The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Towards a Complete Safe Path Planning for Robotic Manipulators

Bakir Lacevic and Paolo Rocco

Abstract— **We propose a novel method of path planning for robotic manipulators that is based on the tree expansion via bubbles of free configuration space. The algorithm is designed to yield collision-free paths that also tend to minimize a certain danger criterion. This is achieved by embedding a suitably tailored heuristics within the algorithm. For that purpose we use a recently proposed safety assessment based on the concept of the danger field - an easily computable quantity that captures the complete kinematic behavior of the manipulator. Under the assumption that a systematic graph search technique dictates the tree growth, we prove the algorithm's completeness.**

I. INTRODUCTION

In the field of human-robot interaction, the issue of safety is clearly vital. It is triggered mostly by the growing demands on humans and robots to share the same workspace or task whether within an industrial, domestic or any kind of environment. Not surprisingly, a large attention is devoted to this matter in the literature.

Ikuta et al. [1], [2] discuss the minimization of the risk in interaction by means of mechanical design and by means of control. They introduced the first systematic quantitative methods (danger index, safety index, etc.) in safety evaluation, concerning human robot interaction in general. Heinzmann and Zelinsky [3] proposed a control scheme for robotic manipulators that restricts the torque commands that comply to predefined quantitative safety restrictions. They defined a quantity called impact potential as a maximum impact force that a moving mechanical system can create in a collision with a static obstacle. Bicchi et al. [4] presented the variable impedance approach as a mechanical/control co-design that allows the mechanical impedance parameters to change during the task execution. By solving the so-called safe brachistochrone problem, the authors have shown that low stiffness is required at high speed and vice versa. Henrich and Kuhn [5] divide all safety aspects of the robot behavior into four groups (states) that easily fit into the formalism of state transition diagram. A similar mechanism is used by Guiochet et al. [6] to develop quite rigorous framework to facilitate the specification of safety rules used by an independent safety monitor. In [7] and [8] comprehensive overviews of safe human-robot interaction are presented.

Kulic and Croft [9], [10] propose several safety strategies, as a components of an extensive methodology for safe planning and control in human-robot interaction. Several danger indices have been formulated and used as an input to a real-time trajectory generation. A motion strategy consists in minimizing the danger index during a stable robot operation. In [9], the authors tackle safety by minimizing a danger criterion during the path planning stage. Brock and Khatib [11] propose a general framework for motion planning in a human environment.

Typical path planning aims at obtaining a collision-free path in the configuration space (C-space) of a robot that connects a start configuration $\mathbf{q}_{start}$ with a goal configuration $\mathbf{q}_{goal}$ [12]. Planning algorithms that are based on the a priori knowledge of the complete C-space are proven to be exponential in the dimensionality of the C-space [13]. Due to the questionable applicability of such an approach, probabilistic roadmap (PRM) paradigm has gained more popularity [14]. In particular, rapidly-exploring random tree (RRT) algorithm [15], [16] turns out to be the most prominent approach in recent years. Its scope also includes the systems with kinodynamic constraints. An interesting approach that joins the RRT-based path planning and redundancy resolution for manipulators is presented in [12]. A large overview of path planning algorithms can be found in [17], [18], [19].

Most of the planning algorithms provide feasible paths than only need to be collision-free [20]. In order to seek for safe paths, a suitable safety assessment needs to be defined. Some efforts to impose additional requirements upon the path like staying away from certain areas as much as possible could be found in [20], [21]. As for robotic manipulators, apart from [2] and [10], we know of no attempt to tackle safety in the planning stage using a dedicated safety criterion. In [2], a danger index based on the distance and velocity between the human and the planar manipulator's end effector is used. An obvious drawback of this approach is that it deals with a planar case and it does not consider the state of the whole manipulator. The approach from [10] is very elaborate, yet it is highly descriptive and still suffers from the local minima problem, although some attention is devoted to its mitigation. In a recent work of the authors [22], a dedicated safety criterion is used within a standard PRM context.

In this work we propose a method for path planning that has a safety information embedded in its heuristics. Hence the planner outputs not only the collision-free paths, but also strives for safer ones. The planner is based on a tree expansion using the bubbles of free configuration space [23] and is proven complete under certain assumptions.

The remainder of the paper is organized as follows. In Section II, the concept of the danger field is introduced, while in Section III we describe the proposed planning algorithm that seeks for safe paths. Section IV contains the analysis of the algorithm's completeness. Simulation examples are given in Section V and concluding remarks in Section VI.

