 100 performed simulations with the three compared approaches in which the human cyclically intrudes in the robot workspace, while the manipulator performs random trajectories. Frames with shaded background denote safety stops of the robot. From Fig. 6(a), it appears evident that the sizes of the safety zones in Approach (1) are smaller than those in Approach (2), and further increase in size if Approach (3) is considered. For that exemplary simulation (Fig. 6(d)), it can be seen that the T-TIME is equal to 13.37 s for Approach (1) (only 3.37 s late with respect to the nominal robot task time of 10 s), to 18.19 s for Approach (2), and to 28.81 s for Approach (3), due to the increasing time the robot is stopped by the supervisory controller scrolling through the three approaches. To better appreciate the online scaling of the dynamic safety zones, a video of the robot performing an exemplary trajectory is
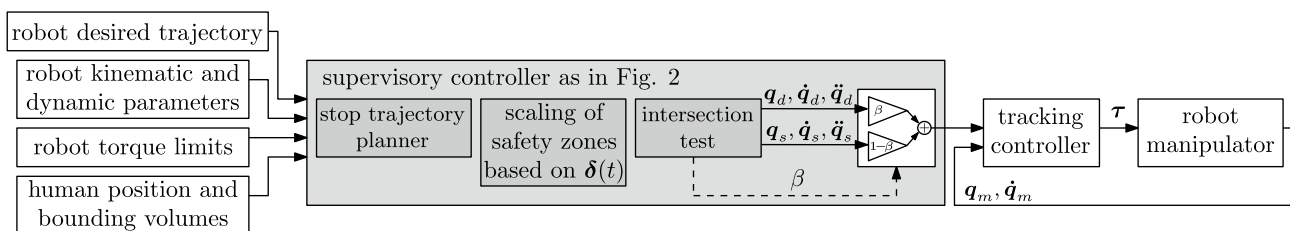
Fig. 5 Block diagram of the robot controller. The blocks in gray run at the supervisory controller rate, the blocks in white at the tracking controller rate
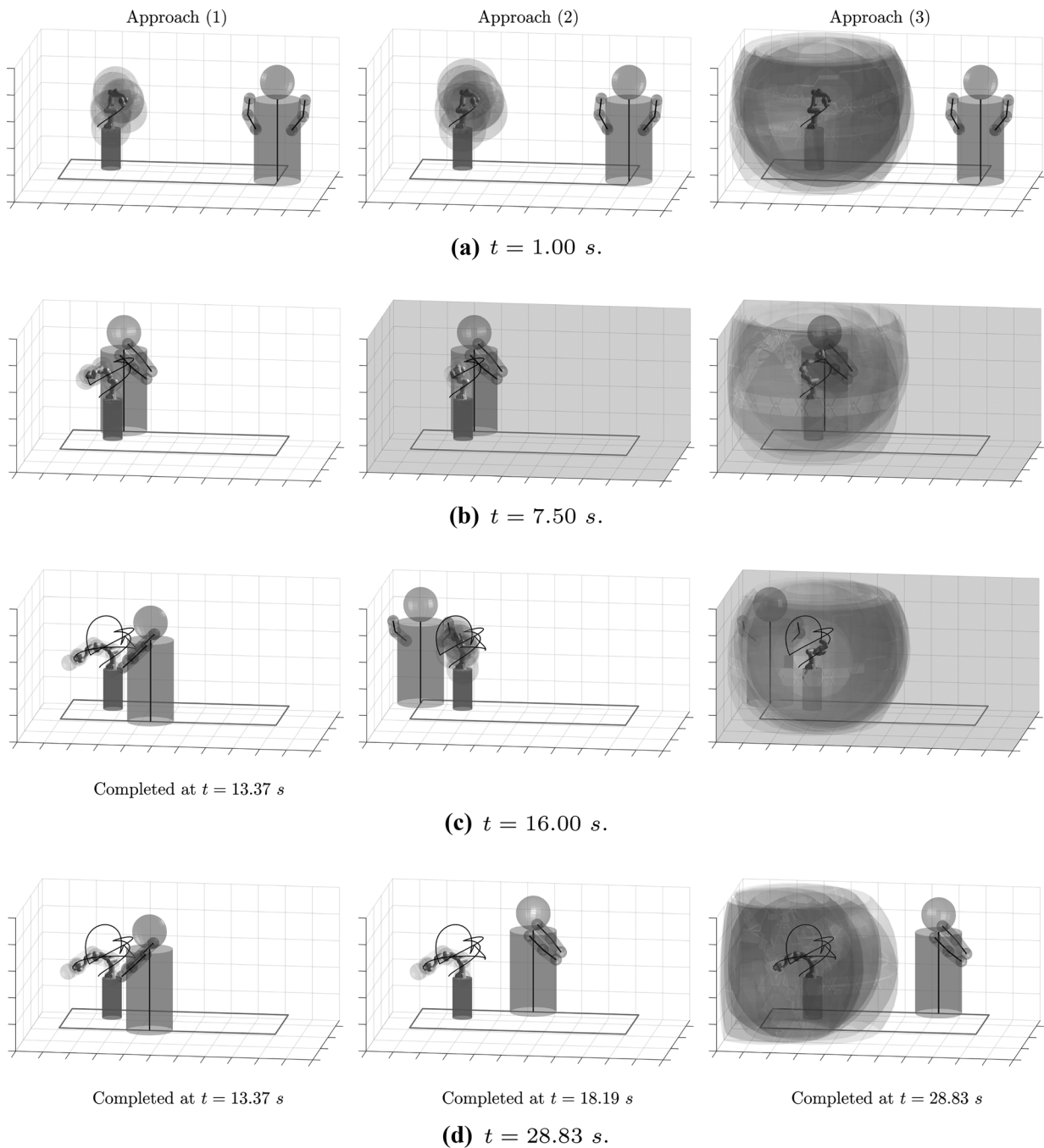
Fig. 6 Frames of sample simulations with the three compared approaches in which the human cyclically intrudes in the robot workspace, while the manipulator performs random trajectories. Frames with shaded background denote safety stops of the robot

available as Supplementary Material. From the video, it can be seen that, differently from Approach (1), the safety zones computed with Approach (2) are discontinuous over time, whereas those computed with Approach (3) show fixed values of the radii over time. It results clear that the proposed strategy based on the online scaling of dynamic safety zones enhances the performance of human-robot collaboration with respect to the compared approaches.

