A computationally efficient safety assessment for collaborative robotics applications

Matteo Parigi Polverini*, Andrea Maria Zanchettin, Paolo Rocco

Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria, Piazza Leonardo Da Vinci 32, 20133, Milano, Italy

Safety during interaction with unstructured and dynamic environments is now a well established requirement for complex robotic systems. A wide variety of approaches focus on the introduction of safety evaluation methods in order to shape a consequent safety-oriented control strategy, able to reactively prevent collisions between the robot and potential obstacles, including a human being. This paper presents a new safety assessment, named kinetostatic safety field, that captures the risk in the vicinity of an arbitrary rigid body "source of danger" (e.g. an obstacle, a human body part or a robot link) moving in $\mathbb{R}^3$. The safety field depends on the position and velocity of the body but it is also influenced by its real shape and size, exploiting its triangular mesh. The introduction of a body-fixed reference frame in the definition of the field provides closed form computability and an effective computation time reduction, that allows for real-time applications. In particular, intensive computations, connected to the specific body geometry, can be performed only once and off-line, ensuring a fast and constant on-line computation time, independently of the number of mesh elements. Furthermore, we combine the safety field concept with a safety-oriented reactive control strategy for redundant manipulators. Our approach allows to enhance safety in several real-time collision avoidance scenarios, including collision avoidance with potential obstacles, self-collision avoidance and safe human–robot coexistence. The proposed control strategy is validated through experiments performed on an ABB FRIDA dual arm robot.

Keywords: Kinetostatic safety field, Safety assessment, Motion control, Physical human–robot interaction, Redundant robots

1. Introduction

Safety in Human–Robot Interaction (HRI) has gained growing relevance in industrial environments, where, in the near future, humans and robots are expected to safely coexist and cooperate, while sharing the same workspace. Clearly, this safety aspect is strictly connected to the crucial task of collision avoidance. Possible collisions, in fact, can occur between a robot and a human being (human–robot coexistence), between a robot and potential obstacles, but also with the own robot structure, e.g. in a dual arm manipulator (self-collision avoidance).

Several methods focus on the assessment of the level of danger or safety, in order to reshape the robot behavior accordingly. In this respect, using repulsive potential fields introduced by Khatib [1] is now a well-established approach to achieve collision avoidance. An application of potential fields with demonstration on the Ranger Dexterous Manipulator can be found in [2]. Herein, repulsive potentials are designed with respect to obstacles, joint limits and singularities in the configuration space. One drawback of such approach is that the potential field does not consider the relative motion between the robot and the obstacles, unlike in [3]. Also for torque controlled manipulators, like the upper body of DLR's humanoid Justin, collision avoidance applications have been developed. Dietrich et al. [4,5] proposed an algorithm for reactive self-collision avoidance based on artificial repulsion potential fields, which extends the work in [6], with the inclusion of a damping design integrating the configuration dependence of the robot. Moreover, they merged the algorithm with a method to incorporate these unilateral constraints into a dynamic task hierarchy.

In [7,8], a novel method for safety assessment, called *danger field*, is proposed. It is essentially based on the potential field method [1], but it considers the robot and not the obstacles as source of danger, 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 [9–11] to shape a danger field-based control strategy that ensures human safety.

Recent state-of-the-art methods aim at achieving real-time safety with formal guarantees by means of set invariance theory and reach-

Received 5 May 2016;
Received in revised form 25 October 2016;
Accepted 15 November 2016
Available online 28 November 2016

---

* Corresponding author.
E-mail addresses: matteo.parigi@polimi.it (M. Parigi Polverini), andreamaria.zanchettin@polimi.it (A.M. Zanchettin), paolo.rocco@polimi.it (P. Rocco).

Published Journal Article available at http://dx.doi.org/10.1016/j.rcim.2016.11.002

ability analysis. In [12,13] the robot motion planning and control problem in a human involved environment is posed as a constrained optimal control problem. A safety index is evaluated using the ellipsoid coordinates (EC) attached to the robot links that represents the distance between the robot arm and the worker. Furthermore, the linear momentum of the center of the robot links towards the direction to the agents is an additional considered factor. The safety index is used as a constraint in the optimization problem so that a collision-free trajectory within a finite time horizon is generated on-line iteratively for the robot to move towards the desired position. To reduce the computational load for real- time implementation, the formulated optimization problem is further approximated by a quadratic problem.

Pereira et al. [14] instead, present a fail-safe control strategy for on-line safety certification of robot movements in a collaborative workspace with humans. This approach plans, predicts and uses formal guarantees on reachable sets of a robot arm and a human obstacle to verify the safety and feasibility of a trajectory in real time.

Alternative approaches include on-line re-planning [15] and real-time collision avoidance methods based on collision models. In [16,17], Täubig et al. present a collision prediction approach that exploits the GJK-algorithm [18] to compute the distance between robot swept volumes. Corrales et al. [19] developed a safety strategy in a real human–robot interaction task, relying on a geometric representation for human operators and robotic manipulators made of a set of bounding volumes based on swept-sphere line primitives. Finally, in [20] real-time collision avoidance is performed through control in the velocity domain, based on introduced cost functions, with experimental validation on the HRP-2 humanoid robot.

In the framework of physical Human–Robot Interaction (pHRI), methodologies and experimental tests are presented for the problem of detecting and reacting to collisions between a robot manipulator and a human being.

In the work by Haddadin et al. [21–23], a human-friendly control architecture has been developed in order to achieve human-friendly behaviors in cooperation tasks. The robot task is defined distinguishing between four major functional modes (autonomous, human-friendly, collaborative, fault-reaction) of the robot potentially working in human vicinity and the information concerning the physical state currently occupied by the human is used to switch between the different functional modes. Though the proposed control architecture does not explicitly use any danger assessment, it turns out to be an effective approach for safe-oriented applications.

More recently, to guarantee interaction even in the absence of direct contact, the use of pointing gestures has been proposed, as well as the integration of vision with force. In this respect, Cherubini et al. [24] propose a framework to develop a human–robot manufacturing cell, requiring direct physical contact between robot and human, that includes trajectory optimization, admittance control and image processing.

Flacco et al. [25,26] propose an approach that evaluates point-to-object distances working in the depth space of a depth camera, recently extended to multiple depth sensors [27]. The distances are used to generate repulsive vectors that are used to control the robot while executing a generic motion task. The real-time performance of the proposed approach is shown by means of collision avoidance experiments.

In this paper, a new safety assessment, the *kinetostatic safety field*, is presented. Our goal is to introduce a safety measure, easily computable for any moving rigid bodies, using its triangular mesh, able to further ensure real-time applicability, independently of the number of mesh elements. The safety field concept is based on the cumulative danger field [7,8] and on the repulsive potential field approach [1] and is meant to overcome most of the limitations connected to these methods, such as the simplified line representation of the source of danger inherent in the danger field definition and the computationally expensive geometric modeling required by potential field-based approaches. In addition, the safety field concept accounts for the relative velocity between the source of danger and the point where the field is computed, which is instead missing in both danger field and potential field concepts. A detailed analysis on the novelty of the contribution with respect to the state-of-the-art approaches is reported in Section 5. Moreover, our approach, combined with a reactive safety-oriented control strategy, presented in Section 3, can provide safety enhancement in various real-time collision avoidance scenarios.

The present paper extends the preliminary work [28] by providing a detailed discussion of the safety field and the related properties, as well as a more complete experimental validation, enhancing the effect of relative velocity in the definition of the field. Furthermore, an accurate description on how to compute the closed form solution of the safety field is provided in Appendix A.

The paper is organized as follows. First, we derive the concepts of elementary and cumulative kinetostatic safety field in Section 2. Section 3 focuses on the integration of the safety field concept in a reactive control strategy. Experiments of real-time collision avoidance on an ABB FRIDA dual arm robot are performed and analyzed in Section 4. Section 5 compares our approach with state-of-the-art techniques and hints at limitations. Finally, concluding remarks and future work directions can be found in Section 6.

2. Kinetostatic safety field

The concept of safety field, recently proposed by the authors in [28], is addressed in this section.

The aim is to define a safety assessment that fulfills the following requirements:

1. it depends on the magnitude and the direction of relative position and relative velocity between a moving rigid body "source of danger" and a generic moving point in space;
2. it depends on the real shape and size of the rigid body "source of danger", exploiting its triangular mesh;
3. it can be efficiently computed in closed form, allowing for real-time applications.

At first, some basic definitions are given for the case of a point mass. An extension to a basic geometric shape (triangle) and rigid bodies is given afterwards.

2.1. Basic definitions

Consider as "source of danger" a generic rigid body moving in $\mathbb{R}^3$ and a local reference frame l (see Fig. 1) such that the position of one of its points T is given by $\mathbf{r}_t = (x_t\ y_t\ z_t)^T$ while its velocity is zero in the introduced frame. We further denote with $\mathbf{r} = (x\ y\ z)^T$, and $\mathbf{v} = (v_x\ v_y\ v_z)^T$ the position and the velocity of a generic moving point in space P, respectively, expressed in local coordinates.

We additionally define:

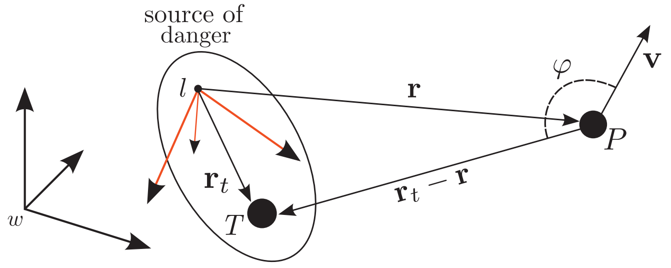


Fig. 1. Elements that play a role in the computation of elementary safety field, expressed in the reference frame l local to the moving rigid body "source of danger".

$$\varphi = \angle(\mathbf{r}_t - \mathbf{r},\ \mathbf{v}) \in [-\pi,\ \pi)$$

as the angle between vectors $\mathbf{r}_t - \mathbf{r}$ and $\mathbf{v}$.

In order to determine the angle φ we can conveniently use the scalar product:

$$cos\varphi = \frac{(\mathbf{r}_t - \mathbf{r})^T \mathbf{v}}{\|\mathbf{r}_t - \mathbf{r}\| \|\mathbf{v}\|}$$

We now denote the elementary Kinetostatic Safety Field (KSSF) induced by the motion of a moving point mass on a moving point in space, as the following scalar function:

$$SF(\mathbf{r}_t, \mathbf{r}, \mathbf{v}) = \|\mathbf{r}_t - \mathbf{r}\|^2 (\gamma - (\mathbf{r}_t - \mathbf{r})^T \mathbf{v}) \tag{1}$$

where γ is a positive constant such that $SF(\mathbf{r}_t, \mathbf{r}, \mathbf{v}) \in \mathbb{R}^+$.