The authors are with the Dipartimento di Elettronica e Informazione, Politecnico di Milano, 20133 Milan, Italy (e-mail: lacevic@elet.polimi.it; rocco@elet.polimi.it)

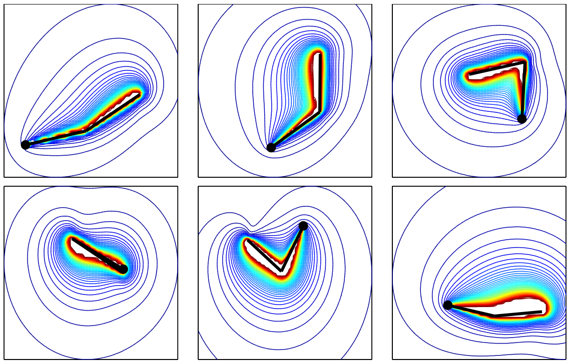

Fig. 1. Snapshots of the CDF contour plot - a 2DOF example: both links accelerate in the counterclockwise direction (the robot base is marked)

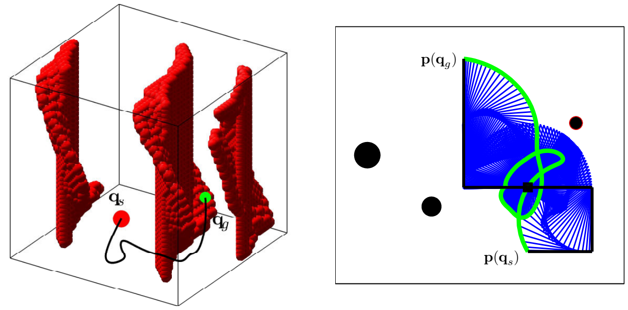

Fig. 2. Path in C-space and the corresponding manipulator motion

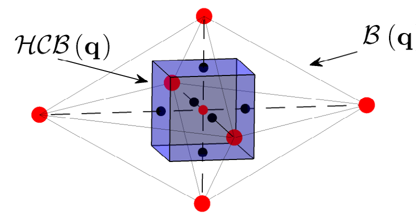

Fig. 3. Diamond-shaped bubble $\mathcal{B}(\mathbf{q})$ and a corresponding hypercube-shaped bubble $\mathcal{HCB}(\mathbf{q})$ (a 3D case)

II. Danger/Safety Evaluation

This section briefly recalls the *cumulative danger field* concept [24]. Due to space constraints, it will not be described in details. For an elaborate description, the reader is referred to [24].

In principle, the cumulative danger field $C\vec{D}F(\mathbf{r},\mathbf{q},\dot{\mathbf{q}})$ is a vector field evaluated at an arbitrary point $\mathbf{r} = (x\ y\ z)^T$ in the robot's workspace. Moreover, it depends on the robot's current configuration vector $\mathbf{q}$ and its velocity captured in vector $\dot{\mathbf{q}}$. Once the environment object has been perceived, the field is computed based on the distance between object and robot, the velocity of the robot, and the angle between the distance and velocity vectors. The danger field is computable in closed form via elementary algebraic expressions [24]. Fig. 1 shows the contour plot of the field induced by the motion of a 2DOF planar manipulator. For making it visually presentable as a function of two variables, the danger field is tacitly restricted to the plane in which the robot moves.

III. Planning Algorithm

The proposed planner is designed to find the collision-free path in C-space from $\mathbf{q}_{start}$ to $\mathbf{q}_{goal}$ that is as short as possible while at the same time tries to minimize the danger induced by the robot's configuration at the obstacles' locations. Fig.2 shows an example of a collision-free path in C-space (left) and a consequent motion of a 3 DOF planar manipulator (right). The C-space is represented as a cube $[-\pi,\pi]^3$ whose axes correspond to joint angles. It includes the C-space obstacles - the set of all configurations that cause the intersection between the robot and the obstacles [12]. The initial and final positions ($\mathbf{p}(\mathbf{q}_s)$ and $\mathbf{p}(\mathbf{q}_g)$ respectively), as well as the path described by the end-effector are indicated.