Figure 7 reports the example test trajectory performed by the robot manipulator using Approach (1) in the same simulation of Fig. 6 (frames on the left column). During the test, the potential stop trajectory is computed by the supervisory controller by scaling online the dynamic safety zones surrounding the robot. The plots show the joint positions, velocities, torques, radii of the safety zones, and stop time over the task time. The vertical solid lines in black
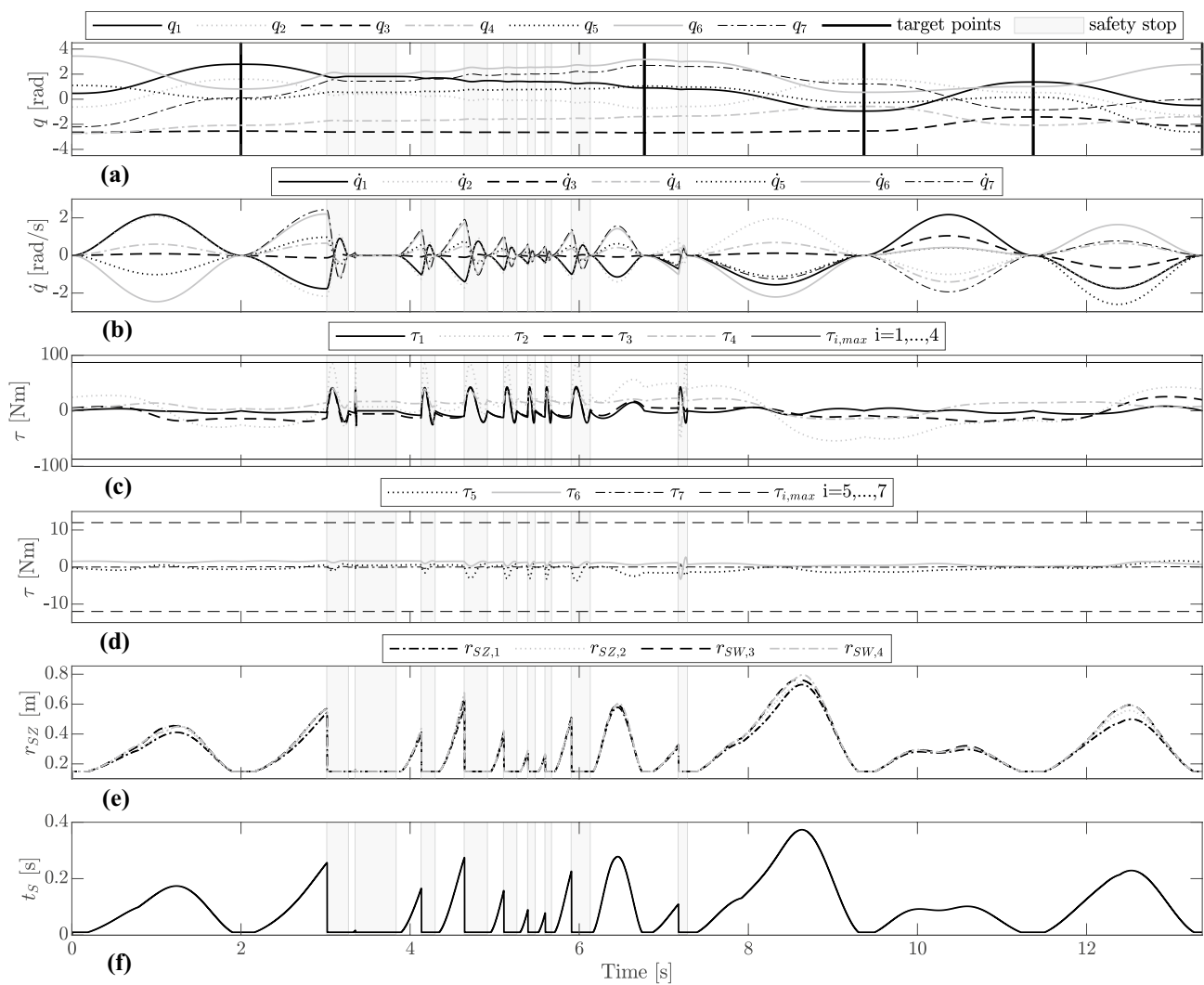
Fig. 7 Example trajectory: joint positions (**a**); velocities (**b**); torques (**c**),(**d**); radii of the safety zones (**e**); and stop time (**f**) computed using Approach (1). Vertical solid lines in black indicate target achievements, whereas shaded areas denote safety stops of the robot

indicate the five target achievements chosen randomly in the robot joint space (Table 1), as described in Sect. 4.3. The shaded areas indicate the time intervals during which a safety stop occurs. From Fig. 7(e) it can be seen that the radii of the dynamic safety zones change according to the dynamic scaling performed online with the proposed strategy. Furthermore, when a potential collision is identified, the

robot engages a trajectory that steers the robot to stop while respecting the joint torque limits, as it is shown in Fig. 7(c) and (d). From Fig. 7(e), it can be seen that the safety zones r_{SZ} occupy a restricted portion of the robot workspace, since they show values of the radius lower than the maximum horizontal reachability of the arm $r_{ws} = 0.855\ m$ during the whole trajectory. Therefore, a human can safely enter the
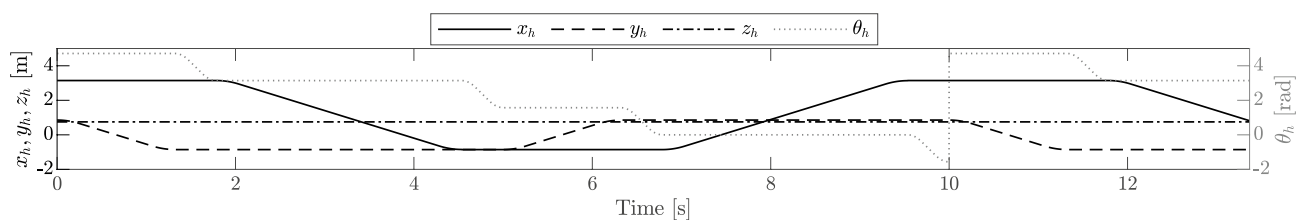
Fig. 8 Human position variables $(x_h, y_h, z_h, \theta_h)$ with respect to the robot base reference frame for the simulation of Fig. 7

