Safety Control of Industrial Robots Based on a Distributed Distance Sensor

Giovanni Buizza Avanzini, *Student Member, IEEE*, Nicola Maria Ceriani, *Student Member, IEEE*, Andrea Maria Zanchettin, *Member, IEEE*, Paolo Rocco, *Member, IEEE*, and Luca Bascetta, *Member, IEEE*

I. Introduction

HUMAN–ROBOT interaction (or HRI) is likely to play a key role in the future development of industrial production, with humans and robots working together at complementary tasks, while sharing a common workspace. The state of the art, however, is that industrial manipulators are still physically separated from human workers and installed in dedicated areas where humans are not allowed to enter. In addition, because of the inherent dangerousness that a moving robot represents for a human being (for the consequences of impacts between robots and humans see [1]–[3]), a preventive stop or a significant speed reduction are commanded by a suitable surveillance system, whenever a worker enters the robot area. The physical separation of robots from humans implies a significant cost for industries, often more relevant than the cost of the robots [4]. It also seriously limits the flexibility of the production system, since the presence of protective barriers prevents quick and inexpensive modifications of the factory layout. In order to allow a human–robot cooperation that better exploits manipulators' capabilities, the physical separation should be eliminated. Obviously, this creates considerable safety problems [5] that represent one of today's main focus of research in industrial robotics.

Manuscript received June 28, 2013; revised October 24, 2013; accepted December 26, 2013. Manuscript received in final form January 14, 2014. Date of publication February 14, 2014; date of current version October 15, 2014. This work was supported by the European Community Seventh Framework Program FP7/2007-2013 - Challenge 2 - Cognitive Systems, Interaction, Robotics - under Grant 230902 - ROSETTA. Recommended by Associate Editor G. Antonelli.

The authors are with the Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria, Milano 20133, Italy (e-mail: giovannimassimo.buizza@polimi.it; nicolamaria.ceriani@polimi.it; andreamaria.zanchettin@polimi.it; paolo.rocco@polimi.it; luca.bascetta@polimi.it).

A. Perception and Active Control

Sensor-based active control of robots is one of the most promising ways to tackle safety issues. The concept is to use information provided by sensors able to perceive changes in the robot's environment (exteroceptive sensors, such as vision, force, and distance sensors) and to dynamically adjust the robot behavior, accordingly. This strategy is particularly useful for industrial robots, that are not in general *(a priori)* mechanically designed for safe interaction with humans and therefore require a specific control strategy. If a contact between a human and a robot has to be avoided, visual and distance information have to be used. An effective way to deal with such information about the environment is the virtual impedance control, i.e., a modification of the well-known impedance control [6] for situations when a real contact between the robot and obstacles has to be avoided. The concept, originally introduced in [7] for mobile navigation and then extended to manipulator control (see [8], [9]), is based on calculating a virtual force through vision/distance sensors. This virtual force then serves as the input of the impedance controller, that regulates the robot dynamics so to achieve a compliant motion.

Though many technologies and principles are available for exteroceptive sensors (time of flight sensors, cameras, proximity sensors, depth space sensors, and so on), a first classification can be made based on the placement of such sensors with respect to the robot. One option is to use sensors mounted in fixed locations of the space. In [10], surveillance cameras fixed on the ceiling are exploited to detect, through image processing, the presence of humans in the robot's workspace, while predicting in real time their destination. In [11], a fusion between a RF localization system and fixed pyroelectric sensors is used for indoor localization of both human beings and robots, obtaining an accuracy of $\sim$1 m. In [12], the depth space sensor Microsoft Kinect is employed to detect humans close to a lightweight KUKA LWR IV, computing their mutual distance and then using such distance to modulate an evasive motion of the robot, thus avoiding collisions.

A second configuration consists in using sensors mounted directly on the robot (onboard). Such concept limits the sensors occlusion problem that affects fixed sensor systems. In addition, sensor mounted on the robots can manage

unstructured environments, while the use of sensors fixed in space requires to some extent a structuring of the workplace, that has to be prepared to house and calibrate the sensors. For these reasons, mobile robots particularly benefit from onboard sensors, also thanks to their possible minimal dimensions (see [13]).

In this paper, the onboard configuration has been exploited, referring in particular to the distributed sensor concept introduced in [4]. The distributed sensor is a sort of sensitive skin covering the surface of the manipulator, made up of a multitude of sensors that can be of different nature. In [14], distance sensors are used as components of the sensitive skin, demonstrating the effectiveness of the distributed distance sensor in monitoring a robot workspace in the context of safe HRI. Other works have developed the concept of distributed skin for control purposes, especially in the field of motion planning in unknown environment, see [15]–[17].

B. Safety Assessment

Beside the need to detect the presence of humans, another important issue is to quantify the level of danger, in order to adjust the robot's behavior accordingly. The first systematic quantitative method of safety evaluation in HRI was introduced and developed in [18]–[20], where the minimization of human risk in the field of human-care robotics is discussed by analyzing both mechanical design and control. In [21], the danger is evaluated by means of reflected inertia, relative velocity, and distance between human and robot. The needed information, acquired through a computer vision system and the measurement of some physiological signals, is integrated into long and short term safety strategies. In [22], a quantification of safety called impact potential is introduced, and a safety control scheme for robotic manipulators that complies to predefined limits of such potential is proposed. Another intuitive way to deal with safety oriented application is based on the potential field introduced in [23], directly connecting the danger assessment to the values of the repulsive potential. One drawback of such approach is that the potential field does not consider the relative motion between the robot and the obstacles, unlike, see [24]. In [25], a novel method for safety assessment called danger field is proposed. It is essentially based on the potential field method, but it considers the robot and not the obstacles as the field's source, taking into account their relative position, velocity, and direction of motion. A control strategy that increases human safety is then built upon the concept of danger field. Such concept has been exploited in [26] and [27] to shape a danger field-based control strategy that enhances human safety.

C. Contribution and Paper Organization

In this paper, we propose a control architecture that enhances safety of humans moving in an industrial environment in the proximity of robotic manipulators. Note that our approach only deals with avoiding collisions between humans and robots. The underlying idea is that our control architecture can be integrated in a more general scenario, managed by a higher lever controller, taking into account the possibility of impacts and of physical cooperation between humans and robots. In particular, in this paper, expanding the work presented in [14], a distributed distance sensor has been conceived, designed, produced, and mounted on the surface of an industrial robot, as part of the active control strategy. A control strategy that avoids collisions with the possibly detected humans has been designed, further developing the results of [28]. With respect to these previous works, this paper brings the following specific contributions:

1) an optimization procedure to obtain the best possible configuration of the distributed sensor;
2) details on the hardware implementation of the sensor;
3) a control strategy that exploits the computation of the danger field in several points as detected by the distributed sensor;
4) the combination of the danger field with a virtual impedance approach;
5) a control framework able to cope with nonredundant robots while preserving task consistency, in a hierarchical sense.

The control architecture has been experimentally tested in a demonstration of human–robot coexistence that resembles a possible industrial application, exploiting an industrial ABB IRB 140 robot endowed with an open controller interface. The rest of this paper is organized as follows. In Section II, the concept of danger field is reviewed. A method to evaluate the danger field using the distributed distance sensor measurements is then presented. In Section III, a danger field-based optimization method to displace the distributed sensor is proposed. Section IV describes the robot control strategy, designed to avoid collisions with the possibly detected humans, while preserving consistency with the main task. In Section V, the experimental setup used to validate our work is presented and detailed. Experimental tests and results are reported in Section VI. Finally, Section VII summarizes our contribution and suggests possible future research directions.

II. Danger Assessment Using a Distributed Distance Sensors

In an active approach to safe HRI, it is of fundamental importance to assess how dangerous a particular robot configuration could be for a human standing in the robot's workspace. In this context, it is obviously necessary for the control system to estimate the position of the human being, by means of suitable sensors' measurements. In this section, a brief review of the concept of danger field [25] is presented. Afterward, a definition of a distributed sensor is given, and a method to evaluate the danger field by exploiting the measurements of such a sensor is provided.

A. Danger Field Definition

In [25], a method for the quantification of danger in HRI through an artificial field, called danger field, is introduced. Basically, the danger field is a scalar quantity that captures how much a specific robot state of motion (position and velocity) is dangerous with respect to a generic point in the robot's

workspace. The idea behind is that the danger field evaluated in a given point decreases with the distance of such point from the robot, whereas it increases with the robot's velocity, in particular, if the robot moves toward the points where the field is computed. For a simple case of a point robot located at $\mathbf{r} \in \mathbb{R}^3$, moving with the velocity $\mathbf{v} \in \mathbb{R}^3$, the elementary danger field at the position $\mathbf{r}_j \in \mathbb{R}^3$ is defined as $\mathrm{DF}_e = \mathrm{SDF}_e + \mathrm{KDF}_e$, where

$$\mathrm{SDF}_e = \frac{1}{\left\|\mathbf{r}_j - \mathbf{r}\right\|} \tag{1}$$

$$\mathrm{KDF}_e = \frac{\|\mathbf{v}\|\left(1 + \cos\angle\left(\mathbf{r}_j - \mathbf{r}, \mathbf{v}\right)\right)}{\left\|\mathbf{r}_j - \mathbf{r}\right\|^2} \tag{2}$$

and SDF_e and KDF_e are the elementary static and kinematic danger fields, respectively. The elementary danger field can be extended to its cumulative version, which considers the position and velocity of the entire ith robot's link, by performing a path integration along the straight line that represents the wire model of the link

$$\mathrm{DF}_i = \int_0^1 \mathrm{SDF}_e\left(l\right) \mathrm{d}l + \int_0^1 \mathrm{KDF}_e\left(l\right) \mathrm{d}l. \tag{3}$$

Equation (3) implicitly assumes the following parameterization of the ith link's position and velocity:

$$\mathbf{r}_{i,l} = \mathbf{r}_i + l\left(\mathbf{r}_{i+1} - \mathbf{r}_i\right), \quad \mathbf{v}_{i,l} = \mathbf{v}_i + l\left(\mathbf{v}_{i+1} - \mathbf{v}_i\right) \tag{4}$$

where $\mathbf{r}_i$ and $\mathbf{r}_{i+1}$ are the endpoints of the ith link, $\mathbf{v}_i$ and $\mathbf{v}_{i+1}$ are the corresponding linear velocities, and $l \in [0, 1]$. For a robot with n_{link} links, the cumulative danger field CDF induced by the whole robot in the position of interest $\mathbf{r}_j$, $j = 1, \ldots, n_{\mathrm{obst}}$ (e.g., the relevant position of obstacles) can be expressed as

$$\mathrm{CDF}\left(\mathbf{r}_j\right) = \sum_{i=1}^{n_{\mathrm{link}}} \int_0^1 \frac{\mathrm{d}l}{\left\|\mathbf{r}_j - \mathbf{r}_{i,l}\right\|} + \sum_{i=1}^{n_{\mathrm{link}}} \int_0^1 \frac{\left\|\mathbf{v}_{i,l}\right\| \rho_{i,j,l}}{\left\|\mathbf{r}_j - \mathbf{r}_{i,l}\right\|^2} \mathrm{d}l \tag{5}$$

where $\rho_{i,j,l} = 1 + \cos\angle\left(\mathbf{r}_j - \mathbf{r}_{i,l}, \mathbf{v}_{i,l}\right)$. It is also possible to easily define a vector counterpart of the scalar CDF through its gradient $\nabla\mathrm{CDF}$

$$\overrightarrow{\mathrm{CDF}} = \mathrm{CDF}\frac{\nabla\mathrm{CDF}}{\|\nabla\mathrm{CDF}\|}. \tag{6}$$

Thus, $\overrightarrow{\mathrm{CDF}}(\mathbf{r}_j)$ is a vector anchored in $\mathbf{r}_j$, with the intensity $\mathrm{CDF}(\mathbf{r}_j)$, pointing in the direction defined by $\nabla\mathrm{CDF}$. For the computational aspect of the danger field, the reader may refer to [25]. For the purpose of this paper, it suffices to say that knowing the robot's joint variables $\mathbf{q}$ and velocities $\dot{\mathbf{q}}$ and the position of interest $\mathbf{r}_j$, it is possible to evaluate $\overrightarrow{\mathrm{CDF}}(\mathbf{r}_j)$ in closed form.