A suitable choice of γ satisfies:

$$\gamma \geq \|\mathbf{r}_t - \mathbf{r}\| \|\mathbf{v}\| \tag{2}$$

Nevertheless, note that for $\gamma \gg \|\mathbf{r}_t - \mathbf{r}\| \|\mathbf{v}\|$ the effect of $(\mathbf{r}_t - \mathbf{r})^T \mathbf{v}$ becomes negligible.

Setting for convenience $\rho_t = \|\mathbf{r}_t - \mathbf{r}\|$ and $v = \|\mathbf{v}\|$, KSSF satisfies the following conditions:

(i) $\frac{\partial SF(\mathbf{r}_t, \mathbf{r}, \mathbf{v})}{\partial \rho_t^2} \equiv \eta > 0,\ \forall\ \rho_t > 0,\ \forall\ v \geq 0,\ \forall\ \varphi \in [-\pi,\ \pi);$

(ii) $\frac{\partial SF(\mathbf{r}_t, \mathbf{r}, \mathbf{v})}{\partial v} < 0,\ \forall\ \rho_t > 0,\ \forall\ v \geq 0,\ \forall\ \varphi \in (-\frac{\pi}{2},\ \frac{\pi}{2});$

(iii) $\varphi\ \frac{\partial SF(\mathbf{r}_t, \mathbf{r}, \mathbf{v})}{\partial \varphi} > 0,\ \forall\ \rho_t > 0,\ \forall\ v \geq 0,\ \forall\ \varphi \in [-\pi,\ \pi).$

The safety field is clearly influenced by the distance, as highlighted in condition (i), but it also captures the module of the velocity vector $\mathbf{v}$, representing the relative velocity between the source of danger and the point where the field is computed, and the declination angle φ. As implied by condition (ii), the safety field decreases with the magnitude of the velocity vector $\mathbf{v}$ if the motion of point $\mathbf{r}$ is directed toward point $\mathbf{r}_t$ (i.e. positive scalar product between $\mathbf{r}_t - \mathbf{r}$ and $\mathbf{v}$). Finally, condition (iii) means that for a fixed velocity magnitude, the safety decreases with decreasing declination angle φ.

Notice that, these conditions are equivalent to those satisfied by the elementary kinetostatic danger field, as shown in [7,8], nevertheless, the safety field concept takes into account both velocity of the source of danger and velocity of the point where the field is evaluated, by means of relative velocity vector $\mathbf{v}$, while the danger field depends on the velocity of the danger source only.

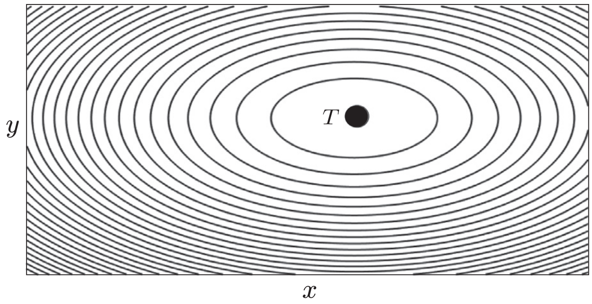
Isosurfaces of the safety field induced by a still and a moving point mass T on a generic still point in the 2D space, are depicted in Fig. 2(a) and in Fig. 2(b), respectively. The safety field increases with the distance between the point mass and the generic point where the field is evaluated and takes low values along the moving direction of T, see Fig. 2(b).

As it will be shown in the following subsection, besides the dependence of KSSF on relative position and relative velocity between the source of danger and the point where the field is computed, the expression introduced in (1) additionally guarantees the existence of a closed form solution of the integral of KSSF over a triangular surface. This was not the case for the previously introduced safety assessment, the danger field, and opens the way to new applications of the field, including its computation for an arbitrarily shaped object, as it will be explained in this paper.

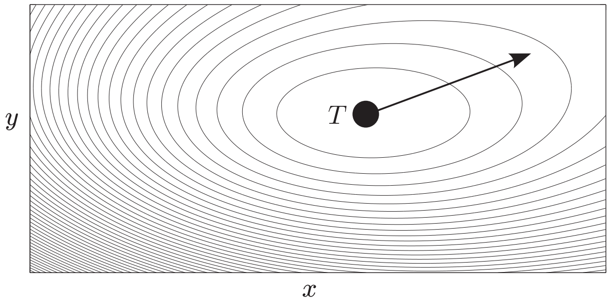
Furthermore, the choice of a local reference frame fixed with the source of danger will provide an effective time reduction in the computation of the KSSF integral over a triangular surface.

2.2. Cumulative kinetostatic safety field of a triangle

The first step for the computation of the safety field of a rigid body, discretized through a triangular mesh, is to extend the definition of the KSSF to a triangle moving in $\mathbb{R}^3$.



(a) Still point mass T.



(b) Point mass T moving with a certain velocity vector (black arrow).

Fig. 2. Contours of a 2D KSSF induced by a point mass T.

In a local body-fixed reference frame such that both linear and angular velocities of the source of danger are zero, once expressed in local coordinates, let $\mathbf{r}_S$ represent the position of a point belonging to the triangular surface S, while $\mathbf{r} = (x\ y\ z)^T$ and $\mathbf{v} = (v_x\ v_y\ v_z)^T$ are position and velocity, respectively, of a generic moving point in space P.

Knowing the absolute position and velocity of the generic moving point in space, represented by $\mathbf{r}_w$ and $\mathbf{v}_w$, respectively (see Fig. 3), and knowing the local reference frame position, linear velocity and angular velocity, expressed once again in the world frame, represented by $\mathbf{r}_{O'}$, $\mathbf{v}_{O'}$ and $\boldsymbol{\omega}$, respectively, one can compute $\mathbf{r}$ and $\mathbf{v}$ (in local coordinates) thanks to the following transformations:

$$\begin{bmatrix} \mathbf{r} \\ 1 \end{bmatrix} = \mathbf{A}_{w \to l} \begin{bmatrix} \mathbf{r}_w \\ 1 \end{bmatrix} \tag{3}$$

$$\mathbf{v} = \mathbf{R}_{w \to l}(\mathbf{v}_w - \mathbf{v}_{O'} - \mathbf{S}(\boldsymbol{\omega})(\mathbf{r}_w - \mathbf{r}_{O'})) \tag{4}$$

where $\mathbf{A}_{w \to l}$ and $\mathbf{R}_{w \to l}$ are the homogeneous transformation matrix and the rotation matrix, respectively, from world to local reference frame, while $\mathbf{S}(\boldsymbol{\omega})$ is the angular velocity tensor skew symmetric matrix.

If $SF(\mathbf{r}_S, \mathbf{r}, \mathbf{v})$ is a KSSF, according to (1), then the Cumulative Kinetostatic Safety Field (CKSSF) is defined as the following surface integral:

$$CSF(\mathbf{r}, \mathbf{v}) = \frac{1}{A} \int\int_S SF(\mathbf{r}_S, \mathbf{r}, \mathbf{v})\ dS \tag{5}$$

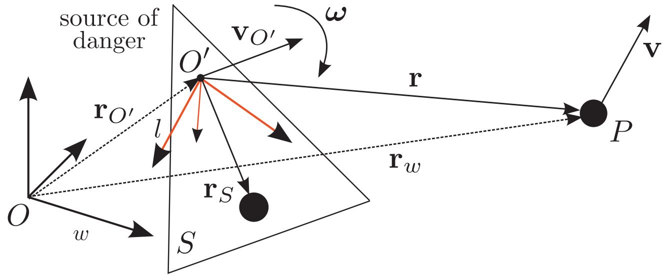


Fig. 3. Elements that play a role in the computation of the Cumulative Kinetostatic Safety Field induced by a moving triangular surface.

where A denotes the area of the triangular surface.

The surface integral is solvable analytically, leading to the following closed form solution:

$$CSF(\mathbf{r},\mathbf{v}) = c_1x^2 + c_2x + c_3y^2 + c_4y + c_5z^2 + c_6z + c_7x^3v_x + c_8y^3v_y + c_9z^3v_z + \cdots + c_{43} \tag{6}$$

where the coefficients c_i for $i = 1 \ldots 6$ together with c_{43} determine the static contribution to the cumulative safety field, named Cumulative Static Safety Field (CSSF):

$$CSSF(\mathbf{r}) = CSF(\mathbf{r},\mathbf{0}) = c_1x^2 + c_2x + c_3y^2 + c_4y + c_5z^2 + c_6z + c_{43} \tag{7}$$

while the coefficients c_i for $i = 7\ldots42$ are used to compute the kinetic contribution of the cumulative safety field. Details on the computation of coefficients c_i in (7) can be found in Appendix A.

It is worth pointing out that the values of coefficients c_i for $i = 1\ldots43$ depend neither on $\mathbf{r}$ nor on $\mathbf{v}$, whereas they are related to the specific triangular surface over which the integration is performed.

It shall also be mentioned that the computation time of $CSF(\mathbf{r},\mathbf{v})$ is deeply influenced by the calculation of coefficients c_i, which turns out to be expensive for real-time applications.

The computational advantage given by the choice of a body-fixed frame (introduced in the definition of KSSF) becomes now more evident: thanks to the introduction of a local reference frame such that the position $\mathbf{r}_S$ of the points belonging to the source of danger is fixed in this frame, the values of c_i remain constant independently from the absolute position and velocity of the source of danger itself. This information is in fact implicitly contained in the position and velocity vectors of the point where the field is evaluated, $\mathbf{r}$ and $\mathbf{v}$, respectively, as can be retrieved from (3) and (4).

Notice that a relevant consequence of this choice of the reference frame, is that, once the coefficients c_i have been computed, $CSF(\mathbf{r},\mathbf{v})$ will be simply affected by the position and velocity of the moving point in space where the field is evaluated.

From a computational perspective, the expensive calculation of the coefficients c_i can be performed just once (off-line), while the value of $CSF(\mathbf{r},\mathbf{v})$ can be updated on-line by simply computing $\mathbf{r}$ and $\mathbf{v}$ and consequently applying (6), therefore leading to a very effective computational time reduction.