Table 1 Random joint positions [rad] for the initial robot configuration $q_{d,0}$ and the five way points $q_{d,i}$, $i = 1, ..., 5$, defined for the example test trajectory of Fig. 7

Joint	1	2	3	4	5	6	7
$q_{d,0}$	0.469	-0.619	-2.637	-2.721	1.101	3.440	-2.208
$q_{d,1}$	2.788	1.602	-2.541	-2.080	0.014	0.817	0.103
$q_{d,2}$	0.903	-0.713	-2.674	-1.387	1.050	3.167	2.698
$q_{d,3}$	-0.945	1.604	-2.531	-0.573	-0.272	0.554	1.219
$q_{d,4}$	1.372	0.532	-1.414	-2.074	0.161	1.008	-0.846
$q_{d,5}$	-0.496	-1.288	-2.126	-1.390	-2.615	2.759	-0.022

workspace of the robot during its operation. This is also testified by the novel fluency metric C-ACT-WS introduced in this paper, as it is described below.

Figure 8 reports the human position variables with respect to the robot base reference frame over the task time for the example simulation of Fig. 7. The human keeps moving on its path and cyclically entering the robot workspace until the manipulator has reached the last way point. Figure 9 shows five frames of the robot traveling the test trajectory of Fig. 7 and the dynamic safety zones computed with Approach (1) for $t = 0.00, 0.50, 1.00, 1.50,$ and 2.00 s.

Figure 10 shows the box plot representation of the simulation results, where the three approaches to compute the potential stop trajectory within the reaction time are compared. In the box plots, the central mark indicates the median, the bottom and top of each box represent the first and third quartiles, whereas the whiskers extend to the most extreme data not considered outliers. For each of the 100 simulations, the root mean square values for the stop time and the radii of the safety zones are computed and reported in aggregate form in Fig. 10(a) and (b). In this figure, Approach (3) is not shown since it presents fixed values equal to 0.40 s and 1.583 m, for the stop time and the radii of the safety zones, respectively. The box plot results for the fluency metrics T-TIME, R-IDLE, C-ACT-WS and R-STOPS are shown in Fig. 10(c)–(f). Furthermore, Table 2 reports the quantitative metrics for stop time, radii of the safety zones, and fluency metrics for human-robot collaboration, as mean, standard deviation, minimum and maximum values across the 100 simulations for each of the three compared approaches.

Fig. 9 Frames of the robot performing the test trajectory of Fig. 7, and dynamic safety zones computed with Approach (1)

(a) $t = 0.00$ s.

(b) $t = 0.50$ s.

(c) $t = 1.00$ s.

(d) $t = 1.50$ s.