For redundant manipulators, the goal configuration $\mathbf{q}_{goal}$ can be extended to the *goal region*, i.e. the set of all configurations that enable the robot to complete the task (e.g. grasp the object) [12]. For instance, if the goal is equivalent to the end-effector reaching the point $\mathbf{p}$ in the workspace, the goal region in C-space is the collision-free portion of the self motion manifold corresponding to $\mathbf{p}$ [25]. The algorithm is based on the concept of so-called bubbles of free configuration space, introduced in [23]. The bubble $\mathcal{B}(\mathbf{q})$ at the current configuration $\mathbf{q}$ is a compact region computed using a distance d_c of the robot in configuration $\mathbf{q}$ from the closest obstacle in the workspace. For the robots with n revolute joints, it takes a diamond shape [23]:

$$\mathcal{B}(\mathbf{q}) = \{\mathbf{x}\ :\ \textstyle\sum_{i=1}^{n} r_i|x_i - q_i| < d_c\}\,. \qquad (1)$$

Quantity r_i is the radius of the cylinder whose axis is collocated with the axis of the i-th joint and that encloses all the links starting from i-th joint to the end-effector. Changing the configuration from $\mathbf{q}$ to an arbitrary configuration within a bubble implies that no point on the kinematic chain will move more than d_c and thus no collision will occur. An elaborate assertion of the simplicity in computing the bubbles can be found in [23]. We define a corresponding hypercube bubble $\mathcal{HCB}(\mathbf{q})$ with a center at $\mathbf{q}$ as the largest possible axis-aligned hypercube that belongs to $\mathcal{B}(\mathbf{q})$ (see Fig.3). The algorithm tries to connect $\mathbf{q}_{start}$ and $\mathbf{q}_{goal}$ by simultaneously chaining bubbles both from initial and goal configurations until these chains intersect. Its principle is given by the procedure BUBBLE_PLANNER. The code is based on the bidirectional A^* search algorithm [26], [18]. Lists named "Closed()" stand for the lists of visited nodes (configurations), while the waiting lists of nodes yet to be considered are labeled "Open()". Index 1 stands for the list (tree) that is currently being processed. At each iteration, function INTERSECT checks whether there are $\mathbf{q}_1 \in$ Closed(1) and $\mathbf{q}_2 \in$ Closed(2) such that $\mathcal{B}(\mathbf{q}_1) \cap \mathcal{B}(\mathbf{q}_2) \neq \emptyset$. If that is the case, there is a collision-free path between the trees expanded from $\mathbf{q}_{start}$ and $\mathbf{q}_{goal}$ and hence a path between $\mathbf{q}_{start}$ and $\mathbf{q}_{goal}$. Otherwise, the most promising node $\mathbf{q}_{new}$ is selected from the list Open(1) that minimizes the heuristic function f (described later). The configuration $\mathbf{q}_{new}$ is deleted from the list Open(1) and added to the list of visited nodes Closed(1). Then the bubble $\mathcal{B}(\mathbf{q}_{new})$ is computed and its endpoints are added to the list Open(1). For the original bubble $\mathcal{B}(\mathbf{q}_{new})$

```
procedure BUBBLE_PLANNER(q_start, q_goal)
    Closed(1) ← [q_start]; Closed(2) ← [q_goal];
    Open(1) ← [q_start]; Open(2) ← [q_goal];
    dir ← 1; q_current_goal ← q_goal;
    for k = 1 to k_max do
        if INTERSECT (Closed(1), Closed(2)) then
            return PATH (Closed(1), Closed(2));
        end if
        q_new ← argmin_{x ∈ Open(1)} f(x);
        REMOVE (Open(1), q_new);
        ADD (Closed(1), q_new);
        ADD (Open(1), BUBBLE_ENDPOINTS (q_new));
        SWAP (Closed(1), Closed(2));
        SWAP (Open(1), Open(2));
        dir ← 3 − dir; q_current_goal ← q_new;
    end for
    return Failure
end procedure
```

these endpoints are its vertices and for the hypercube bubble $\mathcal{HCB}(\mathbf{q}_{new})$ the endpoints are the centers of the hypercube's faces (see Fig.3). Theoretically, the overlapping of the bubbles is allowed. The search direction is then reversed by swapping the corresponding lists. The variable dir preserves the information about the search direction. If we expand the tree originating from $\mathbf{q}_{start}$ then $dir = 1$, otherwise $dir = 2$. If the collision-free path is not obtained within a predefined number of iterations k_{max}, the algorithm returns failure.