B. Danger Field Evaluation

The danger field induced by the robot can be computed in some significant positions of the workspace, e.g., the positions of human workers. The use of exteroceptive sensors able to detect obstacles (e.g., humans) in the robot's surroundings is therefore necessary. Let us consider a robot whose external surface is covered by a n_{spots} number of distance sensors called spots, constituting a sort of sensitive skin, namely a distributed distance sensor (Fig. 1). Each spot can detect an object placed in the sensor's measurement direction, and measure the corresponding distance. Let us assume that potentially each of the n_{link} robot's links can host n_i different spots. A methodology to optimally place such spots on each link will be presented in Section III. Through direct kinematics, knowing the robot joint variables $\mathbf{q}$ and the distance d_k measured by the kth spot placed on the ith link, it is possible to reconstruct the position $\mathbf{r}$ of the detected point in the robot base frame as follows:

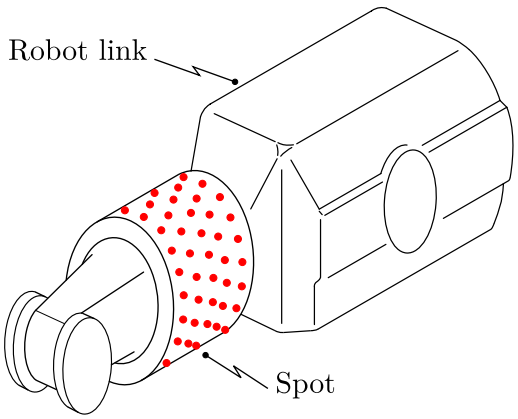


Fig. 1. Conceptual sketch of the distributed distance sensor mounted on a robot link.

$$\begin{cases} \mathbf{r}_{\mathrm{hom}} = {}^0\mathbf{T}_{\mathrm{DH}_i}\left(\mathbf{q}\right) {}^{\mathrm{DH}_i}\mathbf{T}_k \, \mathbf{r}_{k,\mathrm{hom}} \\ \mathbf{r}_{\mathrm{hom}} = \left[r_x, r_y, r_z, 1\right]^T = \left[\mathbf{r}^T, 1\right]^T \\ \mathbf{r}_{k,\mathrm{hom}} = \left[0, 0, d_k, 1\right]^T \end{cases} \tag{7}$$

where ${}^0\mathbf{T}_{\mathrm{DH}_i}(\mathbf{q})$ is the 4-by-4 homogeneous matrix from base frame to the Denavit-Hartenberg frame of the ith joint, function of the robot's pose $\mathbf{q}$. ${}^{\mathrm{DH}_i}\mathbf{T}_k$ is the constant 4-by-4 homogeneous matrix between the Denavit-Hartenberg frame of the ith joint and the kth spot's frame, i.e., a frame centered in the spot position and having the z-axis oriented as the spot's measurement direction. $\mathbf{r}_{\mathrm{hom}}$ is the 4-by-1 homogeneous position vector of the detected point, expressed in the robot's base frame. $\mathbf{r}_{k,\mathrm{hom}}$ is the 4-by-1 homogeneous position vector of the detected point expressed in the kth spot's frame. Knowing the position of the detected point thanks to (7) and the joint variables $\mathbf{q}$ and velocities $\dot{\mathbf{q}}$, it is possible to compute the danger field in that point.

Potentially each spot can detect a different obstacle, or different points on the same obstacle, producing up to n_{spots} cumulated danger field vectors. In order to reduce the possibly large amount of vectors $\overrightarrow{\mathrm{CDF}}(\mathbf{r}_k)$, $k = 1, \ldots, n_{\mathrm{spots}}$, while maintaining an adequate completeness of the spatial information about danger, it is possible to sum the information provided link-by-link into a single vector

$$\overrightarrow{\mathrm{CDF}}_{\mathrm{link}_i} = \sum_{k=1}^{n_i} \left(\overrightarrow{\mathrm{CDF}}\left(\mathbf{r}_k\right)\right). \tag{8}$$

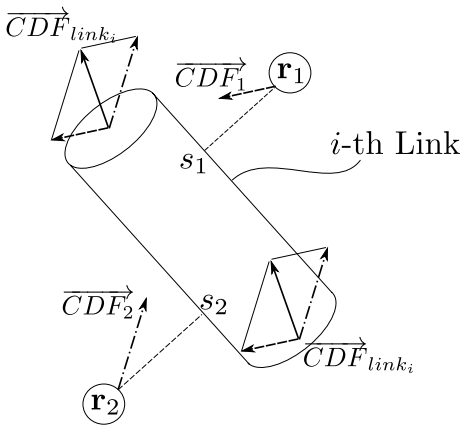


Fig. 2. Two spots s_1 and s_2 on link i detecting two obstacle points r_1 and r_2. The corresponding danger field vectors are summed into $\overrightarrow{\text{CDF}}_{\text{link}_i}$ and applied to the link's endpoints.

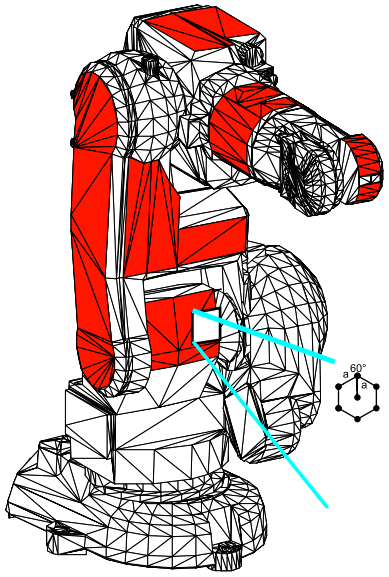


Fig. 3. CAD model of an ABB IRB 140 robot. A possible combination of regions selected to host the sensor spots is highlighted. A portion of the grid of admissible nodes is also reported.

Fig. 2 shows an example of the generic ith link, with two sensor spots s_1 and s_2 detecting two obstacles in $\mathbf{r}_1$ and $\mathbf{r}_2$. The robot configuration induces the cumulative danger field $\overrightarrow{\text{CDF}}_1$ and $\overrightarrow{\text{CDF}}_2$ in the detected points $\mathbf{r}_1$ and $\mathbf{r}_2$, respectively, whose sum results in $\overrightarrow{\text{CDF}}_{\text{link}_i}$. In this way, up to n_{spots} cumulative danger field vectors associated to each detected point are reduced up to n_{link} cumulative danger field vectors associated to each sensorized link of the manipulator.

In general, it is possible that the distributed sensor detects points on the surface of known obstacles that are part of the nominal robot workspace. Such points should obviously not be considered as in danger. For this reason, measurements regarding known obstacles in the robot's workspace are not considered when computing $\overrightarrow{\text{CDF}}_{\text{link}_i}$, by verifying through dedicated software if the detected points belong to a convex hull representation of the known obstacles. The same applies to measurements regarding points detected on some links of the robot itself (self-detection). For safety reasons, however, a threshold minimum distance d_{min} is defined. Measurements $d_k < d_{\text{min}}$ are always considered in the danger field computation.

III. Optimal Placement of the Distributed Sensor

In order to develop a reliable safety control system, it is necessary to distribute the sensor spots of the distance sensor in a way that maximizes the probability to detect a human in the robot's workspace. In [14], a method for the sizing of a distributed distance sensor able to detect obstacles of a certain dimension is first proposed. In this section, an optimization method to deal with a limited number of available spots is presented, obtaining the optimal placement on the robot's surface. The optimization procedure works as follows: first, it is necessary to select some regions on the robot's surface that can actually host the sensor's spots, for example, using computer-aided design (CAD) data provided by the robot manufacturer. A grid of equally spaced nodes laying on such surfaces, representing the admissible spots positions on the manipulator, can be then automatically generated. Fig. 3 reports for example the CAD model of an ABB IRB 140 robot. The selected regions are highlighted, showing the grid of admissible nodes. For details about ABB IRB 140 robot, see Section V. Note that the selection of the optimal nodes exploiting a CAD model of the robot allows to automatically derive the homogeneous matrices ${}^{\text{DH}_i}\mathbf{T}_k$ of (7), thus facilitating the calibration process during the physical deployment of the distributed sensor.

The number n_{nodes} of available nodes can be order of magnitudes larger than the number of available sensor spots, making the selection of the most appropriate distribution a nontrivial task. Let $\mathbf{y} = [y_1\ y_2\ \dots\ y_{n_{\text{nodes}}}]^T$ be a vector of boolean elements characterizing a particular sensor placement, where $y_i = 1$ denotes that one spot has been placed in position i of the grid ($y_i = 0$, otherwise), and n_{nodes} be the total number of points where spots can be placed. Also, let $n_{\text{spots}} \ll n_{\text{nodes}}$ be the number of available sensor spots, which constrains $\mathbf{y}$ to be such that $\sum_{i=1}^{n_{\text{nodes}}} y_i \leq n_{\text{spots}}$.