(e) $t = 2.00$ s.
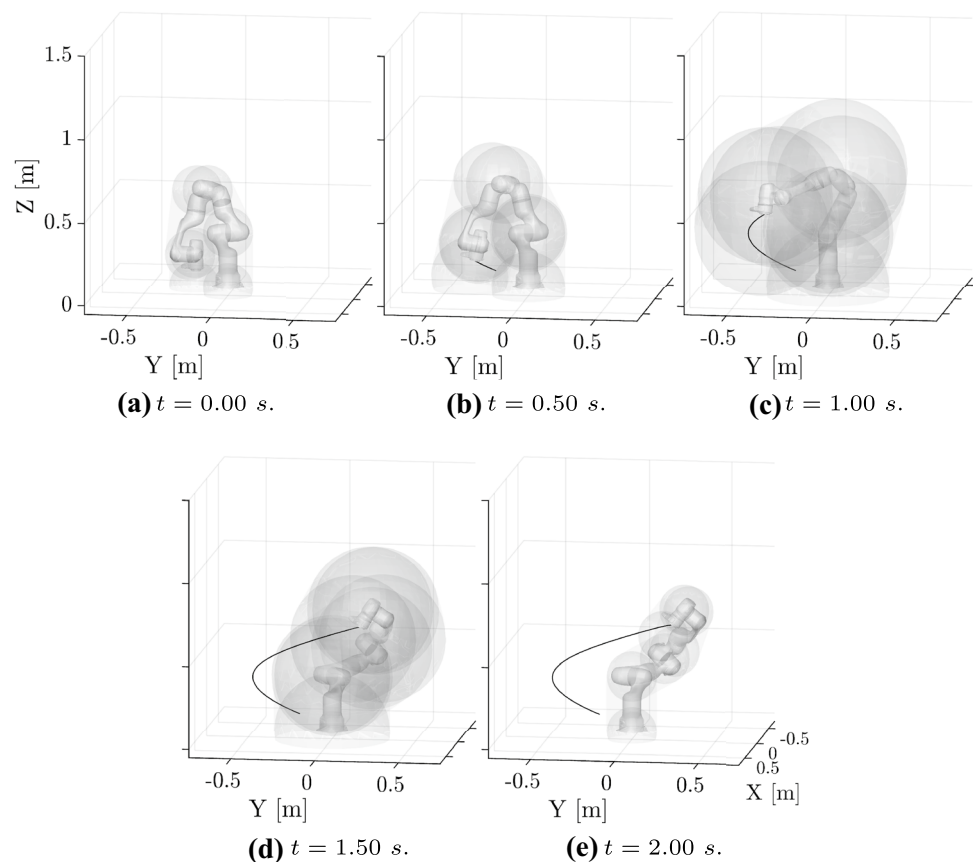
Fig. 10 Box plot representation of the comparison results: stop time (**a**); radii of the safety zones (**b**); and fluency metrics for human-robot collaboration (T-TIME (**c**); R-IDLE (**d**); C-ACT-WS (**e**); R-STOPS (**f**)). In (**a**) and (**b**), Approach (3) is not shown since it presents fixed values equal to 0.40 *s* and 1.583 *m*, for the stop time and the radii of the safety zones, respectively
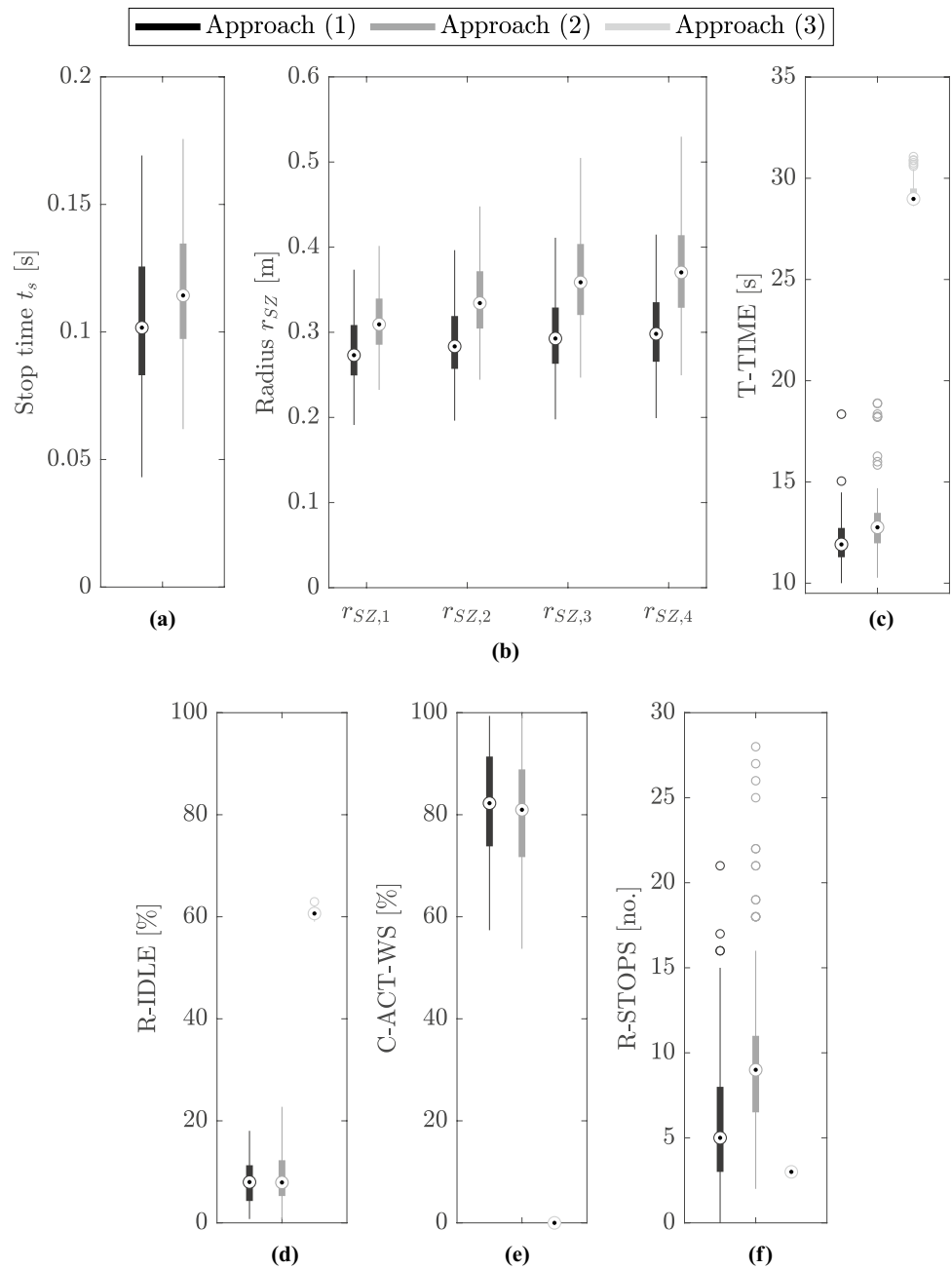
From Fig. 10 and Table 2, it can be seen that the values of stop time and radius of the safety zones are considerably lower for Approaches (1) and (2) with respect to the basic implementation of the technical specification ISO/TS 15066, which results in static safety zones. Furthermore, the optimal scaling of dynamic safety zones implemented in Approach (1), which also accounts for the direction of the robot towards a collision, further reduces both the time required by the manipulator to come to a safe stop, as well as the size of the safety zones, with respect to Approach (2).

The results of the fluency metrics shown in Fig. 10(c)–(f) allow us to further analyze the comparison results. The *total task time* for both Approaches (1) and (2) is considerably lower than Approach (3). In particular, the robot is able to complete the task faster with Approach (1) with respect to Approach (2), showing that the optimal scaling of safety zones we propose reduces the total time required by the robot to reach the five prescribed way points. R-IDLE is lower for Approach (1), indicating that the robot is inactive for a lower percentage of the total task time. The values for this metric rise slightly

Table 2 Quantitative metrics for stop time, radii of the safety zones, and fluency metrics for human-robot collaboration computed with the three compared approaches. Values are reported as mean ± standard deviation (minimum, maximum)