2.3. *Cumulative kinetostatic safety field of a rigid body*

Considering as the danger source a rigid body moving in $\mathbb{R}^3$, we define once again a body-fixed reference frame such that the danger source linear and angular velocities are zero expressed in local coordinates. Now, by exploiting the definition in (6) of cumulative safety field induced by a triangle and using a triangular mesh of the danger source, the cumulative safety field generated by the rigid body can be computed by simple superposition of the cumulative safety fields induced by every single triangle that composes its mesh, based on linearity of integration. Consequently the CKSSF of a generic rigid body represented by a mesh of N triangles is given by:

$$CSF(\mathbf{r},\mathbf{v}) = \frac{1}{N}\sum_{i=1}^{N} CSF_i(\mathbf{r},\mathbf{v}) \tag{8}$$

This leads to:

$$CSF(\mathbf{r},\mathbf{v}) = \frac{1}{N}\left(\sum_{i=1}^{N} c_{1i}x^2 + \sum_{i=1}^{N} c_{2i}y^2 + \cdots + \sum_{i=1}^{N} c_{43i}\right) \tag{9}$$

Hence (9), can be simplified to:

$$CSF(\mathbf{r},\mathbf{v}) = \overline{c_1}x^2 + \overline{c_2}y^2 + \cdots + \overline{c_{43}} \tag{10}$$

The general expression for the coefficients in (10) is given by:

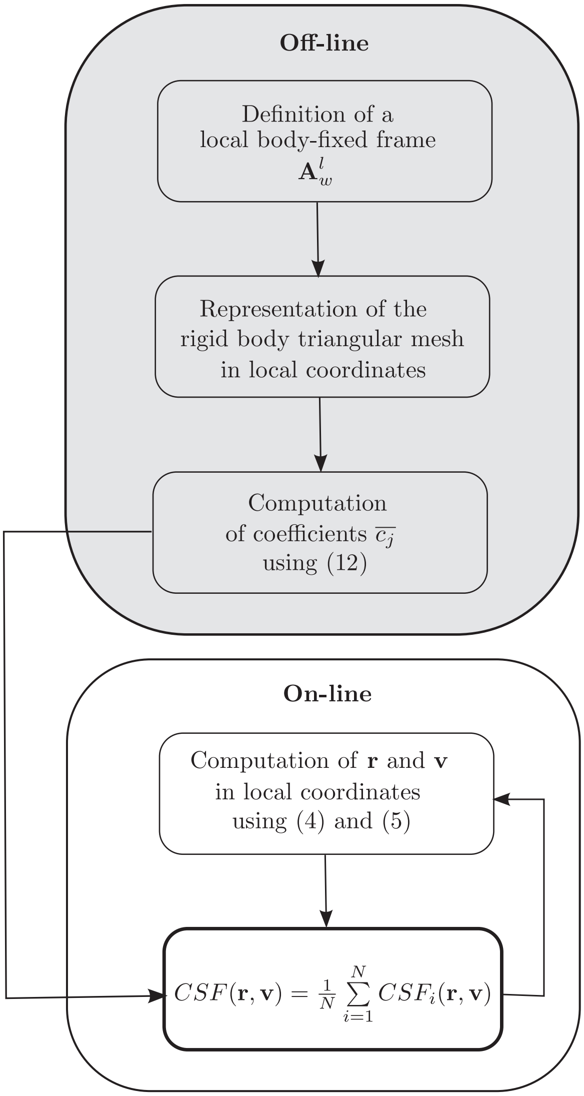


Fig. 4. Methodology for the computation of the Cumulative Safety Field generated by a moving rigid body on an arbitrary moving point in space.

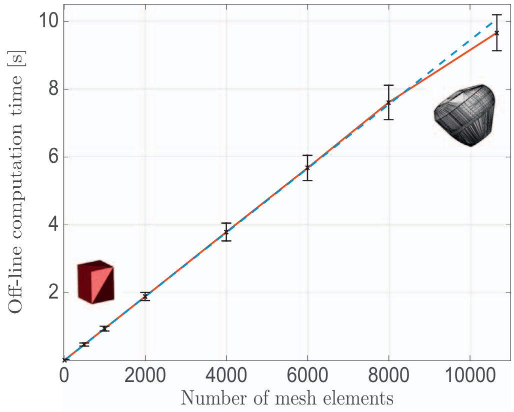
$$\overline{c_j} = \frac{1}{N}\sum_{i=1}^{N} c_{ji}$$

where c_{ji} is the coefficient c_j for $j = 1\ldots43$ related to the cumulative safety field induced by triangle i.

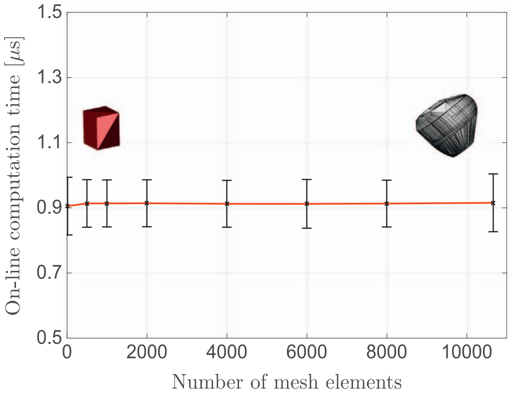
The process to compute the cumulative safety field induced by an arbitrary rigid body source of danger on an arbitrary moving point in space is summarized in Fig. 4.

Remark 1. As already mentioned, thanks to the introduction of a local reference frame fixed to the rigid body, the values of coefficients c_{ji} and therefore the values of coefficients $\overline{c_j}$, are constant. Thus, the computational intensive calculation of $\overline{c_j}$ can be performed only once and off-line (see Fig. 4), whereas the value of $CSF(\mathbf{r},\mathbf{v})$ can be updated on-line by simply computing $\mathbf{r}$ and $\mathbf{v}$ in local coordinates and applying (8).

Mean and standard deviation of both off-line and on-line computation times have been obtained on an Intel Core i7-4650U Processor (1.7 GHz), for the safety field evaluated on the same point in space, generated by a set of sources of danger with an increasing number of mesh elements; from a box (12 elements mesh) to a robot link (10,662 elements mesh). They are shown in Fig. 5(a) and in Fig. 5(b), respectively. Note that the off-line calculation time of coefficients $\overline{c_j}$, see Fig. 5(a), is with good approximation linearly dependent on the



(a) Mean and standard deviation of the off-line computation times of coefficients $\overline{c_j}$ (red solid line) and approximating linear trend (blue dashed line).



(b) Mean and standard deviation of $CSF(\mathbf{r}, \mathbf{v})$ on-line computation times.

Fig. 5. $CSF(\mathbf{r}, \mathbf{v})$ computation times for a set of sources of danger with increasing number of mesh elements. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

number of mesh elements N, i.e.

$$T(\overline{c_j}) \approx O(N).$$

On the other hand, the on-line computation time (approx. $0.9\ \mu s$) is independent of the number of triangles composing the mesh.

The possibility to compute off-line the coefficients $\overline{c_j}$ along with the fast on-line computation time, clearly allows for real-time applications. Furthermore, this also leads to another notable consequence: there is no upper bound in the number of triangular mesh elements of the rigid body, since the number of considered triangles affects off-line computation only, while on-line computation time remains constant.

Remark 2. The field $CSF(\mathbf{r}, \mathbf{v})$ is by definition a scalar quantity. Nevertheless a vector field can easily be constructed upon it. The most natural way to do so is by using its gradient $\nabla CSF(\mathbf{r}, \mathbf{v})$[1], i.e.

$$\overrightarrow{CSF}(\mathbf{r}, \mathbf{v}) = CSF(\mathbf{r}, \mathbf{v}) \frac{\nabla CSF(\mathbf{r}, \mathbf{v})}{\|\nabla CSF(\mathbf{r}, \mathbf{v})\|}$$

Notice that also $\nabla CSF(\mathbf{r}, \mathbf{v})$ is closed form computable.

Remark 3. At this point, it is worth pointing out the properties of the safety field concept:

1. our safety assessment can be applied to any arbitrary rigid body source of danger, described by its triangular mesh i.e. an obstacle, a human body part or a robot link;
2. fast safety field on-line computation time (approx. $0.9\ \mu s$) allows for real-time applications and is independent on the number of the source of danger triangular mesh elements, thus providing the capability of handling triangular meshes with growing number of elements;
3. an advantage of the proposed approach is the possibility to create libraries containing the coefficients characterizing the safety field generated by typical danger sources, to be used in both simulated and real contexts.

3. Safety field based control

This section provides a safety-oriented control strategy for redundant manipulators, based on the introduced concept of safety field and developed entirely at kinematic level. The kinematic redundancy is exploited for simultaneous task performance and collision avoidance, including: collision avoidance with external obstacles, self-collision avoidance and human–robot coexistence.

The control structure is fundamentally based on the danger field-based control proposed in [9,10]: a safety-oriented reactive control strategy that ensures the decrease of the danger itself w.r.t. a simple danger assessment, here modified in order to be consistent with the safety field concept, and extended to enable task relaxation and recovery, based on the safety level.

First, we focus on some general ideas concerning the use of the safety field concept for control purposes. Afterwards, we introduce our safety field based reactive control.

3.1. General principles

Apart from being a simple and flexible danger assessment, the cumulative safety field can be used in the control of a multi-DOF redundant robot in order to ensure the increase of the safety level, while performing a prescribed robot task.

The basic idea is to evaluate the safety field, generated by a rigid body source of danger, on several points of interest along the robot arm. This information is exploited in order to shape the displacement, expressed in the world frame, of the considered points. The safety field-based displacement should ensure an evasive motion of the robot arm away from the source of danger, in order to prevent possible collisions between the robot and the source of danger itself.

Note that, since the evasive displacements are applied to the robot arm, i.e. a set of rigid bodies endowed with a single actuator each, they can be conceived as applied to one point per link. With this respect, as it will be shown in Section 3.2, a suitable choice consists in selecting the robot links' centers of mass.

In order to enable collision avoidance between a robot and an external obstacle, e.g. a box, we can consider the box as the source of danger and apply the evasive displacements on the robot links' centers of mass (see Fig. 6). The evasive displacements will be the result of the safety field generated by the danger source (the box) and evaluated on the selected points. Notice that, the same principle can be applied to multiple obstacles.

Furthermore, in the specific case of a dual-arm robot, for which self-collision avoidance turns out to be an additional safety issue, we can also consider as sources of danger the links on each arm of the manipulator. In this way, the resulting evasive displacements applied on each arm will be the result of the safety field generated by an external obstacle and by the links on the opposite arm (see Fig. 7). This simultaneously allows for both collision avoidance with obstacles and self-collision avoidance. Note that, when a robot link represents the source of danger, a natural choice would be to use the corresponding Denavit–Hartenberg frame as the body-fixed reference frame.

---

[1] Here $\nabla CSF(\mathbf{r}, \mathbf{v})$ is obtained by partial derivative of $CSF(\mathbf{r}, \mathbf{v})$ with respect to the position variables.