The idea is to exploit the CDF in its static form as a measure of the effectiveness of placing one spot in the ith node by defining the following function:

$$\mathbf{Q}_i(\mathbf{q}, \boldsymbol{\xi}) = \begin{cases} \text{SDF}(\mathbf{q}, \mathbf{p}_i(\mathbf{q}, \boldsymbol{\xi})), & \text{if detected} \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where $\mathbf{q}$ are the already defined robot joint variables, while $\boldsymbol{\xi}$ are a set of coordinates necessary to define the position and orientation of the considered obstacle. $\mathbf{p}_i$ is the position of the particular point detected by the ith spot on the considered obstacle. Such position therefore depends both on $\mathbf{q}$, whose value determines the spot position and orientation, and on $\boldsymbol{\xi}$, whose value determines the obstacle position and orientation. Equation (9) basically returns the static danger field in the detected point $\mathbf{p}_i$ if there is a detection, or zero if no detection occurs. Let now $\max_{\mathbf{r}_j} \text{SDF}\left(\mathbf{q}, \mathbf{r}_j(\boldsymbol{\xi})\right)$ be the maximum value of the static danger field induced by the robot in a given

configuration $\mathbf{q}$ among all the points $\mathbf{r}_j$ belonging to the considered obstacle's surface, whose positions obviously depend on $\boldsymbol{\xi}$. The following optimization problem is then set:

$$\begin{aligned} &\min_{\mathbf{y}} \mathrm{CF}\,(\mathbf{y}) \\ &\text{s.t.} \sum_{i=1}^{n_{\text{nodes}}} y_i \leq n_{\text{spot}}, \quad y_i \in \{0,1\} \end{aligned} \tag{10}$$

where

$$\mathrm{CF}\,(\mathbf{y}) = \sum_{i=1}^{n_{\text{nodes}}} \mathrm{CF}_i\,(y_i) \tag{11}$$

and

$$\mathrm{CF}_i\,(y_i) = \sum_{\mathbf{q},\boldsymbol{\xi}} \left(\max_{\mathbf{r}_j} \mathrm{SDF}\left(\mathbf{q}, \mathbf{r}_j\right) - y_i\, \mathbf{Q}_i\,(\mathbf{q}, \mathbf{q}_{\text{obst}}) \right). \tag{12}$$

$\mathrm{CF}_i\,(y_i)$ assesses the difference between the maximum static danger field evaluated on the obstacle surface and the static danger field that is actually computed in the point detected by the considered ith spot. The cost function $\mathrm{CF}\,(\mathbf{y})$ consequently quantifies how a sensor placement $\mathbf{y}$ is effective in detecting the obstacles, possibly all, in the workspace of the robot. The result of the optimization is therefore the sensor arrangement $\mathbf{y}$ that not only possibly detects all the obstacles, but that is also able to detect those points that correspond to the highest value of danger field. The resulting configuration thus provides the most accurate information on how dangerous the robot is with respect to the detected obstacles.

Note that the cost function defined in (12) requires an infinite number of robot and obstacles configurations to be selected in order to completely characterize the workspace. Given the unfeasibility of such an approach, and following the methodology developed in [29], it is possible to randomly select a limited number of configurations, still guaranteeing a reliable approximated result, in a statistical sense.

The resulting problem is a typical knapsack-like combinatorial one, whose optimal solution can be obtained with well-known state-of-the-art solvers, see [30].

The proposed framework not only allows to select the optimal placement for a given n_{spot} number of sensors, but it can also be exploited to compute what is the minimum number of spots (and their optimal placement) that is able to guarantee a certain probability of detecting any obstacle in the robot workspace. Once the desired minimum detection probability $\mathscr{P}_{\text{thr}} \in [0, 1]$ has been fixed, such problem can be solved through a feasibility problem

$$\begin{aligned} &\min_{\mathbf{y}} \sum_{i=1}^{n_{\text{nodes}}} y_i \\ &\text{s.t.}\, \mathscr{P}\left(\sum_{i=1}^{n_{\text{nodes}}} \mathbf{Q}_i\,(\mathbf{q}, \boldsymbol{\xi}) > 0 \right) \geq \mathscr{P}_{\text{thr}} \end{aligned} \tag{13}$$

where $\mathscr{P}\,(\cdot)$ represents the detection probability function. An example of the optimization procedure is reported in Section V-C.

IV. Design of the Control Strategy

Thanks to the distributed distance sensor mounted on the robot manipulator, it is possible to detect the presence of a human in the robot workspace. His or her danger level can be evaluated by computing the danger field in the detected points. The idea is to exploit the $\overrightarrow{\mathrm{CDF}}_{\text{link}_i}$, $i = 1, \ldots, n_{\text{link}}$, defined in (8) to shape a control action enhancing the safety level of the human being, while trying to maintain a certain consistency with the robot task. A first danger field-based control law was proposed in [25], in a centralized and model-based form that requires the knowledge of the robot dynamics and the access to low-level motion control, producing the reference torques for the robot joints. This could be an issue in industrial robotics; although, in recent years, a growing number of industrial robots have been endowed with centralized, model-based, control laws, these require access to the low level joint torques, which is not always possible. In addition, for an effective use of such control laws, the joints should be torque controlled, and this is not the case in state-of-the-art industrial robotics. Therefore, in [27], a new control strategy that solves the problem completely on kinematic level is presented. The concept has been used in [28], where experimental evidences of the feasibility of a danger field-based control has also been given. A further development of the previously proposed control schemes is proposed here, integrating the concept of virtual impedance [6], [8] with the danger assessment evaluated through the distributed distance sensor (thus with several measurements of danger field).

A. Evasive Motion

As stated in Section II, $\overrightarrow{\mathrm{CDF}}_{\text{link}_i}$ lies in the direction of maximum danger increase for the detected obstacles with respect to the robot position and velocity. Moving the robot away from the obstacles in such a direction is therefore a natural approach to the problem of danger reduction. An effective way to have the ith link moving in that direction is by using a virtual impedance approach, interpreting $\overrightarrow{\mathrm{CDF}}_{\text{link}_i}$ as a virtual force to be applied at the end points of the corresponding ith link (i.e., the $(i-1)$th and ith Denavit-Hartenberg frames). This procedure can be applied to each of the n_{link} links of the robot, using the respective cumulated danger field vector. Exploiting the static relationship between the joint torques and the wrench vector (see [31]), two vectors of virtual joint torques $\boldsymbol{\sigma}_{i-1}$ and $\boldsymbol{\sigma}_i$ can be then easily computed for each link as

$$\begin{cases} \boldsymbol{\sigma}_{i-1} = \mathbf{J}_{i-1}\,(\mathbf{q})^T\, \mathbf{W}_i = \mathbf{J}_{i-1}\,(\mathbf{q})^T \left[\overrightarrow{\mathrm{CDF}}^T_{\text{link}_i}\, 0\, 0\, 0 \right]^T \\ \boldsymbol{\sigma}_i = \mathbf{J}_i\,(\mathbf{q})^T\, \mathbf{W}_i = \mathbf{J}_i\,(\mathbf{q})^T \left[\overrightarrow{\mathrm{CDF}}^T_{\text{link}_i}\, 0\, 0\, 0 \right]^T \end{cases} \tag{14}$$

where $\mathbf{J}_i\,(\mathbf{q})$ is the Jacobian matrix associated to the ith Denavit-Hartenberg frame, while $\mathbf{W}_i$ is the virtual wrench vector associated to the ith link. The wrench vector is by definition composed by the forces and torques applied to the link. Since no virtual torque is applied to the ith link, only the first three rows of the virtual wrench $\mathbf{W}_i$ are nonzero, and are set equal to the virtual force $\overrightarrow{\mathrm{CDF}}_{\text{link}_i}$.

In order to obtain a single torque set $\boldsymbol{\sigma}$ that takes into account all the virtual forces applied to the links, the superposition of effects can be used

$$\boldsymbol{\sigma} = \sum_{i=1}^{n_{\text{link}}} (\boldsymbol{\sigma}_{i-1} + \boldsymbol{\sigma}_i). \tag{15}$$

The resulting virtual torque $\boldsymbol{\sigma}$ represents the control action to be applied to the robot. Being shaped on the virtual force $\overrightarrow{\text{CDF}}_{\text{link}_i}$, such action entails an evasive motion that steers the robot away from all the detected obstacles, according to their danger level. To obtain the evasive joint velocities $\dot{\mathbf{q}}_0$ corresponding to $\boldsymbol{\sigma}$, the virtual torque can be processed by a mass-damper admittance filter

$$\dot{\mathbf{q}}_0 = (\mathbf{M}s + \mathbf{D})^{-1} \boldsymbol{\sigma} \tag{16}$$

where s is the frequency domain operator. $\mathbf{M} = \mathbf{M}^T > 0$ and $\mathbf{D} = \mathbf{D}^T > 0$ are the mass matrix and damping matrix of the virtual impedance filter, respectively, whose values affect the actual dynamic response of the robot to the virtually applied torques $\boldsymbol{\sigma}$. In the following, we describe how to process $\dot{\mathbf{q}}_0$ in order to compute the joint references to be fed to the industrial robot controller.

B. Task Consistency

In (16), $\dot{\mathbf{q}}_0$ are the joint velocities that steer the robot away from the detected obstacles. However, the robot is typically performing a task (from now on, the main task). Note that in industrial robotics, the main task is typically defined by specifying the end-effector's Cartesian position and orientation (requiring $n_{\text{DoF}} = 6$ degrees of freedom), or some subset (requiring $n_{\text{DoF}} < 6$). A task is therefore characterized by the Jacobian matrix $\mathbf{J}_{\text{task}}(\mathbf{q})$, i.e., a submatrix of the end-effector Jacobian $\mathbf{J}_{\text{ee}}(\mathbf{q})$ containing only the appropriate rows. It would obviously be unproductive to interrupt the main task every time a human is detected. The goal of this section is then to describe a methodology to maintain task consistency while performing the evasive motion. The idea is to exploit only the (possibly) redundant degrees of freedom to deliver the evasive motion when the danger level is sufficiently low. On the other hand, when the danger level is high, the main task should be released and a complete evasive motion executed, exploiting all the degrees of freedom.

1) Null-Space Projection: A method to maintain task consistency is to consider the evasive velocities as a robot's subtask, by means of a null-space projection in the main task's Jacobian matrix $\mathbf{J}_{\text{task}}(\mathbf{q})$ (see [32], [33]). The reference joint velocities $\dot{\mathbf{q}}$ in this case result

$$\begin{cases} \dot{\mathbf{q}} = \dot{\mathbf{q}}_{\text{task}} + \dot{\mathbf{q}}_{\text{subtask}} = \dot{\mathbf{q}}_{\text{task}} + \mathbf{N}_{\text{task}}(\mathbf{q})\, \dot{\mathbf{q}}_0 \\ \mathbf{N}_{\text{task}}(\mathbf{q}) = \mathbf{I} - \mathbf{J}^{\dagger}_{\text{task}}(\mathbf{q})\, \mathbf{J}_{\text{task}}(\mathbf{q}) \end{cases} \tag{17}$$

where $\mathbf{J}^{\dagger}_{\text{task}}(\mathbf{q})$ is the Moore-Penrose pseudoinverse of $\mathbf{J}_{\text{task}}(\mathbf{q})$ and $\mathbf{N}_{\text{task}}(\mathbf{q})$ is the projector in $\mathbf{J}_{\text{task}}(\mathbf{q})$ null-space. The term $\dot{\mathbf{q}}_{\text{task}}$ expresses the main task's desired velocities and will be detailed in the following. By projecting $\dot{\mathbf{q}}_0$ in the null-space of $\mathbf{J}_{\text{task}}(\mathbf{q})$, the evasive motion is (partially) performed exploiting only the redundant degrees of freedom. Thus, the robot main task remains unaffected, assuring task consistency. Note that if the considered robot has a number of degree of freedom $n_{\text{DoF}} > 6$, the robot is inherently redundant, therefore $\dot{\mathbf{q}}_{\text{subtask}}$ always affects the manipulator's posture.