Approach	t_s [s]	$r_{SZ,1}$ [m]	$r_{SZ,2}$ [m]	$r_{SZ,3}$ [m]	$r_{SZ,4}$ [m]	T-TIME [s]	R-IDLE [%]	C-ACT-WS [%]	R-STOPS [no.]
(1)	0.104 ± 0.028	0.279 ± 0.042	0.289 ± 0.046	0.298 ± 0.048	0.302 ± 0.050	12.052 ± 1.234	8.0 ± 4.8	81.8 ± 11.9	5.9 ± 3.9
	(0.043, 0.169)	(0.191, 0.374)	(0.196, 0.397)	(0.198, 0.411)	(0.199, 0.415)	(10, 18.347)	(0.8, 18.0)	(57.4, 99.4)	(0, 21)
(2)	0.118 ± 0.026	0.313 ± 0.039	0.339 ± 0.049	0.366 ± 0.060	0.378 ± 0.064	13.077 ± 1.736	8.7 ± 5.1	80.4 ± 11.9	10.0 ± 5.6
	(0.062, 0.176)	(0.232, 0.401)	(0.244, 0.448)	(0.247, 0.505)	(0.250, 0.530)	(10.269, 18.892)	(0.7, 22.7)	(53.7, 99.4)	(2, 28)
(3)	0.400	1.583	1.583	1.583	1.583	29.209 ± 0.586	60.6 ± 0.8	0	3
						(28.624, 31.069)	(59.0, 62.9)		

for Approach (2), and considerably for Approach (3), with whom R-IDLE is equal to the 60.6% of the total task time. Furthermore, the *concurrent activity in the robot workspace* indicates that with our proposed approach (Approach (1)) robot and human can be concurrently active for up to the 81.8% of the time the human enters the robot workspace at least partially. C-ACT-WS assumes a mean value of 80.4% with Approach (2). On the contrary, no concurrent activity is possible if an approach based on static safety zones is implemented, since the radii of the safety zones enclosing the robot would be always greater than r_{ws}. Finally, in Fig. 10(f) the *number of robot stops* during the prescribed task is shown. With Approach (3) the robot is stopped three times in all the simulations, since it can be active only when the human is moving outside the robot workspace on the farthest side of its rectangular path, and it is stopped for all the remainder time. This results in T-TIME with a mean value equal to approximately 29 *s*, i.e., the human moves around the robot for almost three cycles before the robot can complete its task. On the contrary, with Approach (2) the mean *number of robot stops* is equal to 10.0. This value is further reduced to 5.9 if Approach (1) is considered.

Finally, Fig. 11 reports the box plot representation of the comparison results showing the maximum torques of the robot during a stop for the three approaches. The joint torque limits equal to 87 and 12 *Nm*, for the first four joints and for the last three, respectively, are always respected during the numerical simulations. The maximum available torques resulting with Approach (1) are closer to the limits with respect to the other two comparison approaches, especially for the first and the second joints. This shows that the optimization problem solved online in (6) with (7) allows the robot to better exploit the full range of available torques during a safety stop.

5 Experiments

To verify the effectiveness of the proposed approach on a real collaborative scenario, the method is implemented on a 7-DOF Franka Emika manipulator, available at the Laboratory for Artificial Intelligence for Human-Robot Collaboration (AI4HRC) at the University of Udine (Italy). The robot has a reachability of 855 *mm* and can hold a payload of 3 *Kg*. A workstation running Ubuntu 18.04 LTS Bionic Beaver with an Intel i5-10600k and 32 GB of RAM has been adopted. The robot is controlled using ROS Melodic Morenia (Robot Operating System) with Python.

To track the position of the human online, we adopt a Intel® RealSense™ Depth Camera D435. The camera has a depth field of view of 87° × 58°, a depth output resolution
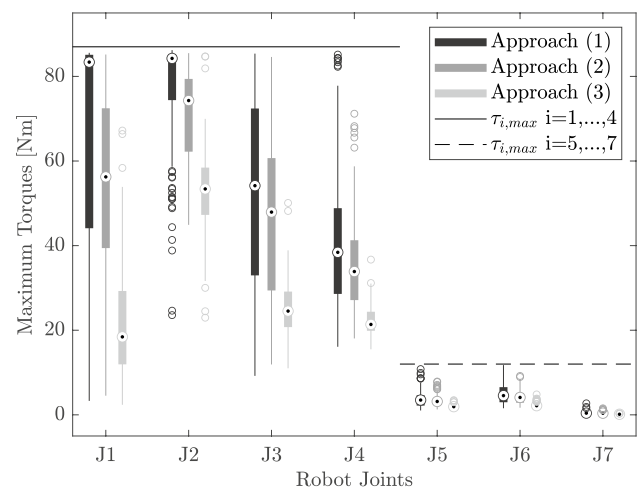
Fig. 11 Box plot representation of the comparison results showing the maximum torques of the robot during a stop

**Fig. 12** Frames of an example experiment. The corresponding robot trajectory is shown in Fig. 13

(a) $t = 0.5\ s$.

(b) $t = 5.0\ s$.

(c) $t = 10.0\ s$.

(d) $t = 12.5\ s$.

(e) $t = 15.0\ s$.

(f) $t = 20.0\ s$.
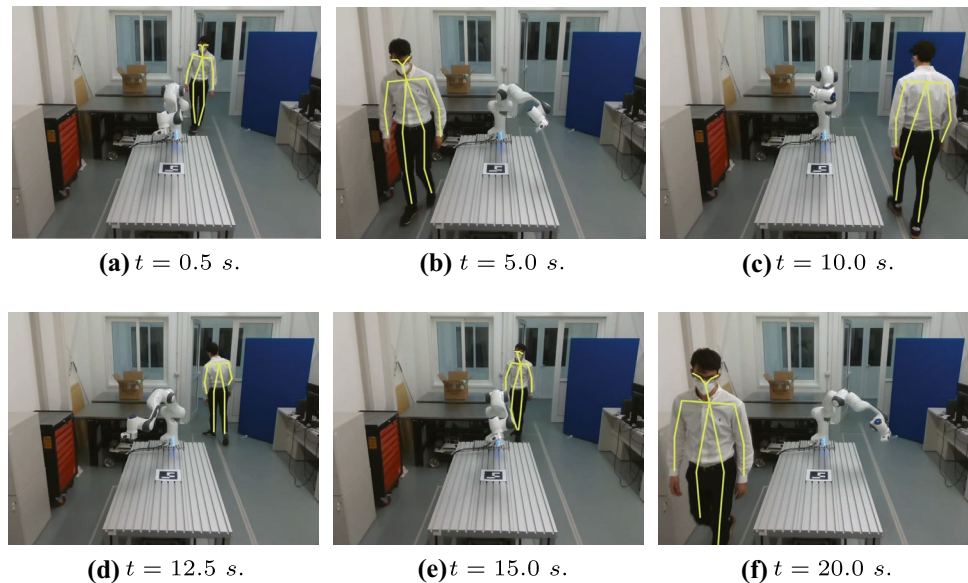
up to 1280×720, a depth frame rate up to 90 Hz, and a global shutter on the depth sensor, ideal for fast-moving applications. The CubeMOS Skeleton Tracking SDK is used to identify and simultaneously track the 3D joint coordinates of the human in real time. The skeleton tracking algorithm can identify 18 body feature points corresponding to the body joints for each person present in the camera field of view. However, in our experiments we limit the collaboration with the robot to one person, and we simplify the acquired skeleton by keeping only the joints necessary to reconstruct the SSLs of the arms, head and chest. The dimensions of the static bounding volumes enclosing the human have been taken from [40]. The human joint coordinates, acquired in the camera reference frame, are then transformed in the robot reference frame, in order to compute the distance between each couple of human and robot line segments in a consistent reference frame. For this purpose, the human joint coordinates are first transformed into a marker reference frame, by means of the ArUco open-source library based on OpenCV that provides a marker detection in real time [41, 42]. Then, the joint coordinates are transformed in the robot reference frame with a proper marker-to-robot calibration.