The heuristic function f is defined as:

$$f(\mathbf{x}) = g(\mathbf{x}) + h(\mathbf{x}) + \alpha C\hat{D}F(\mathbf{x}) - \beta \min_{z \in C} \|\mathbf{x} - \mathbf{z}\|, \quad (2)$$

where $g(\mathbf{x})$ is a cost function of a traversed path from root to $\mathbf{x}$, $h(\mathbf{x}) = \|\mathbf{x} - \mathbf{q}_{current\_goal}\|_1$ is the underestimate of the distance between $\mathbf{x}$ and the current goal $\mathbf{q}_{current\_goal}$. The term $C\hat{D}F(\mathbf{x})$ serves as the estimate of the maximum value of the danger field $\|C\vec{D}F(\mathbf{r}, \mathbf{x}, \dot{\mathbf{x}})\|$ induced by the robot in configuration $\mathbf{x}$ over all of the relevant subjects/obstacles locations $\mathbf{r}$. More precisely, for a single obstacle, its representative location is a point that is closest to the robot. The last term in (2) represents the contribution of the point $\mathbf{x}$ to the spatial diversity of the samples (in the corresponding tree with a list of visited nodes $C = \text{Closed}(dir)$) already processed during the search. The point is considered better if it has a larger minimum distance from the set of already visited nodes. The idea is to increase the tendency of the algorithm to explore less visited parts of the configuration space. Variables α and β are positive tunable parameters.

Unlike the planners that are based on RRT paradigm, the proposed algorithm updates the search trees in a deterministic manner. As for the computational effort needed to extend the tree towards the new configuration, the proposed algorithm does not require the collision checking routine (unlike the RRT algorithm) because the entire edge that is being added to the tree lies within the bubble of free configuration space. On the other hand, the computation of the bubble around a given configuration requires only marginally more effort than determining solely whether a configuration is collision-free [23]. The only additional requirement is the memory space needed for maintaining the Open() lists. Partial resolution to this problem may be the substitution of the A^* search with the *iterative deepening* A^* algorithm [18], [26].

IV. Completeness analysis

The natural question is whether the proposed algorithm guarantees to find a collision-free path from $\mathbf{q}_{start}$ to $\mathbf{q}_{goal}$, if such a path exists. If there exists a feasible path that can be generated via sequence of bubbles as described in the algorithm, the planner is able to find it because it is based on the A^* search that is systematic[1] [18], provided that parameter k_{max} is large enough. On the other hand, the set of such paths might seem too restrictive, since they are piecewise axis-aligned and their segments have specific lengths (constrained by the size of the bubbles). However, in this section we prove that the existence of an arbitrary collision-free path implies the existence of the path attainable by the algorithm. Moreover, the proof considers the completeness of the unidirectional version of the algorithm that is a stronger result. For now, we conduct the proof for the case when the hypercube-shaped bubbles are used. Note that the completeness is preserved if the A^*-based search is replaced with another systematic search technique (e.g. *iterative deepening* or *iterative deepening* A^* [18], [26]).

Definition 4.1: Let $B = \{\mathbf{b}_1,\ \mathbf{b}_2,\ \ldots,\ \mathbf{b}_n\}$ be the basis of vectors that span the configuration space $\mathcal{C}$. We say that a sequence of bubbles $\mathcal{B}(\mathbf{q}_1)$, $\mathcal{B}(\mathbf{q}_2)$, ..., $\mathcal{B}(\mathbf{q}_N)$, where $N \in \mathbb{N}$ is B-feasible if the following holds:

i) $\forall \mathbf{q}_i,\ i = 1, 2, \ldots, N-1\ \ \exists j \in \{1, 2, \ldots, n\}$ such that $\mathbf{q}_{i+1} - \mathbf{q}_i = \delta \mathbf{b}_j,\ \delta \in \mathbb{R}$

ii) $\mathbf{q}_{i+1} \in \partial\mathcal{B}(\mathbf{q}_i),\ i = 1, 2, \ldots, N-1$, where $\partial\mathcal{B}(\mathbf{q}_i)$ denotes the border of the bubble $\mathcal{B}(\mathbf{q}_i)$.

Namely, a B-feasible sequence is the one where each element is obtained from the predecessor through a displacement along one vector of the basis and belongs to the border of the bubble around its predecessor.

Lemma 4.1: Let $\mathbf{q}_A$ and $\mathcal{HCB}(\mathbf{q}_A)$ be a point in a $2D$ configuration space and its corresponding axis-aligned square-shaped bubble of free space. If $\mathbf{q}_B$ is a point on the border of $\mathcal{HCB}(\mathbf{q}_A)$ such that $\mathcal{HCB}(\mathbf{q}_B)$ has nonzero size, there exists an E_2-feasible sequence of axis-aligned square-shaped bubbles starting from $\mathcal{HCB}(\mathbf{q}_A)$ with the final bubble containing $\mathbf{q}_B$, where $E_2 = \left\{[1\ 0]^T,\ [0\ 1]^T\right\}$.