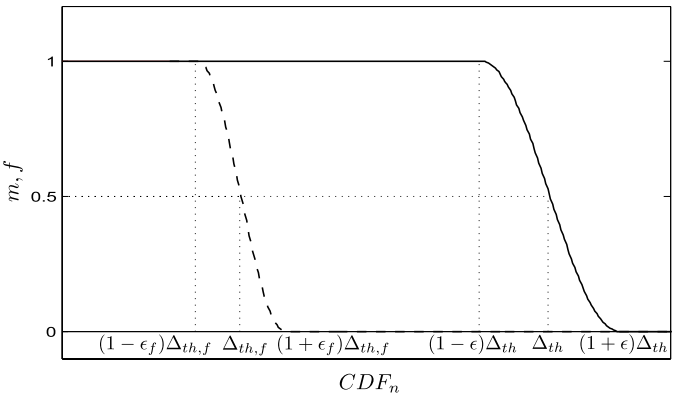


Fig. 4. Parameters m (continuous line) and f (dashed line), used to hierarchically release tasks, depending on the cumulated danger field norm CDF_n.

2) Main Task's Release: When the robot configuration is too dangerous, a complete evasive motion should be performed to enhance safety, interrupting the main task, and exploiting all the available degrees of freedom. Equation (17) can be then modified as follows (see [27]):

$$\begin{cases} \dot{\mathbf{q}} = m\, \dot{\mathbf{q}}_{\text{task}} + \mathbf{N}_m(\mathbf{q})\, \dot{\mathbf{q}}_0 \\ \mathbf{N}_m(\mathbf{q}) = \mathbf{I} - m\, \mathbf{J}^{\dagger}_{\text{task}}(\mathbf{q})\, \mathbf{J}_{\text{task}}(\mathbf{q}). \end{cases} \tag{18}$$

Note that when $m = 0$, the main task is released and the reference velocities $\dot{\mathbf{q}}$ becomes the sole evasive motion defined by $\dot{\mathbf{q}}_0$. In particular, m is defined as

$$m = \begin{cases} 1, & \text{if } \text{CDF}_n \le \Delta_{th}(1-\epsilon) \\ 0, & \text{if } \text{CDF}_n \ge \Delta_{th}(1+\epsilon) \\ \dfrac{1}{2} - \dfrac{1}{2}\sin\dfrac{\pi\,(\text{CDF}_n - \Delta_{\text{th}})}{2\epsilon\,\Delta_{\text{th}}}, & \text{otherwise} \end{cases} \tag{19}$$

where $\text{CDF}_n = L_\infty\big(\|\overrightarrow{\text{CDF}}(\mathbf{r}_1)\|, \ldots, \|\overrightarrow{\text{CDF}}(\mathbf{r}_{n_{\text{spot}}})\|\big)$ is the L_∞-norm of the cumulated danger field evaluated at each detected point $\mathbf{r}_k$, $k = 1, \ldots, n_{\text{spots}}$. ϵ is a parameter that allows a smooth transition between $m = 1$ and $m = 0$ (Fig. 4). In (19), m turns out to be equal to 1 if and only if CDF_n does not exceed a certain threshold $\Delta_{\text{th}}(1-\epsilon)$, i.e., if no detected point is considered to be in excessive danger. If at least one of the detected points is in such a danger that $\text{CDF}_n \ge \Delta_{\text{th}}(1+\epsilon)$, the algorithm sets $m = 0$ and $\dot{\mathbf{q}}$ becomes equal to the sole evasive action $\dot{\mathbf{q}}_0$. Thus, in a dangerous situation, all the degrees of freedom are exploited for the safety action, that is fully executed.

3) Task Prioritization: If the robot is nonredundant with respect to the main task, no evasive motion can be performed unless $m = 0$. It is then possible to subdivide the main task into two (or more) subtasks having different priority (see [34]–[37]). Tasks with lower priority can be suspended to partially perform the evasive action even when $m = 1$. With this goal, the reference velocities $\dot{\mathbf{q}}_{\text{task},L}$ of low priority

TABLE I
ROBOT'S BEHAVIOR DEPENDING ON DANGER LEVEL

	$m=0$	$m=1$
$f=0$	High danger level. The main task is suspended in favor of the sole evasive action.	Medium danger level. The low priority task is suspended in favor of the evasive action, while maintaining the high priority task.
$f=1$	Impossible situation (see Figure 4).	Low danger level. The main task is preserved, exploiting possible redundancy to perform the evasive action.

tasks are projected in the null-space of the higher priority Jacobian $\mathbf{J}_{\text{task},H}$, guaranteeing the fulfillment of at least the most relevant task

$$\begin{cases} \dot{\mathbf{q}}_{\text{task}} = \dot{\mathbf{q}}_{\text{task},H} + f\mathbf{N}_H(\mathbf{q})\,\dot{\mathbf{q}}_{\text{task},L} \\ \mathbf{N}_H(\mathbf{q}) = \mathbf{I} - \mathbf{J}^{\dagger}_{\text{task},H}(\mathbf{q})\,\mathbf{J}_{\text{task},H}(\mathbf{q}) \end{cases} \tag{20}$$

with the obvious meaning of $\dot{\mathbf{q}}_{\text{task},H}$ and $\mathbf{J}^{\dagger}_{\text{task},H}(\mathbf{q})$. f is a smoothing parameter similar to m (Fig. 4), which allows to selectively release or resume the low priority task depending on the danger level

$$f = \begin{cases} 1, & \text{if } \text{CDF}_n \le \Delta_{\text{th},f}\left(1-\epsilon_f\right) \\ 0, & \text{if } \text{CDF}_n \ge \Delta_{\text{th},f}\left(1+\epsilon_f\right) \\ \dfrac{1}{2} - \dfrac{1}{2}\sin\dfrac{\pi\left(\text{CDF}_n - \Delta_{\text{th},f}\right)}{2\epsilon_f\,\Delta_{\text{th},f}}, & \text{otherwise} \end{cases} \tag{21}$$

with $\Delta_{\text{th},f} < \Delta_{\text{th}}$ being a given threshold, and ϵ_f a suitable smoothing parameter. Equation (18) is then modified as follows:

$$\begin{cases} \dot{\mathbf{q}} = m\,\dot{\mathbf{q}}_{\text{task}} + \left(f\,\mathbf{N}_m(\mathbf{q}) + (1-f)\,\mathbf{N}_{H,m}(\mathbf{q})\right)\dot{\mathbf{q}}_0 \\ \mathbf{N}_{H,m}(\mathbf{q}) = \mathbf{I} - m\mathbf{J}^{\dagger}_{\text{task},H}(\mathbf{q})\,\mathbf{J}_{\text{task},H}(\mathbf{q}). \end{cases} \tag{22}$$

The task prioritization and the use of the m and f parameters allow to modulate the robot behavior according to the detected danger level, as summarized in Table I. Note that whenever $\Delta_{\text{th},f}\left(1-\epsilon_f\right) \le \text{CDF}_n \le \Delta_{\text{th},f}\left(1+\epsilon_f\right)$ or $\Delta_{\text{th}}(1-\epsilon) \le \text{CDF}_n \le \Delta_{\text{th}}(1+\epsilon)$, there is a smooth transition between phases of Table I, thanks to the shape of m and f parameters.

During the robot working cycle, the definition of the high and low priority tasks may change. $\mathbf{J}_{\text{task},H}(\mathbf{q})$ and $\mathbf{J}_{\text{task},L}(\mathbf{q})$ should be changed accordingly, selecting the appropriate rows of $\mathbf{J}_{\text{ee}}$.

4) Generating Robot References $\mathbf{q}$*:* Let $\mathbf{x}_{d,H}$ be a vector containing the desired end-effector's Cartesian position and orientation for the high priority task, and $\dot{\mathbf{x}}_{d,H}$ its time derivative. Let also $\mathbf{x}_{d,L}$ be the desired end-effector's Cartesian position and orientation for the low priority task, and $\dot{\mathbf{x}}_{d,L}$ its time derivative. The corresponding velocities in the joint space, i.e., $\dot{\mathbf{q}}_{\text{task}}$, and subsequently the robot references $\mathbf{q}$ and $\dot{\mathbf{q}}$ of (22), can be computed through a CLIK algorithm (see [38], [39]). Fig. 5 shows the block diagram of the CLIK algorithm combined with (22). $\mathbf{x}_L$ and $\mathbf{x}_H$ are the actual commanded end-effector references for the low and high priority tasks, respectively, and are used to close the CLIK loop.

In summary, the proposed control scheme is able to maintain consistency with the robot's main task, exploiting possible redundant degrees of freedom to perform the safety motion. Such a motion is tuned on the danger level of all the detected obstacles so to steer the robot away from them. If the danger is sufficiently high for at least one of the obstacles, the main task is released and the safety motion exploits all the degrees of freedom. If the robot is nonredundant with respect to the main task, it is possible to subdivide it into two or more tasks with hierarchical priority. The task with low priority can be released as a function of the danger level, assigning the new redundant degrees of freedom to the safety motion while maintaining the high priority task's consistency.

C. Recovery Phase

Let $\mathbf{x}_d$ and $\dot{\mathbf{x}}_d$ be the task references (to be later subdivided into $\dot{\mathbf{x}}_{d,H}$, $\mathbf{x}_{d,H}$, $\dot{\mathbf{x}}_{d,L}$, and $\mathbf{x}_{d,L}$). Let also assume that $\mathbf{x}_d$ and $\dot{\mathbf{x}}_d$ are computed online by a trajectory generation algorithm. Such references are fed to the CLIK algorithm, which evaluates the corresponding $\mathbf{q}$ joint references. As previously explained, whenever the danger level is sufficiently high, the main task is suspended and a complete evasive motion is performed. Let $\mathbf{x}_{d,s}$ be the robot's reference at the instant when CDF_n exceeds the threshold $\Delta_{\text{th}}(1-\epsilon)$ and $\mathbf{x}_{d,r}$ be the robot's configuration at the instant when CDF_n decreases again and the main task is resumed. Let also $\mathbf{e}_{d,r} = \mathbf{x}_{d,s} - \mathbf{x}_{d,r}$ be the difference between these two configurations. In general, $\mathbf{e}_{d,r} \neq \mathbf{0}$ at the end of the evasive motion. During the task suspension, it is advisable to interrupt the trajectory generation, maintaining $\mathbf{x}_d = \mathbf{x}_{d,s}$ and $\dot{\mathbf{x}}_d = \mathbf{0}$ as fixed references. In this way, when the main task is resumed, the CLIK algorithm computes references that make the robot come back from $\mathbf{x}_{d,r}$ to $\mathbf{x}_{d,s}$. At this point, the main task's trajectory generation can restart exactly from the point it has been suspended.