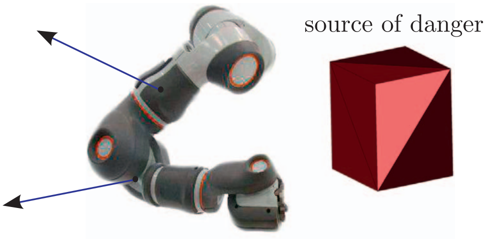


Fig. 6. Evasive displacements (blue vectors) applied on a set of points corresponding to two robot links' centers of mass, induced by the presence of a box obstacle source of danger. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

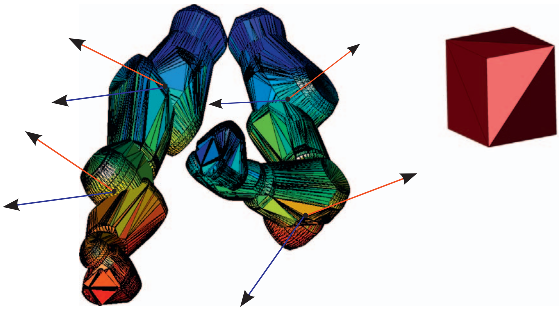


Fig. 7. Evasive displacements applied on a set of points located on each arm links' centers of mass, induced by the presence of a box obstacle (blue vectors) and by the robot links on the opposite arm (red vectors). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

Clearly enough, if we introduce a 3D simplification of the human body parts, considered as sources of danger, our method can further fulfill human–robot coexistence.

Notice that, the robot can be locked in a stationary position when obstacles cause evasive displacements from opposite directions on the robot arm. Nevertheless, notice that, although this behavior is theoretically possible and needs to be investigated in the future, it can arise only if the obstacles' geometrical representation, combined with the obstacles' position and velocity and the own robot state, lead to a perfect cancellation of all the evasive displacement contributions applied on the robot arm.

3.2. *Kinematic control of robotic manipulators*

In the following, we consider a robot manipulator with n joints and generalized coordinates $\mathbf{q} \in \mathbb{R}^n$.

Denoting with m the dimension of the task space, we assume that the robot is kinematically redundant with respect to the task, i.e. $m < n$ and $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{m\times n}$ is the Jacobian associated to the task. Our approach, similar to the one proposed in [9,10], addresses the problem at the kinematic level by using the cumulative safety field of a rigid body source of danger (an obstacle, a human body part or a robot link) to directly modulate the position/velocity commands, by means of evasive displacements. Without loss of generality, the displacement vector $\Delta\mathbf{p}_0$ of a generic point S located on the manipulator's arm, defining its evasive displacement, can be given by:

$$\Delta\mathbf{p}_0 = \mathbf{R}_{l\to w}\frac{1}{CSF(\mathbf{r},\mathbf{v})}\frac{\nabla CSF(\mathbf{r},\mathbf{v})}{\|\nabla CSF(\mathbf{r},\mathbf{v})\|} \tag{11}$$

Note that, $\Delta\mathbf{p}_0$ is a vector, anchored in S, with the intensity $CSF(\mathbf{r},\mathbf{v})^{-1}$ pointing in the direction defined by $\nabla CSF(\mathbf{r},\mathbf{v})$. This clearly ensures a repulsive motion of point S from the source of danger, proportional to the safety level, along the direction of the deepest increase in the safety field level.

Since $CSF(\mathbf{r},\mathbf{v})$ is defined in a local reference frame, the rotation matrix $\mathbf{R}_{l\to w}$ from the local frame l to the world frame w, is exploited in order to express $\Delta\mathbf{p}_0$ in world coordinates. In case of multiple danger sources, the overall displacement that captures the influence of all the sources of danger is given by the following vector:

$$\Delta\mathbf{p}_0 = \sum_{i=1}^{n}\Delta\mathbf{p}_{0,i}$$

Once the evasive displacements have been computed, they can be transformed by means of transposed Jacobian into corresponding increment of joint angles $\Delta\mathbf{q}_0$, resembling the CLIK (Closed-Loop Inverse Kinematics) algorithm.

For a single desired displacement $\Delta\mathbf{p}_0$ of a generic point S located on the manipulator's arm, the corresponding $\Delta\mathbf{q}_0$ is obtained by:

$$\Delta\mathbf{q}_0 = k_p\,\mathbf{J}_{S,v}^T(\mathbf{q})\,\Delta\mathbf{p}_0 \tag{12}$$

Here, $\mathbf{J}_{S,v}(\mathbf{q})$ represents the first three rows of the Jacobian matrix $\mathbf{J}_S(\mathbf{q})$ that are associated to point S while k_p is a positive real parameter.

By division with the sampling time ΔT in Eq. (12), the joint velocity command that yields the evasive displacement can be expressed as:

$$\dot{\mathbf{q}}_0 = k_v\,\mathbf{J}_{S,v}^T(\mathbf{q})\Delta\mathbf{p}_0$$

where $k_v = k_p/\Delta T$.

The overall velocity command can be defined as:

$$\dot{\mathbf{q}}_0 = \sum_S k_v\mathbf{J}_{S,v}^T(\mathbf{q})\Delta\mathbf{p}_0$$

where the sum covers all the relevant points S on the manipulator kinematic chain.

Remark 4. As previously stated in Section 3.1, a suitable choice for S consists in selecting the robot links' centers of mass, independently of the specific robot link geometry and relative position w.r.t. the source of danger. In order to support this statement, we considered a simple line representation of an ABB FRIDA robot link and the triangular mesh representation of the same link. Subsequently, the evasive displacement (11) computed on the link center of mass has been compared with the resultant of the displacements evaluated on a set of points located on the link surface, for the two considered geometric representations. A set of 100 random relative positions and velocities between the source of danger and the robot link (in an area comparable with the robot workspace) has been generated, leading to significantly low vector norm error and angle deviation, reported in Table 1.

The signal $\dot{\mathbf{q}}_0$ is then fed to the standard CLIK algorithm that computes the final position/velocity commands (see Fig. 8).

The analytical expression for the control scheme from Fig. 8 relative to a single arm takes the form:

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_T + [\mathbf{I} - \mathbf{J}^{\dagger}(\mathbf{q})\mathbf{J}(\mathbf{q})]\dot{\mathbf{q}}_0 \tag{13}$$

Here $\dot{\mathbf{q}}_T$ stands for the task-related velocity command and can be computed as:

$$\dot{\mathbf{q}}_T = \mathbf{J}^{\dagger}(\mathbf{q})(\dot{\mathbf{x}}_d + \mathbf{K}_e(\mathbf{x}_d - \mathbf{x}))$$

where $\mathbf{K}_e$ is a positive definite symmetric matrix, $\mathbf{x}_d$ and $\mathbf{x}$ are desired and actual operational space coordinates, $\mathbf{J}(\mathbf{q})$ is a corresponding Jacobian matrix and

Table 1
Comparison between the evasive displacement evaluated on an ABB FRIDA robot link center of mass w.r.t. the resultant displacement evaluated on the same robot link with different geometric representation.

Link geometric representation	Line	Triang. mesh
Vector norm error (max abs.)	4.41%	9.04%
Vector norm error (mean ± SD)	(1.68 ± 1.13)%	(3.63 ± 2.78)%
Angle deviation (max abs.)	1.54°	2.92°
Angle deviation (mean ± SD)	(1.29 ± 0.22)°	(2.45 ± 0.41)°

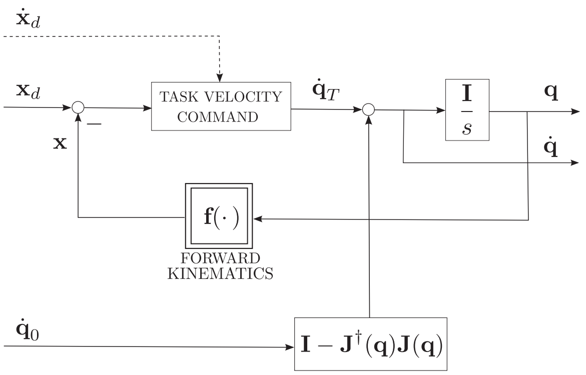


Fig. 8. Feeding signal $\dot{\mathbf{q}}_0$ to CLIK algorithm.

$$\mathbf{J}^\dagger(\mathbf{q}) = \mathbf{J}^T(\mathbf{q})(\mathbf{J}(\mathbf{q})\mathbf{J}^T(\mathbf{q}))^{-1}$$

is the right pseudo-inverse of the Jacobian matrix.

The vector $\dot{\mathbf{q}}_0$ does not affect the motion of the corresponding end-effector because a suitable nullspace projection is performed. Thus, the task consistency is preserved and the end-effector's motion can be performed preventing, simultaneously, the possible collisions with the sources of danger.

To enable the task relaxation, when e.g. the danger exceeds a certain threshold, and the subsequent task recovery, when the danger falls under the threshold, Eq. (13) is modified as follows:

$$\dot{\mathbf{q}} = m\ \dot{\mathbf{q}}_T + [\mathbf{I} - m\ \mathbf{J}^\dagger(\mathbf{q})\mathbf{J}(\mathbf{q})]\dot{\mathbf{q}}_0$$

Setting:

$$SF \equiv \sum_S \|\Delta\mathbf{p}_0\|$$

as the sum of the norms of the evasive displacement vectors applied on the relevant points S on the kinematic chain, the signal m is defined as depicted in Fig. 9, where $\varepsilon < 1$ is a small design constant. SF is primarily used to switch between different values of m that correspond to different behaviors of the robot in task execution. If m=1 the robot is fulfilling its task, while if the danger exceeds a certain limit (m=0) the control leads to an increase in the safety level without task consistency. To enable continuous transition, SF is processed by a non-increasing continuous static function.

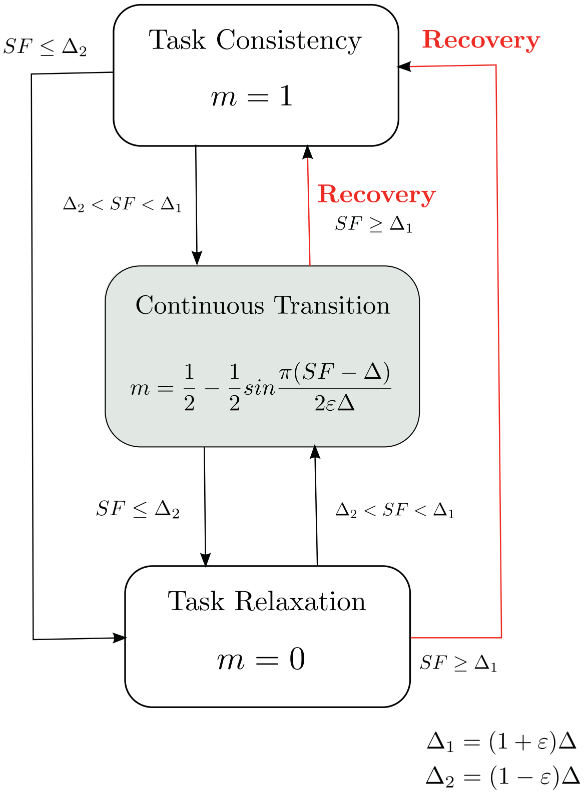


Fig. 9. Selection of m.