Proof: Without loss of generality, assume $\mathbf{q}_A = \mathbf{0}$ and that $\mathbf{q}_B$ falls into the first quadrant, and belongs to the vertical edge of $\mathcal{HCB}(\mathbf{q}_A)$ (see Fig. 4). We proceed with the propagation of bubbles from $\mathbf{q}_{A_1}$, where $\mathbf{q}_{A_1}$ is the endpoint of the bubble $\mathcal{HCB}(\mathbf{q}_A)$ that belongs to the said edge. If $\mathbf{q}_B \in \mathcal{HCB}(\mathbf{q}_{A_1})$, the process is over, otherwise we choose the point $\mathbf{q}_{A_2} = \partial\mathcal{HCB}(\mathbf{q}_{A_1}) \cap \overline{\mathbf{q}_{A_1}\mathbf{q}_B}$ as the new starting point. Since the segment $\overline{\mathbf{q}_{A_1}\mathbf{q}_B}$ lies entirely within the

[1]If a solution exists, then the search algorithm must report it in finite time; however, if a solution does not exist, it is acceptable for the algorithm to search forever [18].

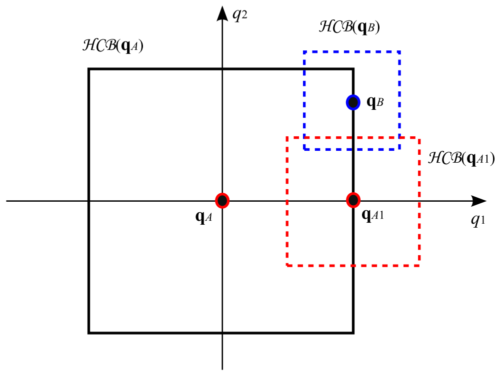

Fig. 4. Reaching the point on the bubble's border via E_2-feasible sequence of bubbles

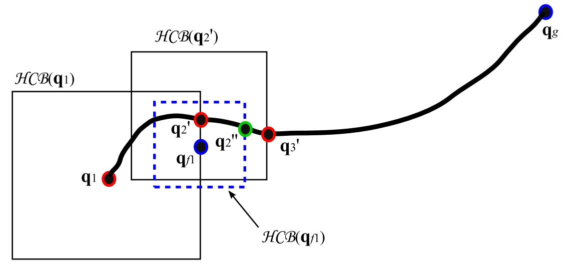

Fig. 5. Reaching the goal via E_2-feasible sequence of bubbles

free configuration space $\mathcal{C}_{free}$ with some minimum positive clearance [2], the linear size of the bubbles propagated along $\overline{\mathbf{q}_{A_1}\mathbf{q}_B}$ will not go below a certain positive value, say ε. Thus, the target point $\mathbf{q}_B$ will eventually be covered after a finite number of steps.

Moreover, we can guarantee that the point $\mathbf{q}_B$ is "deep enough" in the last bubble such that $d\left\{\mathbf{q}_B,\ \partial\mathcal{HCB}(\mathbf{q}_f)\right\} \geq \frac{\varepsilon}{4}$, where $\mathcal{HCB}(\mathbf{q}_f)$ is the last bubble in the sequence. The function $d\{\cdot,\cdot\}$ stands for the minimum distance between two input arguments. Assume the opposite, i.e. that $\mathcal{HCB}(\mathbf{q}_f)$ covers $\mathbf{q}_B$ but $d\left\{\mathbf{q}_B,\ \mathcal{HCB}(\mathbf{q}_f)\right\} < \frac{\varepsilon}{4}$. In that case, we can construct one more bubble at the point $\mathbf{q}_C = \mathcal{HCB}(\mathbf{q}_f) \cap \mathcal{HCB}(\mathbf{q}_A)$, where $d\left\{\mathbf{q}_B,\ \mathbf{q}_C\right\} < \frac{\varepsilon}{4}$. Since the linear size of $\mathcal{HCB}(\mathbf{q}_C)$ is greater or equal than ε, it will certainly enclose $\mathbf{q}_B$ such that $d\left\{\mathbf{q}_B,\ \mathcal{HCB}(\mathbf{q}_C)\right\} \geq \frac{\varepsilon}{4}$. Clearly we can declare $\mathcal{HCB}(\mathbf{q}_C)$ a final bubble. ■