If $\mathbf{x}_{d,s}$ is considerably different from $\mathbf{x}_{d,r}$, however, the initial error $\mathbf{e}_{d,r}$ fed to the CLIK algorithm can be very large, producing quick and possibly unsafe motions. To avoid this problem, a recovery phase has been designed to take place when the main task should be resumed. During the recovery phase, a trajectory generation algorithm computes the $\mathbf{x}_{d,\text{rec}}$ and $\dot{\mathbf{x}}_{d,\text{rec}}$ references that smoothly move the robot from $\mathbf{x}_{d,r}$ to $\mathbf{x}_{d,s}$. At each time instant, the CLIK algorithm is thus fed with the difference between the actual robot's configuration and $\mathbf{x}_{d,\text{rec}}$, small enough not to over stress the robot. When the actual robot's configuration is equal to $\mathbf{x}_{d,s}$, the recovery phase ends and the main task's trajectory generation is restarted.

V. EXPERIMENTAL SETUP

A. Robotic System

The control system presented in Section IV has been tested using an ABB IRB 140 robot, a six degrees of freedom

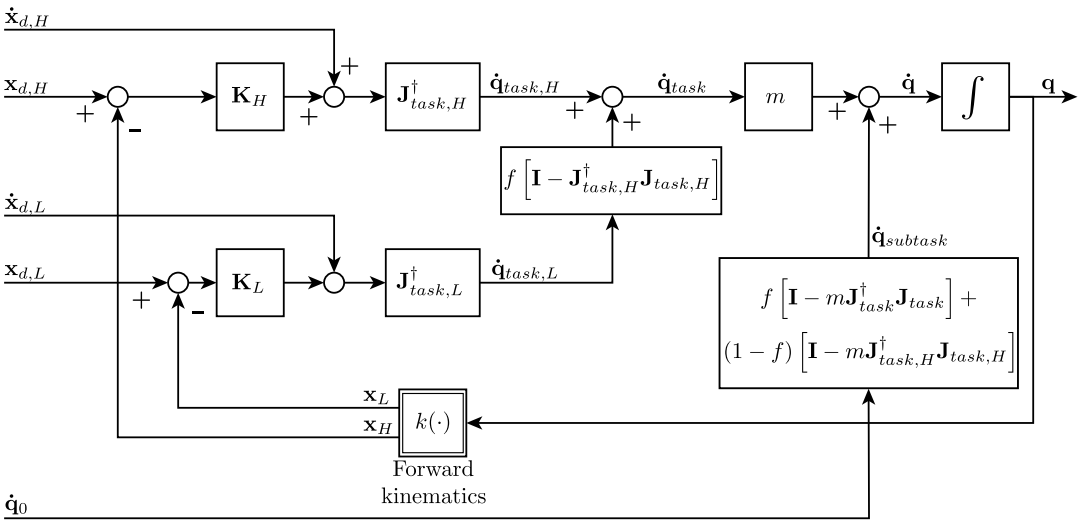


Fig. 5. Block diagram of the proposed control strategy based on a CLIK algorithm. The low priority task and the evasive action are projected into the proper null-spaces. Parameters m and f enforce the hierarchical release of tasks according to Table I.

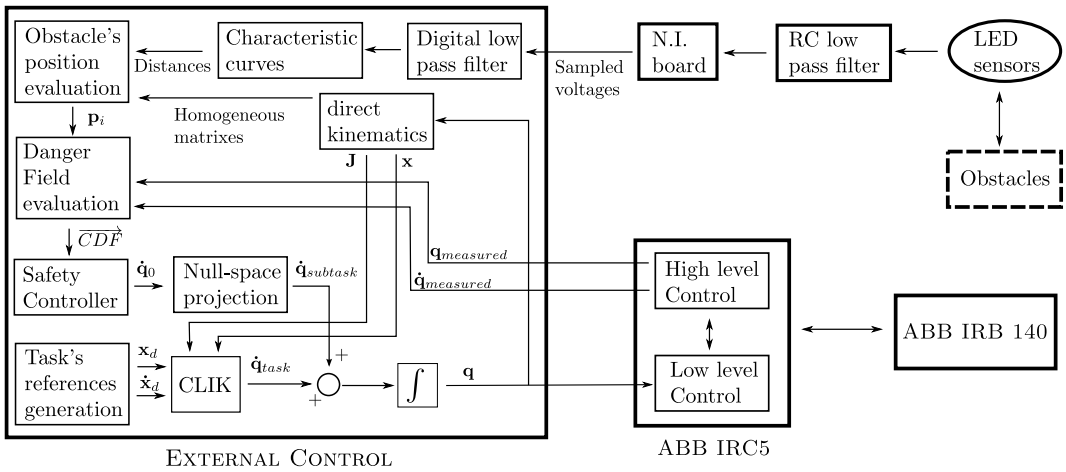


Fig. 6. Diagram of the experimental hardware and software setup. The external control, performing the control strategy, is interfaced with both the distance sensors and the ABB IRC5 industrial controller, thus allowing to modify the references fed to the ABB IRB 140 robot.

industrial manipulator, with a maximum payload of 6 kg and a maximum reach of 0.810 m. The ABB IRB 140 is endowed with an ABB IRC5 control unit. In particular, the considered control unit has been connected via Ethernet to an external PC running under Linux operating system with Xenomai patch, which allows hard real-time operations, obtaining an open controller configuration of the robot, where the PC serves as an external control.

Exploiting an interface developed by Lund University [40], it is in fact possible to conceptually insert the external PC between the IRC5 high level control, that generates the joint references, and the IRC5 low-level control, that receives the high-level references and produces the motors references. It is therefore possible to compute the desired reference on the external PC and to feed them to the low-level control.

In particular, the desired control strategy can be developed using a Simulink GUI, and then converted into an executable code through the Simulink Real-Time Workshop. Exploiting the open control interface, the executable code runs in real-time dialogue with the IRC5 controller with a frequency of 250 Hz, feeding the joint references computed by the external PC to the low-level control. Thanks to the bidirectionality of the interface, the external control can also use as inputs a number of information about the robot state coming from the industrial controller (e.g., motors and joints actual and commanded positions, velocities, and torques).

B. Distance Sensors

A distributed distance sensor prototype, based on the concepts described in Sections II and III, has been designed and deployed on the robot. It makes use of 20 off-the-shelf infra-red light-emitting diode (LED) sensors, model Sharp GP2Y0A02YK, as sensor spots (Fig. 7). Their working principle is based on triangulation, so that the output voltage is inversely proportional to the distance between sensor and obstacle. The voltage signals of the sensor spots are acquired through a National Instrument PCI 6071E board.

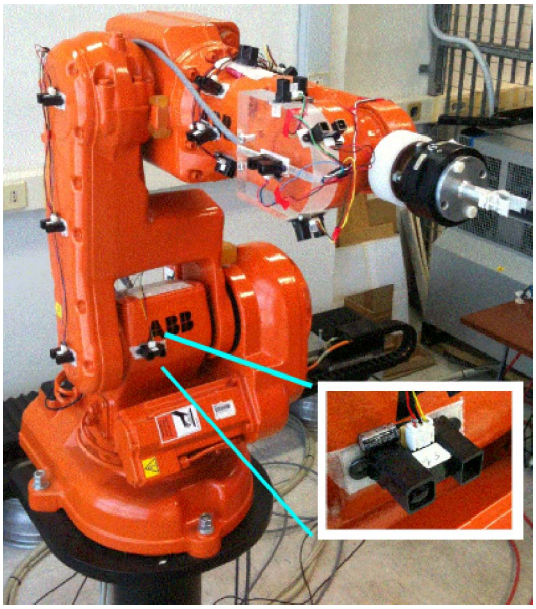


Fig. 7. ABB IRB 140 robot equipped with 20 Sharp IR-LED sensor spots.

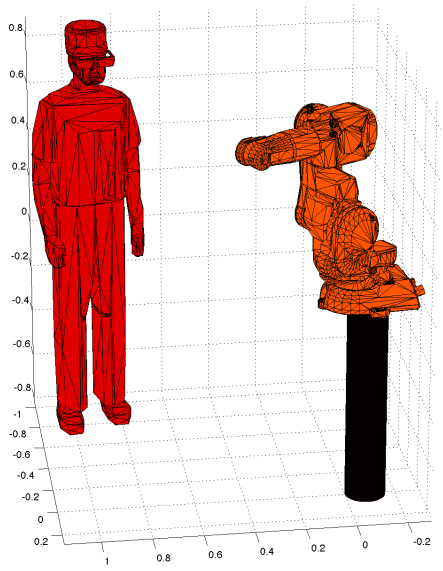


Fig. 8. 3-D models of the ABB IRB 140 and of the worker dummy, used for the Monte Carlo simulation computing the spots optimal placement.

Through this board, it is possible to acquire all of the 20 sensor spots' measurements with a 12-bit DAC and a sampling speed of 1.25 MSamples/s. The external PC connected to the ABB IRC5 controller is also interfaced with the acquisition board through appropriate analogy drivers, which sample the signals at a 250-Hz frequency. In this way, the acquired signals can be used as inputs by the external control.

The nonlinear characteristic curve relating the sensor's voltage output with obstacle's distance has been experimentally identified for each of the sensor spots in the range 20–80 cm. The nominal measuring range is 20–150 cm, but only the 20–80-cm range has been considered due to a low signal/noise ratio for higher distances. Some analog and digital filtering on the output voltage was also necessary. Fig. 6 shows a conceptual sketch of the hardware and software integration that has been realized for the experiments described in Section VI.