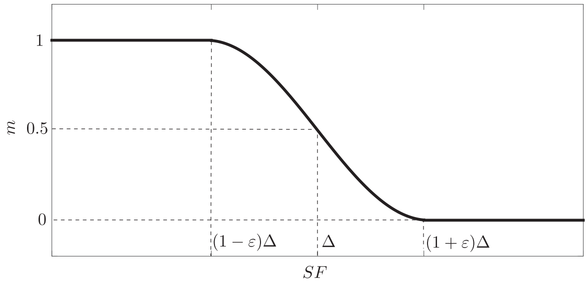


Fig. 10. Impact of ε on function m.

Fig. 10 shows how ε affects the function m. Setting a value for ε close to 1, hence imposing a remarkably smooth transition, may result in a premature task suspension as well as a retarded task resumption. Notice also, that allowing m to take values from the interval [0, 1] improves the system behavior by eliminating chattering and oscillatory motion.

4. Experiments on the ABB Frida

In this section we describe a relevant case study which might benefit from the proposed approach. We here consider for the implementation of our reactive safety field-based control strategy, the 14-DOF dual-arm redundant robot prototype ABB FRIDA. Located inside the robot torso is a control system based on ABB IRC5 industrial controller.

As previously described in Section 3, a dual-arm manipulator is suitable for experimental validation of self-collision avoidance along with human–robot coexistence.

4.1. Experimental setup

An external PC with a Simulink GUI runs under Linux OS with the Xenomai patch, enabling a hard real-time system. It is interfaced to the acquisition board through appropriate Analogy drivers and to the FRIDA controller through an Ethernet-based interface partly developed within the ROSETTA project [29] by Lund University (see [30] for details). Using this interface, one can develop a controller within MATLAB Simulink on the external PC, and then compile it to get the executable code that runs in real time dialog with the IRC5-based controller at a 250 Hz frequency. The corresponding system cycle of 4 ms is an inherent property of the current implementation of the real-time control, thus it cannot be changed. As such, taking into consideration the reduced on-line computation time of the safety field (approx. 0.9 μs), as shown in Section 2, the 4 ms system cycle provides ample margins for all the computations necessary for control.

A possible selection of parameter γ can be performed, according to (2), considering a maximum human–robot distance of 1 m (for the case of human–robot coexistence), the maximum velocity allowed for the tool center point (TCP) by the standard ISO 10218 (250 mm/s) and the average maximum human walking speed (2, 5 m/s). This leads to: $\gamma \geq 2, 75$.

The Reflexxes Motion Library [31] has been exploited to allow real-time task recovery, i.e. when task consistency occurs after task relaxation (see Fig. 9). Fig. 11 shows the input and output parameters of the corresponding algorithm.

To this end, using as target state of motion the desired end-effector position and orientation, and as current state of motion the actual end-

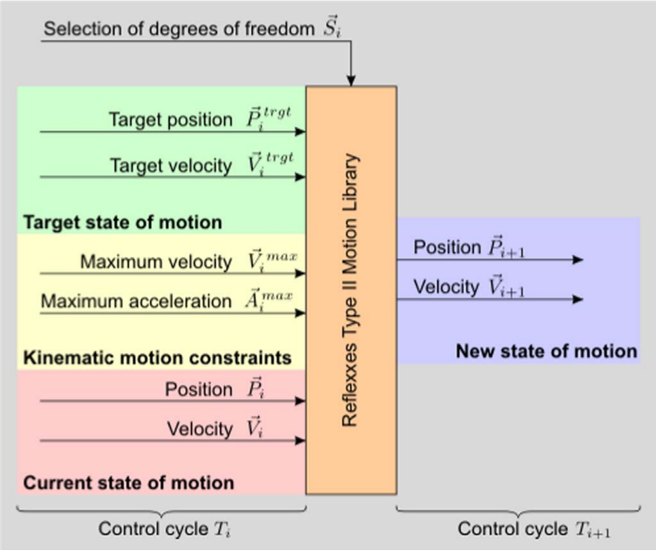


Fig. 11. Interface of the on-line trajectory generation algorithm. Based on the current state of motion and the kinematic motion constraints, a new state of motion is calculated which lies exactly on the time-optimal trajectory reaching the desired target state of motion.

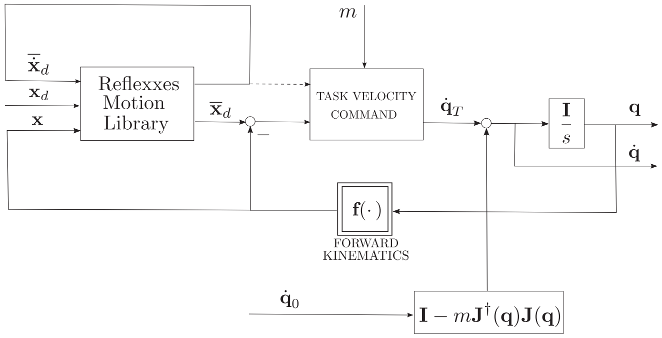


Fig. 12. Integration of the Reflexxes Motion Library for on-line trajectory generation in the CLIK algorithm.

effector position and orientation, given by forward kinematics, the new state of motion, provided by the Reflexxes based on kinematic motion constraints, becomes the end-effector position and orientation reference for the CLIK algorithm depicted in Fig. 12.

Furthermore, for the experimental setup of our reactive control strategy for human–robot coexistence purposes, a range camera (Microsoft Kinect) with the OpenNI drivers has been selected for workspace surveillance [32]. To this end, skeletal tracking method has been exploited in order to obtain the position of a set of points located on the upper human body simplification.

4.2. Experimental validation of self-collision avoidance

Denoting with $\mathbf{q}^L$ and $\mathbf{q}^R$ the vectors of joint coordinates for the left and right arm, respectively, the vector $\mathbf{q}$ of joint coordinates of the manipulator can be organized as:

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}^L \\ \mathbf{q}^R \end{bmatrix}$$

If the task is described in terms of position and orientation for each end-effector, the goal is identified by the following vector:

$$\mathbf{x}_d = \begin{bmatrix} \mathbf{x}_d^L \\ \mathbf{x}_d^R \end{bmatrix}$$

where $\mathbf{x}_d^L$ and $\mathbf{x}_d^R$ stand for the vectors of desired Cartesian position coordinates and unit quaternion related to the end-effector on the left and right arms, respectively. The corresponding Jacobian of the manipulator $\mathbf{J}(\mathbf{q})$ is the block diagonal matrix:

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} \mathbf{J}^L(\mathbf{q}^L) & \mathbf{0} \\ \mathbf{0} & \mathbf{J}^R(\mathbf{q}^R) \end{bmatrix}$$

where $\mathbf{J}^L(\mathbf{q}^L)$ and $\mathbf{J}^R(\mathbf{q}^R)$ are the Jacobian matrices associated to the left and right arm task, respectively.

As shown in Section 3, in order to perform self-collision avoidance in a dual arm robot, one can consider as sources of danger the robot links on each arm of the manipulator and apply the evasive displacements on a set of points located on the opposite arm, corresponding to the robot links' centers of mass (recall Fig. 7).

Thus, the overall velocity command that yields the evasive displacements is given by: $\dot{\mathbf{q}}_0 = \dot{\mathbf{q}}_0^{arm}$, being $\dot{\mathbf{q}}_0^{arm}$ the joint velocity vector corresponding to desired displacements applied on each robot arm, for the safety field generated by the links belonging to the opposite arm.

Experiment #1: A trajectory for the FRIDA's right arm has been programmed in RAPID language, in order to perform "dangerous" sequential approaching motions with respect to the left arm. The robot task, once the safety field based control scheme has been activated, consists in correctly performing FRIDA's right end-effector trajectory (in terms of both position and orientation), see Fig. 13, while the left end-effector maintains the programmed position and orientation, simultaneously ensuring self-collision avoidance. Fig. 14 shows several snapshots of the robot during the experiment.

Notice that, as the right end-effector approaches the opposite end-effector, robot null space motion, exploited by our control strategy, is able to prevent the risk of self-collisions. This behavior becomes evident focusing on the posture of the two robot elbows. As can be seen in Fig. 14(c) and (f) in fact, the distance between the two elbows is maximized, thus limiting the risk of collision with the own robot structure.

4.3. Experimental validation of safe human–robot coexistence

In order to simultaneously perform both self-collision avoidance and human–robot coexistence, when controlling the motion of one of the two arms, we can consider as sources of danger both the robot links on the other arm and a 3D simplification of the human body parts.

Thus, the overall velocity command is now given by:

$$\dot{\mathbf{q}}_0 = \dot{\mathbf{q}}_0^{arm} + \dot{\mathbf{q}}_0^{human}$$

where the two terms represent the joint velocity vectors corresponding to desired displacements applied on a set of points on each robot arm, for the danger induced by the opposite arm ($\dot{\mathbf{q}}_0^{arm}$) and by the human body reconstruction ($\dot{\mathbf{q}}_0^{human}$). To this end, a possible approach to easily compute the safety field induced by a human in the sensor range and thus obtain the corresponding joint velocity vector ($\dot{\mathbf{q}}_0^{human}$), is to consider simplifying boxes of the upper body parts, as sources of

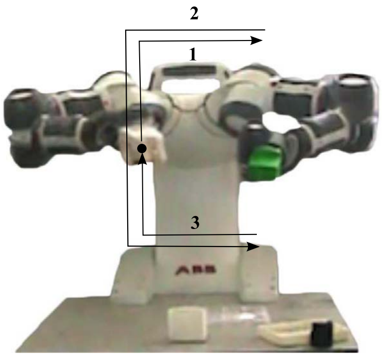


Fig. 13. Programmed FRIDA's right end-effector motion in experiment #1.