Theorem 4.1: Let $\mathbf{q}_s$ and $\mathbf{q}_g$ represent a start and a goal in $2D$ configuration space respectively. Assume there exists a collision-free path given with $\mathbf{c}:[0,1] \rightarrow \mathcal{C}_{free}$, such that $\mathbf{c}(0) = \mathbf{q}_s$ and $\mathbf{c}(1) = \mathbf{q}_g$ and that $\forall s \in [0,1]$ the following holds:

$$d\left\{\mathcal{R}\left(\mathbf{c}(s)\right),\ \mathcal{WO}\right\} \geq d_{min} > 0, \tag{3}$$

where $\mathcal{R}(\mathbf{q})$ represents the operational space occupied by the robot at the configuration $\mathbf{q}$ and $\mathcal{WO}$ is the set of workspace regions occupied by the obstacles.

Then, there exists an E_2-feasible sequence of axis-aligned square-shaped bubbles $\mathcal{HCB}(\mathbf{q}_s) = \mathcal{HCB}(\mathbf{q}_1)$, $\mathcal{HCB}(\mathbf{q}_2)$, ..., $\mathcal{HCB}(\mathbf{q}_N)$, where $\mathbf{q}_g \in \mathcal{HCB}(\mathbf{q}_N)$

Proof: Quinlan [23] has shown that the collision-free path that has a nonzero minimum clearance can be covered with a finite set of bubbles. We emulate his approach using $\mathcal{HCB}$s. Let $\mathcal{HCB}(\mathbf{q}_1) = \mathcal{HCB}(\mathbf{q}_s)$ intersect $\mathbf{c}(s)$ at $\mathbf{q}_2'$. We further construct $\mathcal{HCB}(\mathbf{q}_2')$ that intersects $\mathbf{c}(s)$ at $\mathbf{q}_3'$ and we repeat the procedure until $\mathbf{q}_g$ is covered (see Fig. 5). Condition (3) implies that the size of the bubbles will not go below a certain positive number. Thus, the path $\mathbf{c}(s)$ can be covered in a finite number of steps. Of course, the sequence of bubbles constructed this way is not E_2-feasible in general. We now show how this can be achieved actually. According to Lemma 4.1, point $\mathbf{q}_2'$ can be reached from $\mathbf{q}_1$ using E_2-feasible sequence of bubbles. Let the final bubble of this sequence be $\mathcal{HCB}(\mathbf{q}_{f_1})$. Since $\mathbf{q}_2'$ falls into $\mathcal{HCB}(\mathbf{q}_{f_1})$ "deep enough", i.e. $d\left\{\mathbf{q}_2',\ \mathcal{HCB}(\mathbf{q}_{f_1})\right\} \geq \frac{\varepsilon_1}{4}$, where ε_1 is a positive constant, it means that $\mathcal{HCB}(\mathbf{q}_{f_1})$ intersects the path $\mathbf{c}(s)$ at a certain point $\mathbf{q}_2''$ outside the $\mathcal{HCB}(\mathbf{q}_1)$ such that $d\left\{\mathbf{q}_2'',\ \mathbf{q}_2'\right\} \geq \frac{\varepsilon_1}{4}$ (see Fig. 5). Hence, we have "moved" along a path $\mathbf{c}(s)$ using a finite E_2-feasible sequence of bubbles. Moreover, according to Lemma 4.1, we can construct an E_2-feasible sequence of bubbles from $\mathbf{q}_{f_1}$ to $\mathbf{q}_2''$. The final bubble of this sequence will enclose $\mathbf{q}_2''$ and intersect $\mathbf{c}(s)$ at a further point. Proceeding in this way it is easy to conclude that the goal configuration $\mathbf{q}_g$ can be reached if we repeat this procedure. Note that this is achievable in a finite number of steps, since each advancement along a path is greater or equal to a certain positive constant. ■

In the sequel of the section we exploit the conclusions above to tackle the problem in higher dimensions.

Lemma 4.2: If $\mathbf{q}_A$ is a point in a $3D$ configuration space and $\mathcal{HCB}(\mathbf{q}_A)$ is a corresponding axis-aligned cube-shaped bubble of free space, then for each point $\mathbf{q}_B \in \partial\mathcal{HCB}(\mathbf{q}_A)$ such that the bubble $\mathcal{HCB}(\mathbf{q}_B)$ has a nonzero size, there exists an E_3-feasible sequence of axis-aligned cube-shaped bubbles starting from $\mathcal{HCB}(\mathbf{q}_A)$ with the final bubble $\mathcal{HCB}(\mathbf{q}_f)$ covering $\mathbf{q}_B$.