C. Sensor Spots Placement

The 20 sensor spots have been distributed along the robot surface through the optimization procedure described in Section III. First, a meshed model of the IRB 140 has been considered. Some candidate regions, suitable for spot placement in terms of size and accessibility, have been selected on the robot's meshed surface (Fig. 3), and the maximum admissible spot positions on every region have been identified, for a total of 254 possible spots location. The obstacle to be detected by the robot was represented by a dummy of a human being, free to occupy every position in the robot workspace (Fig. 8). A Monte Carlo simulation has then been performed, letting the robot assume randomly generated configurations $\mathbf{q}$, while the human dummy occupied an equally randomly generated position of the workspace. In particular, a number of 3000 robot/obstacle combinations have been simulated. For each robot/obstacle combination, knowing the current distribution $\mathbf{y}$ of the 20 spots, the detection of the dummy

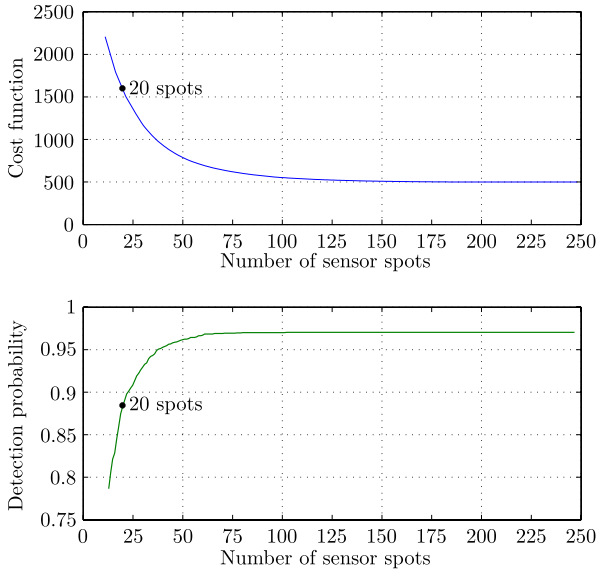
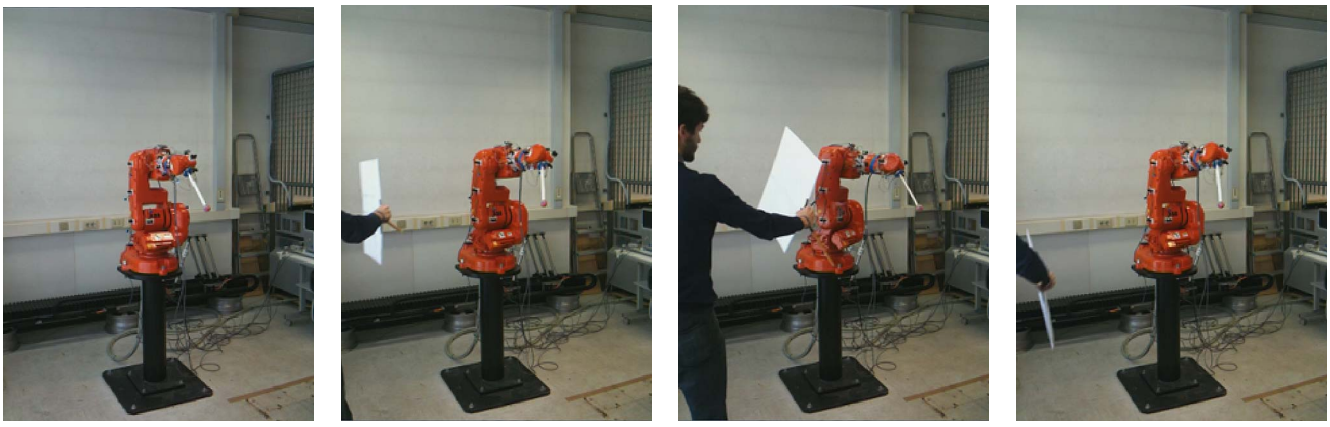


Fig. 9. Values of the cost function and of the detection probability, computed through the Monte Carlo simulation, with respect to the number of spots constituting the distributed distance sensor.

is evaluated through the simulated sensor measures, and the corresponding value of the cost function CF ($\mathbf{y}$) is computed. The optimization procedure (see [30]) allows to select those 20 nodes, among all the 254 possibilities, that minimize CF when hosting the sensor spots. Fig. 9 shows some of the results of [14] concerning the distributed sensor prototype. In particular, the detection probability and the value of the cost function are reported as a function of the number of spots constituting the distributed sensor, from a minimum of one spot to a maximum of 254 spots. It can be seen that there exists a threshold in the number of spots above which the distributed sensor's capability of detecting the dummy does not increase significantly. When using 20 spots, the probability



(a) (b) (c) (d)

Fig. 10. Snapshots taken from the first experiment (fixed references). An obstacle is detected by the distributed sensor. For increasing levels of danger, the robot first releases the orientation task $\mathbf{x}_{d,L}$ and then the position task $\mathbf{x}_{d,H}$. When danger decreases, tasks are resumed in the reverse order. (a) Starting configuration. (b) $\mathbf{x}_{d,L}$ released. (c) $\mathbf{x}_{d,H}$ released. (d) $\mathbf{x}_{d,H}$ retaken, $\mathbf{x}_{d,L}$ still released.

of detecting the obstacle (i.e., the number of simulated cases in which at least one spot detects the dummy) is 89%. Fig. 7 shows the IRB140 robot covered by the 20 IR-LED sensors in their optimal configuration. For further details about the detecting capability of the distributed sensor prototype, please refer to [14].

VI. Experimental Validation

Exploiting the hardware-software architecture outlined in Section V and the distributed sensor described in Sections II and III, the control strategy presented in Section IV has been tested. In the following, some details on the selection of the controller parameters are first given. Then, two simple experiments are presented, with a human holding a card board as an obstacle to be detected by the distributed distance sensor. Finally, a more complex and significant demonstration is presented, with the robot performing a pick and place task, while humans move around it.

The values of the required parameters (i.e., $\mathbf{M}$, $\mathbf{D}$, $\mathbf{K}_H$, $\mathbf{K}_L$, Δ_{th}, $\Delta_{\text{th},f}$, ϵ, and ϵ_f) have been selected through experimental tuning. An automatic procedure for parameters selection is in fact still an open issue. Nevertheless, we here provide some guidelines for the selection. $\mathbf{K}_H$ and $\mathbf{K}_L$ have been chosen as diagonal matrices, with $\mathbf{K}_H = k_H \mathbf{I}_H$ and $\mathbf{K}_L = k_L \mathbf{I}_L$, where $\mathbf{I}_i$ is the identity matrix of adequate dimensions and $k_H, k_L > 0$. Note that when dealing with a CLIK algorithm for prioritized tasks, the choice of k_H and k_L may have an influence on the system's stability (see [41]). For our experiments, we selected $k_H = 1$ and $k_L = 10$ and we never experienced instability issues. Thresholds $\Delta_{\text{th}} = 11.5$ and $\Delta_{\text{th},f} = 7.5$ have been tuned according to CDF_n computed through sensor's measurements, and they obviously depend on the danger field parameters (for some details on such parameters the reader may refer to [28]). $\epsilon_{\text{th}} = 0.15$ and $\epsilon_{\text{th},f} = 0.25$ allow a smooth transition between the controller behavior of Table I. Finally, the values of $\mathbf{M}$ and $\mathbf{D}$ affect the dynamic response of the manipulator when subject to the virtual external torque $\boldsymbol{\sigma}$. We chose both matrices as diagonal for simplicity, in order to obtain a decoupled behavior. The ratio $d_{\text{ii}}/m_{\text{ii}}$, $i = 1 \ldots n_{\text{DoF}}$ therefore affects the $q_{0,i}$ joint velocity response to a virtual torque input σ_i as in a common first order system.

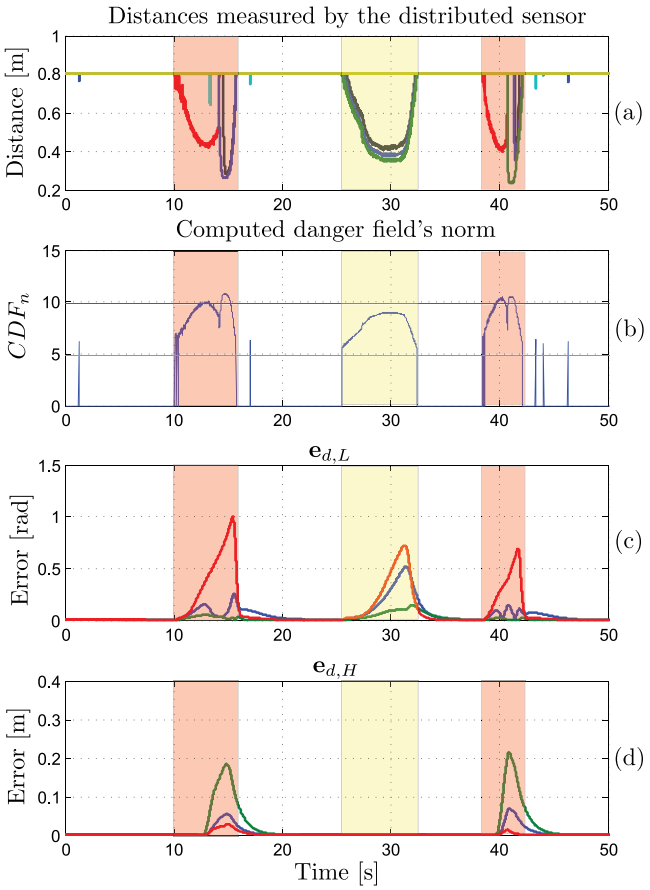
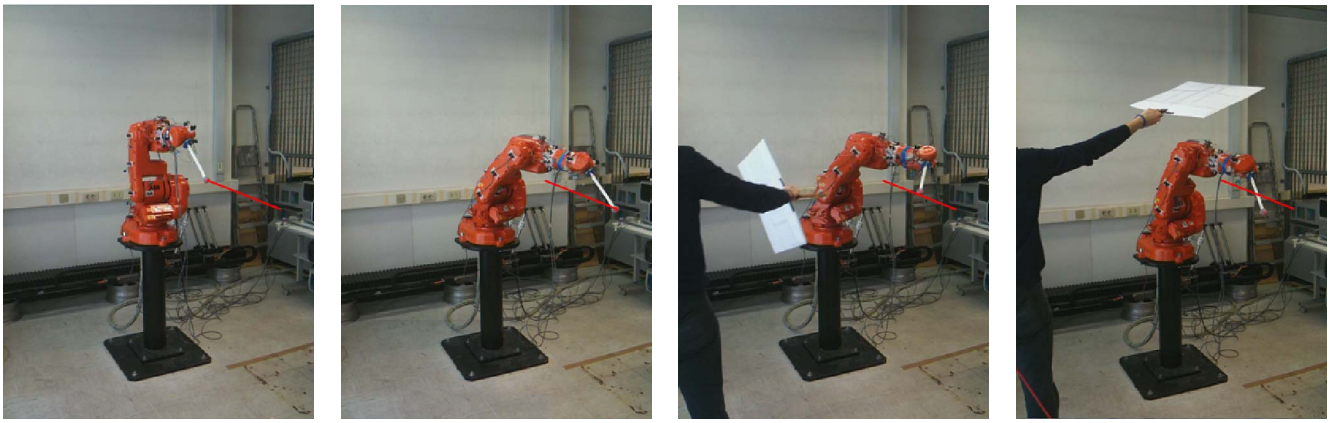


Fig. 11. First experiment results. Medium and high danger situations are highlighted in yellow and red, respectively. (a) Distances measured by the spots (distances above 0.8 m are omitted for clarity). (b) Computed CDF_n. Thresholds $\Delta_{\text{th}}(1-\epsilon)$ (dashed red line) and $\Delta_{\text{th},f}(1-\epsilon_f)$ (dashed green line) are also reported. (c) Orientation errors norm $\mathbf{e}_{d,L}$ (ϕ in blue, θ in green, and ψ in red). (d) Position errors norm $\mathbf{e}_{d,H}$ (x in blue, y in green, and z in red).