The optimization problem is implemented with the open-source tool for nonlinear optimization CasADi [43]. The use of this tool requires the recursive Newton-Euler inverse dynamics of the robot to be written in a symbolic form. For this reason, we adopt `urdf2casadi`, a software library for computing functions of the robot dynamics that can be adopted with symbolic expressions in the CasADi framework, based on a URDF description of the robot [44]. The parameters of the dynamic model of the Franka arm have been taken from the model that was experimentally verified in [38].

For a simpler implementation on the real robot, we consider the minimum-distance line segment $\delta(t)$ constant during the stop time. Therefore, the maximum linear velocity v_{δ_k} of a link is computed as in (9) using the same $\delta(t)$ during the stop time. This assumption does not introduce a significant error, as we experienced in our experiments.

Figure 12 shows six frames of an example experiment in which a human subject performs a cyclical motion around the robot so as to intrude in the robot workspace while the manipulator is performing a task. A video of this example experiment is reported in the supplementary material of the paper. The approach proposed in this work is general, meaning that the robot could perform any task, as for instance, assembly, pick-and-place, gluing or packaging operations. However, to ensure generality in the applicability of the proposed approach, a random joint-space nominal trajectory is performed by the manipulator during the experiment. The corresponding trajectory of the robot is shown in Fig. 13, where the joint positions, velocities, radii of the safety zones, and stop time obtained using Approach (1) are reported over time. Safety stops of the robot are denoted as shaded areas.

For comparison, Approaches (2) and (3) are also implemented experimentally. The total task time with the three different methods is evaluated with an additional experiment and an arbitrarily selected task for the human, which considered the same task execution instructions in all three cases. More in detail, the motion of the human is designed so as to cyclically intrude in the robot workspace, while the robot performs a random motion with a nominal duration of 30 s. The experimental results of this example test report a total task time equal to 37.80 s for Approach (1), 51.45 s for Approach (2), and 96.31 s for Approach (3). The proposed approach allows us to achieve a percentage improvement
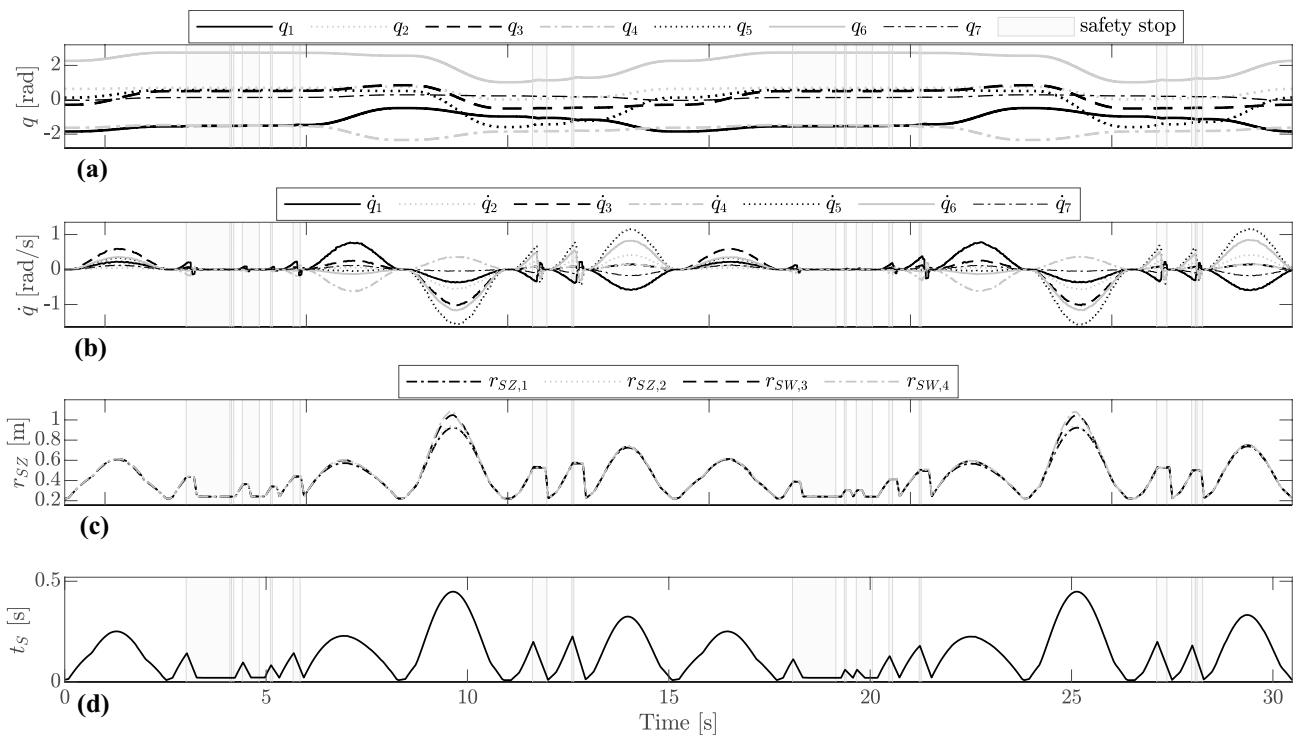
Fig. 13 Example experimental trajectory: joint positions (**a**); velocities (**b**); radii of the safety zones (**c**); and stop time (**d**) obtained using Approach (1)

of 26.53% with respect to Approach (2), and of 60.75% if Approach (3) is considered as the reference. The results of the experiments show a good matching with the extensive numerical simulations we performed, highlighting the effectiveness of the proposed approach for real collaborative scenarios.