Proof: Let $\mathbf{q}_{A_1}$ be the centroid of the face of the cube $\mathcal{HCB}(\mathbf{q}_A)$ that contains $\mathbf{q}_B$. We continue with the bubbles propagation from $\mathbf{q}_{A_1}$ (see Fig.6 - left). The remaining part of the path to cover has a maximum of two perpendicular segments $\overline{\mathbf{q}_{A_1}\mathbf{q}_{A_2}}$ and $\overline{\mathbf{q}_{A_2}\mathbf{q}_B}$ that clearly lie in the plane i.e. a face of the bubble $\mathcal{HCB}(\mathbf{q}_A)$). Restricting the propagation in a way that the center of the bubble has to belong to the same face, the problem reduces to the covering of a path in the plane. According to Theorem 4.1, the path in the plane can be covered in a finite number of steps. ■

Corollary 4.1: If there is a collision-free path $\mathbf{c}(s)$ in $3D$ configuration space from $\mathbf{q}_s$ to $\mathbf{q}_g$, then there is an E_3-feasible sequence of axis-aligned cube-shaped bubbles starting from $\mathcal{HCB}(\mathbf{q}_s)$ and ending with $\mathcal{HCB}(\mathbf{q}_f)$ such that $\mathbf{q}_g \in \mathcal{HCB}(\mathbf{q}_f)$

Proof: Using Lemma 4.2, we may proceed in the same way as it is done in the proof of Theorem 4.1. ■

Theorem 4.2: The statement of the Theorem 4.1 holds for

[2] If $\mathbf{q}_B$ coincides with the corner of $\mathcal{HCB}(\mathbf{q}_A)$, there is a theoretical possibility that it lies at on the border of $\mathcal{C}_{free}$. However, a nonzero size of $\mathcal{HCB}(\mathbf{q}_B)$ ensures that it is not the case.

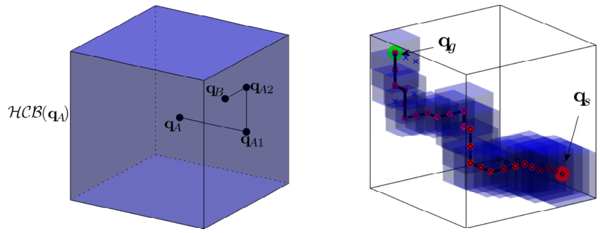

Fig. 6. Reaching the point on the bubble's border via E_3-feasible sequence of bubbles (left). The chain of bubbles that connect $\mathbf{q}_s$ and $\mathbf{q}_g$ and a path passed through it (right).

any dimension of configuration space.

Proof: The problem of finding an E_4-feasible sequence of axis-aligned hypercube-shaped bubbles starting from $\mathcal{HCB}(\mathbf{q}_s)$ and ending with $\mathcal{HCB}(\mathbf{q}_f)$ such that $\mathbf{q}_g \in \mathcal{HCB}(\mathbf{q}_f)$, where $\mathbf{q}_s$ and $\mathbf{q}_g$ are start and goal configurations in $4D$ configuration space can be reduced to obtaining an E_4-feasible sequence from the center of the $4D$ hypercube to any point on its boundary. The latter can be reduced to finding an E_3-feasible sequence that covers a path in the $3D$ subspace of the $4D$ hypercube, that is proven achievable.

Proceeding inductively, we can obtain an E_n-feasible sequence of bubbles for a problem in any dimension. ■

The performed analysis proves that if a collision-free path exists, then an E_n-feasible sequence of bubbles can be built upon it. The proposed algorithm outputs such a sequence and a path is easily generated by connecting the successive bubble centers. Note that the result is not only a path, but a "tube" constructed by the chain of bubbles (see Fig.6 - right). This region of free C-space allows for a further path modifications, e.g. smoothing or the application of elastic bands/strips framework [11], [23].