(a) (b) (c) (d)

Fig. 12. Snapshots taken from the second experiment (varying references, robot path highlighted in red). As for the first experiment, for increasing levels of danger, the robot first releases the orientation task $\mathbf{x}_{d,L}(t)$ and then the position task $\mathbf{x}_{d,H}(t)$. (a) Initial configuration. (b) Final configuration. (c) $\mathbf{x}_{d,L}$ released. (d) $\mathbf{x}_{d,H}$ released.

A. Testing of the Control Strategy

In the first experiment, a tool was fixed to the robot. With reference to the nomenclature introduced in Section IV, the main task consists in maintaining the tool tip's position and orientation (defined in terms of Euleur angles ϕ, θ, and ψ) fixed to the reference values $\mathbf{x}_d$. The task is subdivided into two tasks with different priorities (Section IV-B). The high priority task, specified by the 3-by-1 vector $\mathbf{x}_{d,H}$, in this case contains the three tool tip's reference positions, while the 3-by-1 vector $\mathbf{x}_{d,L}$ of the low priority task contains the three orientation references of the tool. Note that, as the robot has six degrees of freedom, no evasive motion can be performed while both the high and the low priority tasks are active.

This experiment was carried out with a human entering the robot workspace carrying a card board as obstacle and approaching the manipulator from different directions. Fig. 10 shows some snapshots taken during the first experiment. Fig. 11 reports the results of the experiment. In particular, the distances detected by the LED sensors, the corresponding values of CDF_n and its thresholds $\Delta_{\mathrm{th},f}$ and Δ_{th} (reduced through the smoothing parameter ϵ_f and ϵ, respectively), the error norm on the reference tool's orientation $\mathbf{e}_{d,L} = |\mathbf{x}_{d,L} - \mathbf{x}_L|$, and the error norm on the tool tip's position $\mathbf{e}_{d,H} = |\mathbf{x}_{d,H} - \mathbf{x}_H|$ is reported. For clarity, distances greater than 0.8 m have been omitted.

All the different behaviors summarized in Table I are recognizable. Medium danger level situations are highlighted in yellow, while high danger level situations are in red. For example, when $t < 10\,s$, the danger level is low and both the position and the orientation tasks are maintained (no errors arise in Fig. 11). This particular situation is shown in snapshot of Fig. 10(a).

Between $t = 25\ s$ and $t = 32\ s$, a medium danger level situation can be observed. The low priority orientation task is released and corresponding errors arise (Fig. 11). The evasive motion exploits the now-redundant three degrees of freedom to modify the manipulator pose, without causing position errors (Fig. 11). Such a situation is shown in snapshot of Fig. 10(b). When CDF_n decreases again below the threshold ($t > 32\ s$), the $\mathbf{e}_{d,L}$ is brought back to 0 (Fig. 11).

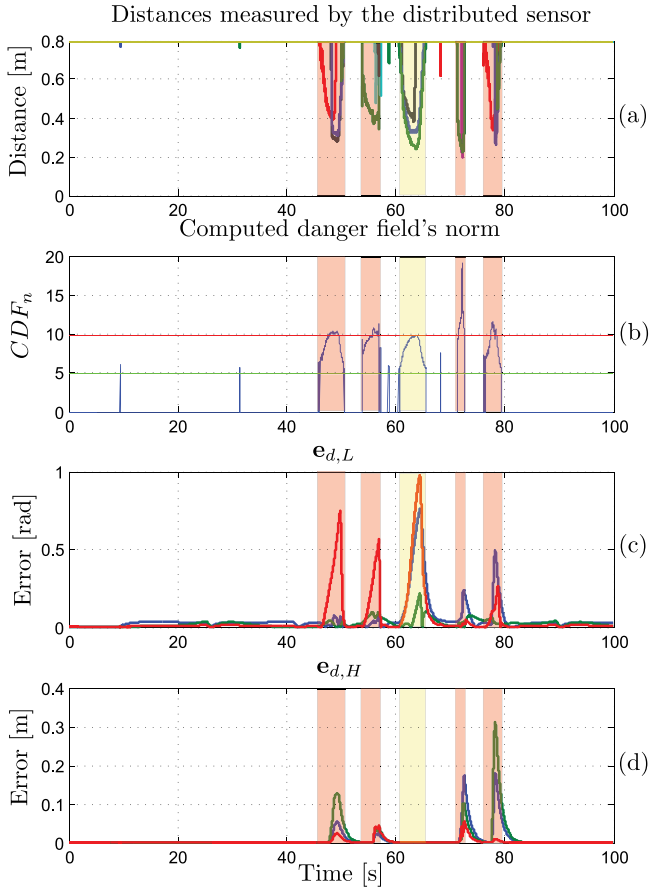


Fig. 13. Second experiment results. Medium and high danger situations are highlighted in yellow and red, respectively. (a) Distances measured by the spots (distances above 0.8 m are omitted for clarity). (b) Computed CDF_n. Thresholds $\Delta_{\mathrm{th}}(1-\epsilon)$ (dashed red line) and $\Delta_{\mathrm{th},f}\left(1-\epsilon_f\right)$ (dashed green line) are also reported. (c) Orientation errors norm $\mathbf{e}_{d,L}$ (ϕ in blue, θ in green, and ψ in red). (d) Position errors norm $\mathbf{e}_{d,H}$ (x in blue, y in green, and z in red).

A high danger level situation is triggered, for example, for $39\ s < t < 41\ s$ and for $10\ s < t < 15\ s$. Both the orientation and the position tasks are released, so that both $\mathbf{e}_{d,L}$ and

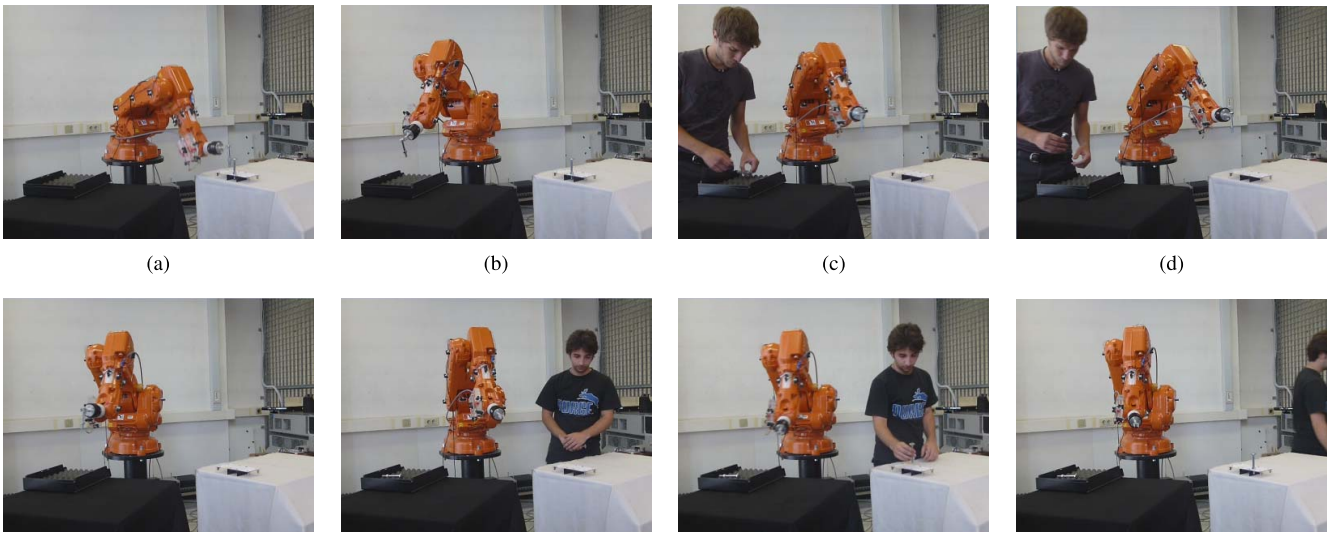


Fig. 14. Snapshots taken from the third experiment. The ABB IRB 140 robot is performing a pick-and-place task. The control strategy avoids collision with humans, detected by the distributed sensor in different locations of the robot workspace. (a) IRB 140 taking a screw from the support. (b) IRB 140 dropping the screw into the box. (c) Human takes the screw while the robot is arriving with another screw. (d) Evasive action prevents the robot from colliding with the human. (e) Robot resumes its task, dropping the screw. (f) Another human approaches the robot from a different side. (g) Evasive action prevents the collision with the human. (h) As the human walks away, the robot retakes its task.

$\mathbf{e}_{d,H}$ arise (Fig. 11). The evasive motion exploits all the six degrees of freedom to steer the robot, as shown in snapshot of Fig. 10(c). When the danger decreases again thanks to the evasive action, as for $t \in (41, 42)\, s$, a medium danger situation is restored, so that $\mathbf{e}_{d,H}$ is brought back to $\mathbf{0}$ [Fig. 11 and snapshot of Fig. 10(d)]. Finally, also the orientation task is resumed when the danger level is low again, as for $t > 41\ s$ ($\mathbf{e}_{d,L} = \mathbf{0}$ in Fig. 11).

The second experiment is similar to the first one, with a person carrying a card board as obstacle while moving around the robot. In this experiment, the main task specifies positions and orientations for the tool's tip that vary during time, following a linear path in the Cartesian space. Once the start and end points of such path have been defined, the Cartesian references vectors $\mathbf{x}_d(t)$ and $\dot{\mathbf{x}}_d(t)$ are generated online by the external control through the task's reference generation block of Fig. 6. Fig. 12 shows some snapshots taken during the second experiment, with the linear path followed by the tool's tip highlighted in red. Fig. 13 shows the distances detected by the LED sensors, the corresponding values of CDF_n, and the errors $\mathbf{e}_{d,L}$ alongside $\mathbf{e}_{d,H}$. The same considerations of the first experiment can be made, the difference being that in this case $\mathbf{e}_{d,L}$ never reaches exactly 0, even for low danger level situations. This is due to the projection of $\dot{\mathbf{q}}_{\mathrm{task},L}$ in the high priority task's null-space combined with the value of $\mathbf{K}_L$ of the CLIK algorithm.