To summarize, our method shows better performance in terms of total task time with respect to the considered alternative approaches, and demonstrates the benefits of the online scaling of dynamic safety zones also at experimental level. Furthermore, as testified by the metrics for human-robot collaboration, the proposed approach is promising for enhancing fluency and productivity in manufacturing applications where the presence of human alongside the manipulators is required.

6 Conclusion

In this paper, we presented an approach to enhance fluency and productivity in collaborative robotics through online scaling of bounding volumes considered as dynamic safety zones. Unlike other approaches, the directed speed between the robot parts and the human, the robot dynamics and its torque limits are here taken into account to plan online the trajectories that minimize the stop time. The strategy has been validated on a case study with extensive simulations in which the supervisory controller verifies online the separation distance between a robot performing random trajectories and a human moving around the manipulator and cyclically entering its workspace.

A comparison with an existing method that defines a smooth stop trajectory by simply considering n linearly spaced tentative stop times, and with an approach that considers an implementation of the technical specification ISO/TS 15066 has been performed. Furthermore, the application of existing objective fluency metrics for collaborative robotics, with the novel introduction of a new one named *concurrent activity in the robot workspace*, has shown the practical benefit and the effectiveness of the proposed approach with respect to the other benchmarks. Finally, the proposed approach has been experimentally tested with online computations on a 7-DOF Franka Emika Panda arm.

Future developments of this work will consider alternative strategies for the search of the best t_s and the solving of the optimization problem. Moreover, promising extensions of the proposed method could include other approaches for the trajectory planning subjected to kinematic and dynamic constraints [45], as well as for the definition of the final joint configuration of the robot at the end of the stop trajectory. Finally, the method could also be adapted for mobile manipulators to provide safety and enhance fluency and

productivity in more flexible and complex human-robot collaborative scenarios.

Declarations

References

1. Gualtieri L, Rauch E, Vidoni R (2021) Methodology for the definition of the optimal assembly cycle and calculation of the optimized assembly cycle time in human-robot collaborative assembly. Int J Adv Manuf Technol 113(7):2369–2384

2. Giusti A, Zeestraten MJ, Icer E, Pereira A, Caldwell DG, Calinon S, Althoff M (2018) Flexible automation driven by demonstration: Leveraging strategies that simplify robotics. IEEE Robot Autom Mag 25(2):18–27

3. Lee ML, Behdad S, Liang X, Zheng M (2022) Task allocation and planning for product disassembly with human-robot collaboration. Robot Comput Integr Manuf 76:102306

4. Realyvásquez-Vargas A, Arredondo-Soto KC, García-Alcaraz JL, Márquez-Lobato BY, Cruz-García J (2019) Introduction and configuration of a collaborative robot in an assembly task as a means to decrease occupational risks and increase efficiency in a manufacturing company. Robot Comput Integr Manuf 57:315–328

5. Saenz J, Behrens R, Schulenburg E, Petersen H, Gibaru O, Neto P, Elkmann N (2020) Methods for considering safety in design of robotics applications featuring human-robot collaboration. Int J Adv Manuf Technol 107(5):2313–2331

6. Zanchettin AM, Ceriani NM, Rocco P, Ding H, Matthias B (2015) Safety in human-robot collaborative manufacturing environments: Metrics and control. IEEE Trans Autom Sci Eng 13(2):882–893

7. Park JJ, Lee YJ, Song JB, Kim HS (2008) Safe joint mechanism based on nonlinear stiffness for safe human-robot collision. In: 2008 International Conference on Robotics and Automation, IEEE, pp 2177–2182

8. Kim U, Jo G, Jeong H, Park CH, Koh JS, Park DI, Do H, Choi T, Kim HS, Park C (2021) A novel intrinsic force sensing method for robot manipulators during human-robot interaction. IEEE Trans Rob

9. Seriani S, Gallina P, Scalera L, Lughi V (2018) Development of n-DoF preloaded structures for impact mitigation in cobots. J Mech Robot 10(5)

10. Kanazawa A, Kinugawa J, Kosuge K (2019) Adaptive motion planning for a collaborative robot based on prediction uncertainty to enhance human safety and work efficiency. IEEE Trans Rob 35(4):817–832

11. Safeea M, Neto P, Bearee R (2019) On-line collision avoidance for collaborative robot manipulators by adjusting off-line generated paths: An industrial use case. Robot Auton Syst 119:278–288

12. Casalino A, Bazzi D, Zanchettin AM, Rocco P (2019) Optimal proactive path planning for collaborative robots in industrial contexts. In: 2019 International Conference on Robotics and Automation, IEEE, pp 6540–6546

13. Scoccia C, Palmieri G, Palpacelli MC, Callegari M (2021) A collision avoidance strategy for redundant manipulators in dynamically variable environments: On-line perturbations of off-line generated trajectories. Machines 9(2):30

14. ISO (2011a) ISO 10218-1: 2011: Robots and robotic devices–Safety requirements for industrial robots–Part 1: Robots. Geneva, Switzerland: International Organization for Standardization

15. ISO (2011b) ISO 10218-2: 2011: Robots and robotic devices–Safety requirements for industrial robots–Part 2: Robot systems and integration. Geneva, Switzerland: International Organization for Standardization

16. ISO (2016) ISO/TS 15066: 2016 Robots and robotic devices-Collaborative robots. International Organization for Standardization, Geneva, Switzerland