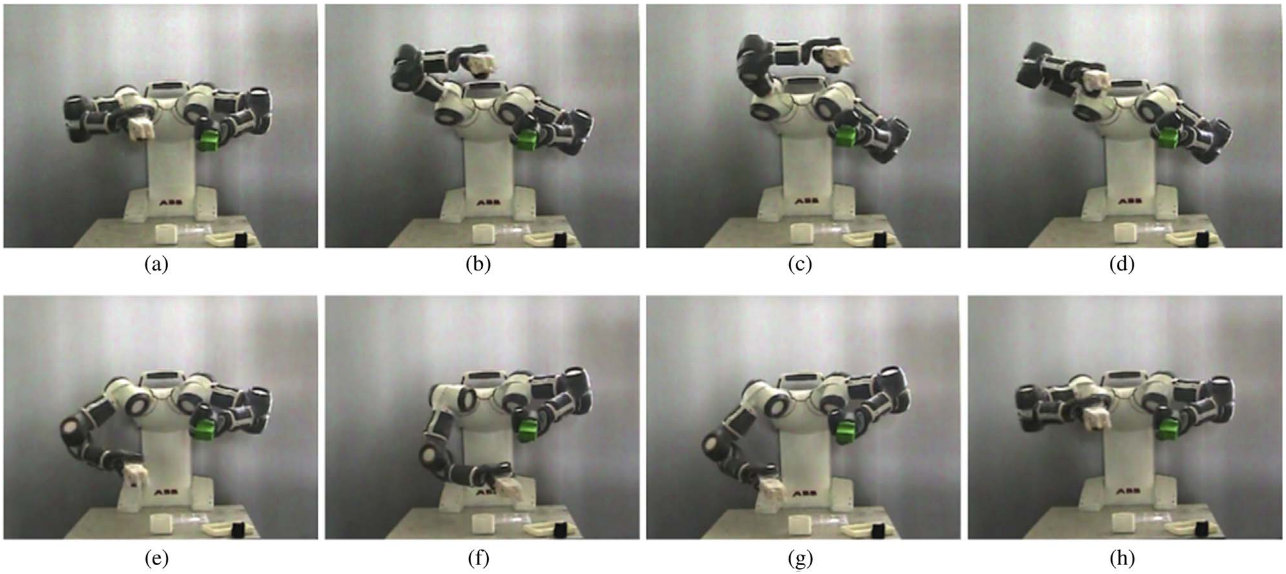

(a) (b) (c) (d)
(e) (f) (g) (h)

Fig. 14. Snapshots during experiment #1: FRIDA's right end-effector performs the programmed trajectory, while the safety field based robot null space motion ensures the prevention of self-collisions.

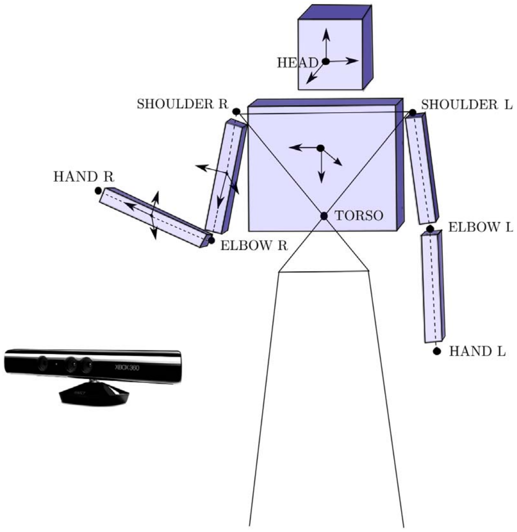


Fig. 15. Upper human body simplification.

danger. In addition, a frame local to each single box (see Fig. 15) can be defined exploiting the position of the set of points on the upper human body representation, returned by Microsoft Kinect skeletal tracking method.

It is worth pointing out that, due to the inaccuracy in human velocity estimation from position data returned by skeletal tracking, only the static contribution to the safety field (7) has been considered in order to determine the corresponding velocity command in experiments #3 and #4.

Experiment #2: In this experiment we want to enhance the effect of a human arm movement with different velocity profiles close to the robot, together with the effect of considering the static contribution of the cumulative safety field (CSSF) in place of the complete kinetostatic expression (CKSSF).

Due to the aforementioned inaccuracy in estimating human velocity from Kinect sensor position data, we here apply a desired trapezoidal velocity profile to a virtual box representing the human lower arm, moving towards the robot right end-effector with constant orientation. In order to better highlight the evasive behavior of the robot right arm, the manipulator's position and orientation tasks are completely relaxed when the virtual human arm starts its approaching motion.

Note that, the end-effector velocity increases as the human moves with higher velocity, see Fig. 16(b) and (e). As a matter of fact, also when only the static contribution of the cumulative safety field is considered, thus neglecting the human velocity, a more reactive robot behavior occurs in the presence of faster human movements. Nevertheless, when the kinetostatic expression of the cumulative safety field is considered, a remarkable 19.3% increase in the minimum robot-obstacle distance can be obtained in the presence of fast human arm motion, see Fig. 16(f), compared to the static case (6.6% increase in the minimum robot-obstacle distance), hence showing experimental evidence of one of the benefits of our approach w.r.t. state-of-the-art potential field-based approaches.

Experiment #3: A human operator, standing in front of FRIDA, moves from different directions towards the robot. The robot task is to keep position and orientation of the two end-effectors constant, while performing the nullspace motion in order to prevent possible collisions with the human and the own robot arms.

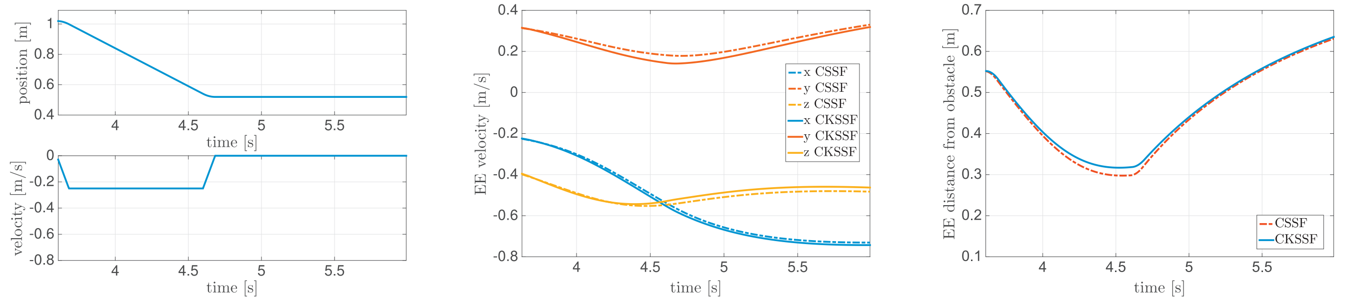
Fig. 17 shows several snapshots of the human moving and the robot performing nullspace motion to prevent possible collisions.

When the operator is outside FRIDA's workspace, see Fig. 17(a), the robot configuration is such that only self-collision avoidance affects the robot behavior (high elbows), whereas, when the operator is standing close to the robot, see Fig. 17(b) and (c), nullspace motion is also affected by the safety field generated by the human rigid body simplification. The combined effect of self-collision avoidance and human–robot coexistence becomes even more evident as the human moves his arms from opposite directions, see for e.g. Fig. 17(d), where the human is moving both his arms towards the manipulator.

Experiment #4: The robot task consists once again in keeping position and orientation of the two end-effectors constant, while this time the human operator intentionally moves his arm onto the robot structure, so that collision will occur. Relaxation of the position task for the two end-effectors has now been introduced in order to prevent the human–robot collision. To handle position task relaxation when the risk of collisions with a human is high, we set for each robot arm:

$$SF \equiv \sum_{S} \|\Delta \mathbf{p}_0^{human}\|$$

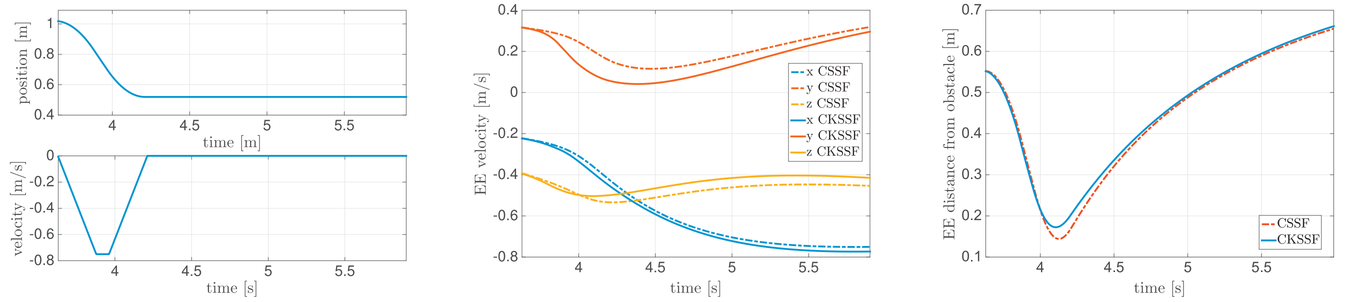
as the sum of the modules of the evasive displacements for all the relevant points on the arm, induced by the presence of a human body, in order to shape the related value of *m*.



(a) Slow human arm movement: virtual human arm position and velocity in the approaching direction (x coordinate).

(b) Slow human arm movement: robot right arm end-effector velocity with CSSF (dashed lines) and CKSSF (solid lines).

(c) Slow human arm movement: distance between the robot end-effector and the virtual human arm with CSSF (red dashed line) and CKSSF (blue solid line).



(d) Fast human arm movement: virtual human arm position and velocity in the approaching direction (x coordinate).

(e) Fast human arm movement: robot right arm end-effector velocity with CSSF (dashed lines) and CKSSF (solid lines).

(f) Fast human arm movement: distance between the robot end-effector and the virtual human arm with CSSF (red dashed line) and CKSSF (blue solid lines).

Fig. 16. Profiles of the virtual human arm motion and robot evasive motion during experiment #2. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

As the operator moves with his arm close to FRIDA, see Fig. 18(b) and (c), robot position task relaxation occurs. Instead, when the operator later retracts his arm, see Fig. 18(e) and (f), task resumption is correctly performed.

Furthermore, Fig. 19 shows the profile of the value of m for FRIDA's left arm and the profile of the corresponding end-effector position coordinates.

Note that, when the human moves close to the robot, m takes values less than 1, hence inducing position task relaxation. Instead, when the human subsequently moves his arm away from the manipulator, m returns to 1 and task recovery occurs, consistent with the scheme in Fig. 9.

Clearly at this stage, real-time trajectory generation, provided by Reflexxes, is crucial in order to allow the end-effector to regain the desired Cartesian position.

5. Discussion: comparison with other approaches and limitations of the method

This section addresses the advantages and drawbacks of our safety assessment compared to the most relevant state-of-the-art approaches, i.e. the repulsive potential field concept [1] and the kinetostatic danger

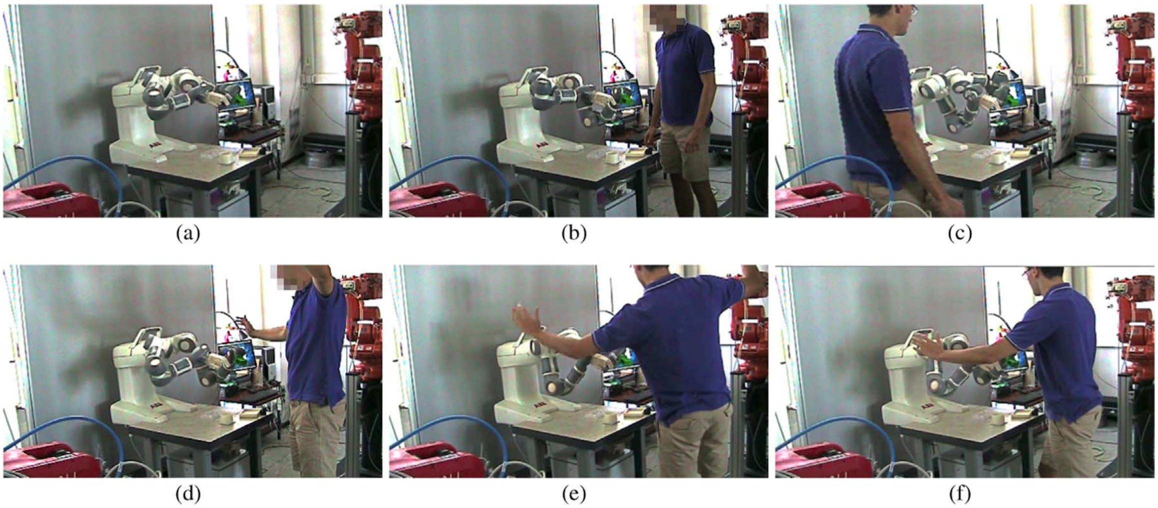


Fig. 17. Snapshots during experiment #3: the human moves close to the robot that performs nullspace motion.