B. Human–Robot Coexistence Demonstration

The last experiment documented in this paper concerns a more significant task in the field of HRI from an industrial perspective. The ABB IRB 140 task consists in picking up a screw from a support located on a table by means of a specific tool, moving the screw over another table, and letting it fall into a box. The working cycle continues with the manipulator picking up a different screw from the support, and so on. The main task is specified by defining a number of intermediate position and orientation of the tool, cyclically varying the starting position so as to be able to take all of the screws contained in the support. Note that the robot main task is subdivided into a high priority task, that specifies the tool's tip position, and a low priority task, that defines the tool orientation. Humans can approach the robot from different directions, for example, to take screws from the box or to place new ones in the support. The presence of a human being can be detected, without occlusions, through the distributed sensor. Following the procedure discussed in Section IV and tested in Section VI-A, an evasive action can be performed to enhance the human safety. In the proposed experiment, the robot is performing its task with no workers in its proximity, as shown in snapshots of Fig. 14(a) and (b). A human then approaches the manipulator to take a screw from the box, while the robot is carrying another screw toward the same box [snapshot Fig. 14(c)]. When the distributed distance sensor detects the human, the task is released in accordance with the computed danger level CDF_n and an evasive action is performed, in order to avoid collisions [snapshot Fig. 14(d)]. The worker then moves away, and the robot resumes its main task, dropping the screw in the box [snapshot Fig. 14(e)]. Another worker approaches the table to put a screw in the support [snapshot Fig. 14(f)]. In this case also, when the distributed sensor detects the human, the robot task and the generation of the reference trajectory are suspended. The evasive action is performed, avoiding collision with the human [snapshot Fig. 14(g)]. When the sensor no longer detects the human being, the main task is resumed and the working cycle can proceed [snapshot Fig. 14(h)].

VII. Conclusion

In this paper, we presented a hardware/software solution that aims at enhancing safety in HRI in industrial environments.

A prototype of a distributed distance sensor to be placed onboard, an industrial robot has been developed for this purpose. The placement of the distributed sensor's spot has been obtained through an optimization method that guarantees almost 90% of human detection probability when using only 20 spots. A reactive control strategy has been designed, integrating the danger assessment deriving from sensor measurements in the robot control. The proposed control strategy allows the robot to perform an evasive action that enhances human safety, at the same time maintaining task consistency (completely or in part, depending on the danger level). A task prioritization has also been included in the control scheme to cope with nonredundant robots.

The safety controller has been experimentally validated. The experiments showed how the integration of the distributed sensor into the robot's control system can enhance human safety in an industrial context.

Future research should concern the improvement of the distributed sensor, increasing the number of spots (and thus the detection probability) and their performance, and the integration of the distributed sensor with other monitoring systems (e.g., surveillance cameras or depth space sensors fixed in space), enhancing the perception (and prediction, through image processing) capabilities of the control system.

REFERENCES

[1] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, "Safety evaluation of physical human-robot interaction via crash-testing," in *Proc. Robot., Sci. Syst.*, Jun. 2007, pp. 217–224.

[2] S. Oberer and R. Schraft, "Robot-dummy crash tests for robot safety assessment," in *Proc. IEEE ICRA*, Apr. 2007, pp. 2934–2939.

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

[4] V. Lumelsky, M. Shur, and S. Wagner, "Sensitive skin," *IEEE Sensors J.*, vol. 1, no. 1, pp. 41–51, Jun. 2001.

[5] A. De Santis, B. Siciliano, A. De Luca, and A. Bicchi, "An atlas of physical human-robot interaction," *Mech. Mach. Theory*, vol. 43, no. 3, pp. 253–270, 2008.

[6] N. Hogan, "Impedance control: An approach to manipulation: Part I—Theory," *J. Dyn. Syst., Meas., Control*, vol. 107, no. 1, pp. 1–7, 1985.

[7] T. Arai, H. Ogata, and T. Suzuki, "Collision avoidance among multiple robots using virtual impedance," in *Proc. IEEE/RSJ IROS*, Sep. 1989, pp. 479–485.

[8] T. Tsuji, H. Akamatsu, M. Hatagi, and M. Kaneko, "Vision-based impedance control for robot manipulators," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, Jun. 1997, pp. 1–53.

[9] Y. Nakabo and M. Ishikawa, "Visual impedance using 1 ms visual feedback system," in *Proc. IEEE ICRA*, vol. 3. May 1998, pp. 2333–2338.

[10] L. Bascetta, G. Ferretti, P. Rocco, H. Ardo, H. Bruyninckx, E. Demeester, *et al.*, "Towards safe human-robot interaction in robotic cells: An approach based on visual tracking and intention estimation," in *Proc. IEEE/RSJ Int. Conf. IROS*, Sep. 2011, pp. 2971–2978.

[11] R. Luo and O. Chen, "Wireless and pyroelectric sensory fusion system for indoor human/robot localization and monitoring," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 3, pp. 845–853, Jun. 2013.

[12] F. Flacco, T. Kröger, A. De Luca, and O. Khatib, "A depth space approach to human-robot collision avoidance," in *Proc. IEEE ICRA*, May 2012, pp. 338–345.

[13] G. Fu, P. Corradi, A. Menciassi, and P. Dario, "An integrated triangulation laser scanner for obstacle detection of miniature mobile robots in indoor environment," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 4, pp. 778–783, Aug. 2011.

[14] N. M. Ceriani, G. Buizza Avanzini, A. M. Zanchettin, P. Rocco, and L. Bascetta, "Optimal placement of spots in distributed proximity sensors for safe human-robot interaction," in *Proc. IEEE ICRA*, May 2013, pp. 5858–5863.

[15] D. Um, B. Stankovic, K. Giles, T. Hammond, and V. Lumelsky, "A modularized sensitive skin for motion planning in uncertain environments," in *Proc. IEEE ICRA*, May 1998, pp. 7–12.

[16] S. Wang, J. Bao, and Y. Fu, "Real-time motion planning for robot manipulators in unknown environments using infrared sensors," *Robotica*, vol. 25, no. 2, pp. 201–211, 2007.

[17] K. Terada, *et al.*, "Development of omni-directional and fast-responsive net-structure proximity sensor," in *Proc. IEEE/RSJ IROS*, Sep. 2011, pp. 1954–1961.

[18] K. Ikuta, M. Nokata, and H. Ishii, "General danger-evaluation method of human-care robot control and development of special simulator," in *Proc. IEEE ICRA*, vol. 4. Jan. 2001, pp. 3181–3188.

[19] M. Nokata, K. Ikuta, and H. Ishii, "Safety-optimizing method of human-care robot design and control," in *Proc. IEEE ICRA*, vol. 2. May 2002, pp. 1991–1996.

[20] K. Ikuta, H. Ishii, and M. Nokata, "Safety evaluation method of design and control for human-care robots," *Int. J. Robot. Res.*, vol. 22, no. 5, pp. 281–297, 2003.

[21] D. Kulić and E. Croft, "Pre-collision safety strategies for human-robot interaction," *Auto. Robot.*, vol. 22, no. 2, pp. 149–164, Feb. 2007.

[22] J. Heinzmann and A. Zelinsky, "Quantitative safety Guarantees for physical human-robot interaction," *Int. J. Robot. Res.*, vol. 22, nos. 7–8, pp. 479–504, 2003.

[23] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.*, vol. 5, no. 1, pp. 90–98, 1986.

[24] D.-H. Park, H. Hoffmann, P. Pastor, and S. Schaal, "Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields," in *Proc. 8th IEEE-RAS Int. Conf. Humanoid Robot.*, Dec. 2008, pp. 91–98.

[25] B. Lacevic and P. Rocco, "Kinetostatic danger field—A novel safety assessment for human-robot interaction," in *Proc. IEEE/RSJ Int. Conf. IROS*, Oct. 2010, pp. 2169–2174.

[26] A. Zanchettin, B. Lacevic, and P. Rocco, "A novel passivity-based control law for safe human-robot coexistence," in *Proc. IEEE/RSJ Int. Conf. IROS*, Oct. 2012, pp. 2276–2281.

[27] B. Lacevic and P. Rocco, "Safety-oriented control of robotic manipulators—A kinematic approach," in *Proc. 18th World Congr. IFAC*, Aug./Sep. 2011, pp. 11508–11513.

[28] B. Lacevic, P. Rocco, and A. M. Zanchettin, "Safety assessment and control of robotic manipulators using danger field," *IEEE Trans. Robot.*, vol. 29, no. 5, pp. 1257–1270, Oct. 2013.

[29] M. C. Campi, S. Garatti, and M. Prandini, "The scenario approach for systems and control design," *Annu. Rev. Control*, vol. 33, no. 2, pp. 149–157, 2009.

[30] L. A. Wolsey, *Integer Programming*. New York, NY, USA: Wiley, 1998.

[31] R. M. Murray, S. S. Sastry, and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation*, 1st ed. Boca Raton, FL, USA: CRC Press, 1994.

[32] B. Siciliano, L. Sciavicco, L. Villani, and L. Oriolo, *Robotics: Modelling, Planning and Control*. New York, NY, USA: Springer-Verlag, 2009.

[33] S. Chiaverini, G. Oriolo, and I. Walker, "Kinematically redundant manipulators," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. New York, NY, USA: Springer-Verlag, 2008.

[34] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *Int. J. Robot. Res.*, vol. 6, no. 2, pp. 3–15, Jul. 1987.

[35] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Trans. Robot. Autom.*, vol. 13, no. 3, pp. 398–410, Jun. 1997.

[36] N. Mansard and F. Chaumette, "Directional redundancy for robot control," *IEEE Trans. Autom. Control*, vol. 54, no. 6, pp. 1179–1192, Jun. 2009.

[37] N. Mansard, O. Khatib, and A. Kheddar, "A unified approach to integrate unilateral constraints in the stack of tasks," *IEEE Trans. Robot.*, vol. 25, no. 3, pp. 670–685, Jun. 2009.

[38] L. Sciavicco and B. Siciliano, "Coordinate transformation: A solution algorithm for one class of robots," *IEEE Trans. Syst., Man Cybern.*, vol. 16, no. 4, pp. 550–559, Jul. 1986.

[39] H. Das, J.-E. Slotine, and T. Sheridan, "Inverse kinematic algorithms for redundant systems," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1. Apr. 1988, pp. 43–48.

[40] A. Blomdell, I. Dressler, K. Nilsson, and A. Robertsson, "Flexible application development and high-performance motion control based on external sensing and reconfiguration of ABB industrial robot controllers," in *Proc. Workshop ICRA*, Anchorage, AK, USA, Jun. 2010, pp. 62–66.
[41] G. Antonelli, "Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems," *IEEE Trans. Robot.*, vol. 25, no. 5, pp. 985–994, Oct. 2009.