17. Byner C, Matthias B, Ding H (2019) Dynamic speed and separation monitoring for collaborative robot applications-concepts and performance. Robot Comput Integr Manuf 58:239–252

18. Zardykhan D, Svarny P, Hoffmann M, Shahriari E, Haddadin S (2019) Collision preventing phase-progress control for velocity adaptation in human-robot collaboration. In: International Conference on Humanoid Robots (Humanoids), IEEE, pp 266–273

19. Corrales J, Candelas F, Torres F (2011) Safe human-robot interaction based on dynamic sphere-swept line bounding volumes. Robot Comput Integr Manuf 27(1):177–185

20. Beckert D, Pereira A, Althoff M (2017) Online verification of multiple safety criteria for a robot trajectory. In: Annual Conference on Decision and Control (CDC), IEEE, pp 6454–6461

21. Kim E, Kirschner R, Yamada Y, Okamoto S (2020) Estimating probability of human hand intrusion for speed and separation monitoring using interference theory. Robot Comput Integr Manuf 61:101819

22. Scalera L, Giusti A, Vidoni R, Di Cosmo V, Matt D, Riedl M (2020) Application of Dynamically Scaled Safety Zones Based on the ISO/TS 15066:2016 for Collaborative Robotics. Int J Mech Control 21(01):41–49

23. Ferraguti F, Bertuletti M, Landi CT, Bonfè M, Fantuzzi C, Secchi C (2020) A control barrier function approach for maximizing performance while fulfilling to iso/ts 15066 regulations. IEEE Robot Autom Lett 5(4):5921–5928

24. Palleschi A, Hamad M, Abdolshah S, Garabini M, Haddadin S, Pallottino L (2021) Fast and safe trajectory planning: Solving the cobot performance/safety trade-off in human-robot shared environments. IEEE Robot Autom Lett 6(3):5445–5452

25. Haddadin S, Haddadin S, Khoury A, Rokahr T, Parusel S, Burgkart R, Bicchi A, Albu-Schäffer A (2012) On making robots understand safety: Embedding injury knowledge into control. Int J Robot Res 31(13):1578–1602

26. Choi SH, Park KB, Roh DH, Lee JY, Mohammed M, Ghasemi Y, Jeong H (2022) An integrated mixed reality system for safety-aware human-robot collaboration using deep learning and digital twin generation. Robot Comput Integr Manuf 73:102258

27. Merckaert K, Convens B, Cj Wu, Roncone A, Nicotra MM, Vanderborght B (2022) Real-time motion control of robotic manipulators for safe human-robot coexistence. Robot Comput Integr Manuf 73:102223

28. Pérez L, Rodríguez-Jiménez S, Rodríguez N, Usamentiaga R, García DF, Wang L (2020) Symbiotic human-robot collaborative approach for increased productivity and enhanced safety in the aerospace manufacturing industry. Int J Adv Manuf Technol 106(3):851–863

29. Lin CH, Wang KJ, Tadesse AA, Woldegiorgis BH (2022) Human-robot collaboration empowered by hidden semi-markov model for operator behaviour prediction in a smart assembly system. J Manuf Syst 62:317–333

30. Hoffman G, Breazeal C (2007) Cost-based anticipatory action selection for human-robot fluency. IEEE Trans Rob 23(5):952–961

31. Nikolaidis S, Shah J (2013) Human-robot cross-training: computational formulation, modeling and evaluation of a human team training strategy. In: ACM/IEEE International Conference on Human-Robot Interaction (HRI), IEEE, pp 33–40

32. Dragan AD, Bauman S, Forlizzi J, Srinivasa SS (2015) Effects of robot motion on human-robot collaboration. 2015 10th ACM/IEEE Int. Conf. on Human-Robot Interaction, IEEE, pp 51–58

33. Hoffman G (2019) Evaluating fluency in human-robot collaboration. IEEE Trans Hum Mach Syst 49(3):209–218

34. Mayima A, Clodic A, Alami R (2020) Toward a robot computing an online estimation of the quality of its interaction with its human partner. In: IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), IEEE, pp 291–298

35. Scalera L, Vidoni R, Giusti A (2021) Optimal scaling of dynamic safety zones for collaborative robotics. In: 2021 International Conference on Robotics and Automation, IEEE, pp 3822–3828

36. Giusti A, Althoff M (2017) Efficient computation of interval-arithmetic-based robust controllers for rigid robots. In: IEEE International Conference on Robotic Computing, IEEE, pp 129–135

37. Ericson C (2004) Real-time collision detection. CRC Press

38. Gaz C, Cognetti M, Oliva A, Giordano PR, De Luca A (2019) Dynamic identification of the Franka Emika Panda robot with retrieval of feasible parameters using penalty-based optimization. IEEE Robot Autom Lett 4(4):4147–4154

39. Pereira A, Althoff M (2017) Overapproximative human arm occupancy prediction for collision avoidance. IEEE Trans Autom Sci Eng 15(2):818–831

40. Pheasant S, Haslegrave CM (2018) Bodyspace: Anthropometry, ergonomics and the design of work. CRC Press

41. Garrido-Jurado S, Muñoz-Salinas R, Madrid-Cuevas FJ, Medina-Carnicer R (2016) Generation of fiducial marker dictionaries using mixed integer linear programming. Pattern Recogn 51:481–491

42. Romero-Ramirez FJ, Muñoz-Salinas R, Medina-Carnicer R (2018) Speeded up detection of squared fiducial markers. Image Vis Comput 76:38–47

43. Andersson JAE, Gillis J, Horn G, Rawlings JB, Diehl M (2019) CasADi - A software framework for nonlinear optimization and optimal control. Math Program Comput 11(1):1–36

44. Johannessen LMG, Arbo MH, Gravdahl JT (2019) Robot Dynamics with URDF & CasADi. 2019 7th International Conference on Control. Mechatronics and Automation (ICCMA), IEEE, pp 1–6

45. Trigatti G, Boscariol P, Scalera L, Pillan D, Gasparetto A (2018) A new path-constrained trajectory planning strategy for spray painting robots. Int J Adv Manuf Technol 98(9):2287–2296