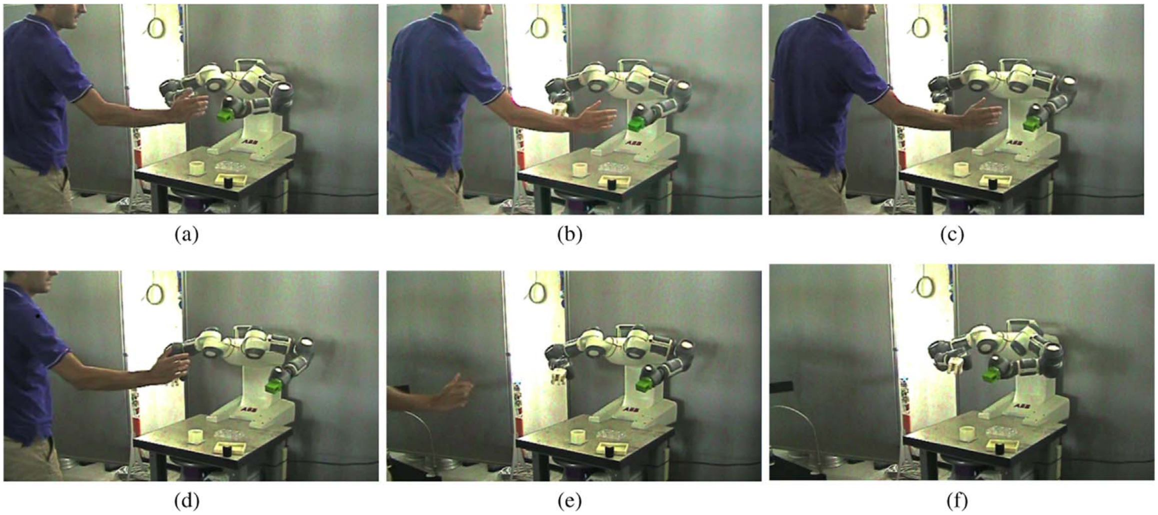


Fig. 18. Snapshot from experiment #4: task relaxation and subsequent recovery.

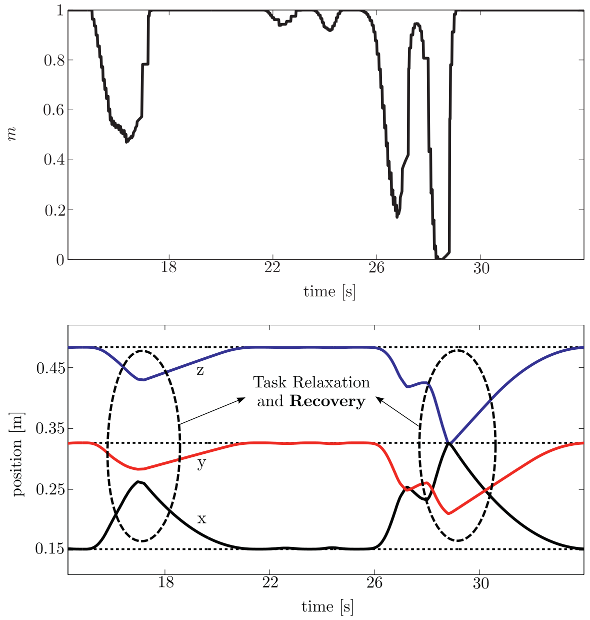


Fig. 19. Profile of the value of m and profile of the end-effector position coordinates for the left arm, in experiment #4.

Table 2
Safety assessments comparison.

Properties	Potential field	Danger field	Safety field
Source of danger	Obstacle/human	Robot	Obstacle/human
Representation	Bounding volume	Linear	Triang. mesh
Relative position	✓	✓	✓
Relative velocity	✗	✗	✓
Safety guarantees	✗	✗	✗

field [7,8], summarized in Table 2.

The connection between the safety field and the virtual potential field used for obstacle avoidance can be easily recognized: both the approaches in fact, consider as source of danger the obstacle to be avoided. On the other hand, in the danger field approach the field is built upon the robot itself, considered as source of danger, and it is evaluated on the point in space where the obstacle is located. Besides the aforementioned similarity, the classical potential field does not capture the velocity neither of the obstacle nor of the robot, while the kinetostatic safety field takes into account both the velocity of the source of danger and the velocity of the point where the field is evaluated, by means of the relative velocity vector $\mathbf{v}$, see Eq. (1). It is also worth pointing out that, in potential field-based collision avoidance methods, obstacle geometric modeling is required. To this end, collision models like bounding volumes [4,5] or swept volumes [16,17,19] can be adopted. In our approach instead, there is no need to introduce a virtual representation of the object source of danger, since its triangular mesh can be directly exploited. Moreover, as the only values needed to update the safety field are the relative position and the relative velocity vector, $\mathbf{r}$ and $\mathbf{v}$, respectively, see Eq. (8), the computationally expensive distance calculation between convex volumes (applying for example the GJK algorithm) can be avoided.

Besides a different choice in the source of danger (the obstacle, in the safety field approach—the robot, in the danger field approach), the safety field represents the consequential evolution of the cumulative danger field concept, extended to an arbitrary moving rigid body in space (a potential obstacle, a human body part or a robot link). Note that, a drawback of the danger field approach is that it does not capture a more complex representation of the source of danger geometry than the line representation: no closed form solution seems to exist, in fact, for the danger field related to easily parameterized surfaces such as cylinders or triangle meshes. Instead, the safety field depends on the shape and size of the rigid body source of danger (discretized through a triangular mesh), rather than only on its length, thanks to the closed form computability of the cumulative safety field generated by a triangular surface. Furthermore, the safety field captures both relative position and relative velocity between the source of danger and the point where the field is computed, whereas the danger field captures only the velocity vector of the danger source, i.e. the robot, while the obstacle is considered still in space.

A few limitations and restriction of our concept can be identified.

As shown in Section 3.2, the evasive displacement evaluated with respect to the robot link center of mass, differs for reasonably small values from the resultant evasive displacement when considering a number of sample points equal to the number of mesh elements. Nevertheless, in order to avoid the selection of candidate points where to evaluate the field, a possible approach could investigate the existence of a closed form solution for the integral of the safety field generated by a rigid body source of danger over an arbitrary rigid body in space, e.g. a robot link.

Other issues concern the lack of formal guarantees of safety and the limitation of our approach to kinematic robot motion constraints. As a matter of fact, a safety constraint with formal guarantees [33,34], together with dynamic, kinematic and task constraints, can be integrated in our methodology by means of hierarchical constraint-based optimization, e.g. following the approach proposed in [35]. In that case the safety field still plays the role of a computationally efficient safety assessment, while the control of the robot is organized in a more elaborate scheme.

Finally, our approach relies on an a priori knowledge of the object source of danger geometry, thus clearly limiting the validity of the proposed methodology in an unknown or cluttered environment, for which object geometry recognition is necessary. On the other hand, this limitation seems less crucial in a typical industrial environment and, furthermore, it does not occur when the robot is the source of danger itself, i.e. for the case of self-collision avoidance, thanks to the availability of commercial robots' meshes.

6. Conclusion

In this paper a new safety assessment, that stands upon the introduced concept of *kinetostatic safety field*, has been given. The safety field captures the risk in the vicinity of an arbitrary rigid body moving in space, intended as a source of danger, and depends on both relative position and relative velocity between the source of danger and the point where the field is computed, along with the shape and size of the danger source itself, using its triangular mesh. Furthermore, the definition of the field w.r.t. a body-fixed reference frame provides closed form computability and allows for real-time applications. Intensive computations, depending on the source of danger geometry, in fact, can be performed one-time-only and off-line, while on-line computation time turns out to be constant and independent of the number of mesh elements. In a second step the introduced safety assessment is merged with a safety-oriented reactive control strategy for multi-DOF redundant manipulators, able to enhance safety in several real-time collision avoidance scenarios, including: collision with potential obstacles, self-collision avoidance and human–robot coexistence. Experimental results on the ABB FRIDA dual arm robot supported our approach.

While this paper specifically focused on real-time collision avoidance through the adoption of a reactive control strategy, there are no limitations to integrate the safety field concept in motion planning algorithms to deal with the same issue. Moreover, future work will include the possibility to regulate the speed of task execution w.r.t. the safety level. Further improvements can be also achieved in terms of alternative task relaxation methods.

Appendix A. Computation of cumulative static safety field of a triangle

This section provides details on how to compute the coefficients c_i occurring in the closed-form solution of the Cumulative Static Safety Field (CSSF) of a triangular surface, according to (7). As a matter of fact, the analytical expression of the Cumulative Kinetostatic Safety Field (CKSSF) in (6), can be as well obtained applying the following procedure and it will be here omitted for brevity.

Consider a triangular surface S, whose non-collinear vertexes coordinates expressed in a local body-fixed reference frame $O'(x_s, y_s, x_S)$ are: $(x_{1_S}, y_{1_S}, z_{1_S})^T$, $(x_{2_S}, y_{2_S}, z_{2_S})^T$ and $(x_{3_S}, y_{3_S}, z_{3_S})^T$.

Let $\mathbf{r}_S$ represent the position of a point belonging to S, while $\mathbf{r} = (x, y, z)^T$ is the position of a generic point in space.

In order to compute the surface integral in (5), let us first find the equation of the plane through the triangle vertexes:

$$ax_S + by_S + cz_S + d = 0 \tag{A.1}$$

where $a, b, c, d \in \mathbb{R}$ can be found e.g. by solving the following system of equations, using Cramer's Rule and basic matrix manipulations:

$$\begin{cases} ax_{1_S} + by_{1_S} + cz_{1_S} + d = 0 \\ ax_{2_S} + by_{2_S} + cz_{2_S} + d = 0 \\ ax_{3_S} + by_{3_S} + cz_{3_S} + d = 0 \end{cases}$$

Subsequently, assuming that the plane (A.1) is not parallel to any of the local frame planes, we can describe the triangular region D, which is the projection of S on one of the frame planes, e.g. the (x_S, y_S)-plane, as the union of two y_S-simple regions:

$$D = D_1 \cup D_2$$

where:

$$D_1\!: \{(x_S, y_S) \in \mathbb{R}^2 | x_{1_S} \le x_S \le x_{2_S},\ \ \hat{y}_{1_S} \le y_S \le \hat{y}_{2_S}\} D_2\!: \{(x_S, y_S) \in \mathbb{R}^2 | x_{2_S} \le x_S \le x_{3_S},\ \ \hat{y}_{1_S} \le y_S \le \hat{y}_{3_S}\}$$

being:

$$\hat{y}_{1_S} = m_1 x_S + b_1 \hat{y}_{2_S} = m_2 x_S + b_2 \hat{y}_{3_S} = m_3 x_S + b_3$$

the three linear equations of the triangle D edges, where the slope coefficients $m_1, m_2, m_3 \in \mathbb{R}$ and the constant terms $b_1, b_2, b_3 \in \mathbb{R}$ are given by elementary algebraic relations.

Finally, according to laws of integration of a function over a triangular surface, the analytical expression of $CSSF(\mathbf{r})$ can be found by solving the following integrals over the two y_S-simple regions D_1 and D_2, leading to:

$$\begin{aligned} CSSF(\mathbf{r}) = CSF(\mathbf{r}, \mathbf{0}) = \frac{1}{A}\int\!\!\int_S SF(\mathbf{r}_S, \mathbf{r}, \mathbf{0})dS = \frac{\alpha}{A}\Bigg[\int_{x_{1_S}}^{x_{2_S}}\left(\int_{\hat{y}_{1_S}}^{\hat{y}_{2_S}} (x_S - x)^2 + (y_S - y)^2 + (\bar{z}_S - z)^2\, dy_S\right)dx_S + \\ \int_{x_{2_S}}^{x_{3_S}}\left(\int_{\hat{y}_{3_S}}^{\hat{y}_{2_S}} (x_S - x)^2 + (y_S - y)^2 + (\bar{z}_S - z)^2\, dy_S\right)dx_S\Bigg] \end{aligned} \tag{A.2}$$

where $\bar{z}_S$ needs to be replaced with:

$$\bar{z}_S = -\frac{ax_S + by_S + d}{c}$$

and the scale factor α is given by

$$\alpha = \sqrt{1 + \left(\frac{\partial \bar{z}_S}{\partial x_S}\right)^2 + \left(\frac{\partial \bar{z}_S}{\partial y_S}\right)^2} = \sqrt{1 + \left(-\frac{a}{c}\right)^2 + \left(-\frac{b}{c}\right)^2}$$

By analytically solving (A.2) and subsequently collecting the terms in the powers of the variables x, y, z, the closed form solution of the CSSF in (7) can be hence obtained, where e.g. the coefficient c_1 of x^2 is given by:

$$c_1 = \frac{\alpha}{2A}((x_{1_S} - x_{2_S})(2b_1 - 2b_2 + m_1x_{1_S} + m_1x_{2_S} - m_2x_{1_S} - m_2x_{2_S}) + (x_{2_S} - x_{3_S})(2b_1 - 2b_3 + m_1x_{1_S} + m_1x_{3_S} - m_3x_{2_S} - m_3x_{3_S}))$$

Note that the coefficients c_3 and c_5 of y^2 and z^2, respectively, have same analytical expression of c_1, while the expression of coefficients c_2, c_4, c_6 of x, y, z, respectively, and the constant term c_{43} has been here omitted for brevity.

Appendix B. Supplementary data

Supplementary data associated with this article can be found in the online version athttp://dx.doi.org/10.1016/j.rcim.2016.11.002.

References

[1] O. Kathib, Real-time obstacle avoidance for manipulators and mobile robots, Int. J. Robot. Res. 5 (1986) 90–98.

[2] N. Scott, C. Carignan, A line-based obstacle avoidance technique for dexterous manipulator operations, in: IEEE International Conference on Robotics and Automation, 2008, pp. 3353–3358.

[3] D.-H. Park, H. Hoffmann, P. Pastor, S. Schaal, Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields, in: IEEE-RAS International Conference on Humanoid Robots, 2008, pp. 91–98.

[4] A. Dietrich, T. Wimbök, H. Täubig, A. Albu-Schäffer, G. Hirzinger, Extension to reactive self-collision avoidance for torque and position controlled humanoids, in: IEEE International Conference on Robotics and Automation, 2011, pp. 3455–3462.

[5] A. Dietrich, T. Wimbök, H. Täubig, A. Albu-Schäffer, G. Hirzinger, Integration of reactive, torque-based self-collision avoidance into a task hierarchy, in: IEEE Transaction on Robotics, vol. 28, 2012, pp. 1278–1293.

[6] A. De Santis, A. Albu-Schäffer, C. Ott, B. Siciliano, G. Hirzinger, The skeleton algorithm for self-collision avoidance of a humanoid manipulator, in: IEEE/ASME International Conference of Advanced Intelligent MEchatronics, 2007.

[7] B. Lacevic, Safe motion planning and control for robotic manipulators-a kinematic approach, (Ph.D. thesis), Dipto. Elettron. Inform. e Bioing., Politecnico di Milano, Milano, 2011.

[8] B. Lacevic, P. Rocco, Kinetostatic danger field - a novel safety assessment for human–robot interaction, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010, pp. 2169–2174.

[9] B. Lacevic, P. Rocco, Safety-oriented control of robotic manipulators - a kinematic approach, in: IFAC World Congress, 2011.

[10] B. Lacevic, P. Rocco, A.M. Zanchettin, Safety assessment and control of robotic manipulators using danger field, IEEE Trans. Robot. 29 (2013) 1257–1270.

[11] A. Zanchettin, B. Lacevic, P. Rocco, A novel passivity-based control law for safe human–robot coexistence, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012, pp. 2276–2281.

[12] C.S. Tsai, J.S. Hu, M. Tomizuka, Ensuring safety in human–robot coexistence environment, in: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2014, pp. 4191–4196.

[13] C. Liu, M. Tomizuka, Algorithmic safety measures for intelligent industrial co-robots, in: 2016 IEEE International Conference on Robotics and Automation, 2016, pp. 3095–3102.

[14] A. Pereira, M. Althoff, Safety control of robots under computed torque control using reachable sets, in: 2015 IEEE International Conference on Robotics and Automation, 2015, pp. 331–338.

[15] E. Yoshida, F. Kanehiro, Reactive Robot Motion using Path Replanning and Deformation, in: IEEE International Conference on Robotics and Automation, 2011, pp. 5456–5462.

[16] H. Täubig, B. Bauml, U. Frese, Real-time swept volume and distance computation for self collision detection, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2011, pp. 1585–1592.

[17] H. Täubig, B. Bauml, U. Frese, Real-time continuous collision detection for mobile manipulators - a general approach, in: IEEE-RAS International Conference on Humanoid Robots, 2012, pp. 461–468.

[18] E.G. Gilbert, D.W. Johnson, S.S. Keerthi, A fast procedure for computing the distance between complex objects in three-dimensional space, IEEE J. Robot. Autom. 4 (1988) 193–203.

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

[20] O. Stasse, A. Escande, N. Mansard, S. Miossec, P. Evrard, A. Kheddar, Real-time (self)-collision avoidance task on a HRP-2 humanoid robot, in: IEEE International Conference on Robotics and Automation, 2008, pp. 3200–3205.

[21] S. Haddadin, M. Suppa, S. Fuchs, T. Bodenmüller, A. Albu-Schäffer, G. Hirzinger, Towards the robotic co-worker, in: International Symposium on Robotics Research (ISRR2009), Lucerne, Switzerland, 2009.

[22] S. Haddadin, S. Parusel, R. Belder, J. Vogel, T. Rokahr, A. Albu-Schäffer, G. Hirzinger, Holistic design and analysis for the human-friendly robotic co-worker, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010, pp. 4735–4742.

[23] S. Parusel, S. Haddadin, A. Albu-Schäffer, Modular state-based behavior control for safe human–robot interaction: A lightweight control architecture for a lightweight robot, in: IEEE International Conference on Robotics and Automation, 2011, pp. 4298–4305.

[24] A. Cherubini, R. Passama, A. Crosnier, A. Lasnier, P. Fraisse, Collaborative manufacturing with physical human–robot interaction, Robot. Comput.-Integr. Manuf. 40 (2016) 1–13.

[25] F. Flacco, T. Kröger, A.D. Luca, O. Khatib, A depth space approach to human–robot collision avoidance, in: 2012 IEEE International Conference on Robotics and Automation, 2012, pp. 338–345.

[26] F. Flacco, T. Kröger, A.D. Luca, O. Khatib, A depth space approach for evaluating distance to objects, J. Intell. Robot. Syst. 80 (1) (2015) 7–22.

[27] F. Flacco, A.D. Luca, Real-time computation of distance to dynamic obstacles with multiple depth sensors, IEEE Robot. Autom. Lett. 2 (1) (2017) 56–63.

[28] M. Parigi Polverini, A.M. Zanchettin, P. Rocco, Real-time collision avoidance in human–robot interaction based on kinetostatic safety field, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2014, pp. 4136–4141.

[29] ROSETTA, EU-FP7 grant no. 230902. URL ⟨http://www.fp7rosetta.org⟩

[30] A. Blomdell, I. Dressler, K. Nilsson, A. Robertsson, Flexible application development and high-performance motion control based on external sensing and reconfiguration of ABB industrial robot controllers, in: Workshop Innovation Robotics Control Architecture Demanding Research Applications, pp. 62–66, 2010.

[31] T. Kröger, Opening the door to new sensor-based robot applications—The Reflexxes Motion Libraries, in: IEEE International Conference on Robotics and Automation, 2011.

[32] OpenNI - the standard framework for 3D sensing, downloaded from ⟨http://http://www.openni.org⟩(March 2013).

[33] A.M. Zanchettin, P. Rocco, Path-consistent safety in mixed human–robot collaborative manufacturing environments, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013, pp. 1131–1136.

[34] A.M. Zanchettin, N.M. Ceriani, P. Rocco, H. Ding, B. Matthias, Safety in human–robot collaborative manufacturing environments: Metrics and control, in: IEEE Transaction on Automation Science and Engineering, vol. PP, 2015, pp. 1–12.

[35] N.M. Ceriani, A.M. Zanchettin, P. Rocco, Collision avoidance with task constraints and kinematic limitations for dual arm robots, in: Menegatti, Emanuele, Michael, Nathan, Berns, Karsten, Yamaguchi, Hiroaki, Intelligent Autonomous Systems 13, Advances in Intelligent Systems and Computing, Springer, 2016