V. Examples

In this section we present some validation of the proposed algorithm. The tests are performed on the models of a 3-DOF planar arm and a 6-DOF arm that move in a more or less cluttered environment. The robots are modeled using the Robotics Toolbox for Matlab [27]. Since the output of the algorithm is a path, still not a trajectory parameterized in time, the velocity information is not available. For now, the estimate of the danger field takes into account only the static part of the danger field, i.e. the velocity contribution is neglected. Fig.7 shows two different arrangements of the obstacles with two corresponding start and goal configurations. Figures on the left show the resulting path when the tree is expanded via hypercube-shaped bubbles while the result of the search via original bubbles is depicted in the figures on the right. Not surprisingly, less iterations (i.e. nodes in both trees) are required for the algorithm to obtain the path when the diamond-shaped bubbles are used since they provide larger search steps. Although we still do not have a proof of the algorithm completeness when the diamond-shaped bubbles are used, it is not yet observed that this approach fails to find the path if the algorithm based on the hypercube-shaped bubbles succeeds. For the first scenario

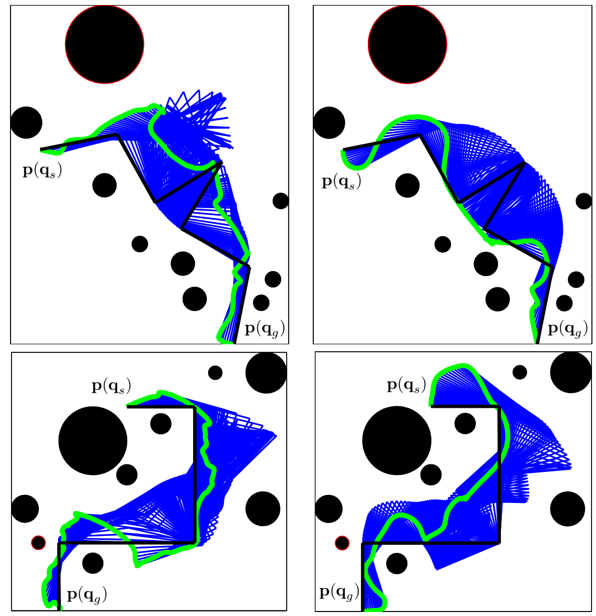

Fig. 7. A 3-DOF example: the path generated via hypercube-shaped bubbles (left) and via diamond-shaped bubbles (right). The path described by the end-effector is indicated.

the path is obtained in 457 iterations for the hypercube-shaped bubbles and in 49 iterations for the diamond-shaped bubbles. The second example required 1080 iterations for the hypercube-shaped bubbles and 247 iterations for the diamond-shaped bubbles. Fig.8 shows the results for the 6-DOF manipulator. For the first scenario (7 obstacles), the algorithm based on the hypercube-shaped bubbles obtains the path in 146 iterations whereas the diamond-shaped based algorithm finishes after 26 iterations. For the second scenario (45 obstacles), the corresponding numbers of iterations are 227 and 41. Note that in all the scenarios, the robot's distance from the obstacles is considerably large along the path, indicating its high quality in terms of safety. As for the implementation details, some improvements still need to be done. Namely, the running times for the examples above are couple of seconds, which is clearly behind the state of the art implementations. This is mostly due to the fact that all of the routines are written in MATLAB and that no state of the art libraries for proximity queries are utilized. Nevertheless, the number of iterations and distance computations provide an unambiguous information about the algorithm's performance.

We have observed that the safety assessment part in the heuristic function usually speeds up the algorithm. This may be attributed to the fact that the bubbles (and thus the search steps) are likely to be larger in safer areas. As a matter of fact, the slowness of the algorithm without the safety heuristic is a culprit for establishing how much is the quality of the resulting path increased by the inclusion of safety assessment. To have a clue how a safety heuristic affects the quality of a path when e.g., the PRM approach is used, the reader is referred to [22].

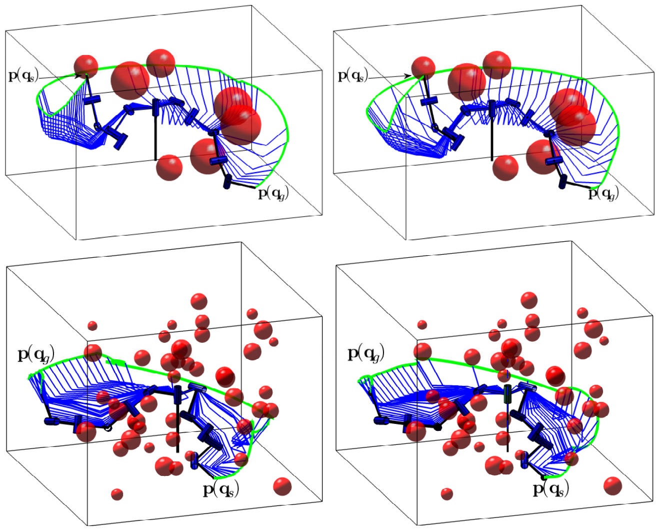

Fig. 8. A 6-DOF example: the path generated via hypercube-shaped bubbles (left) and via diamond-shaped bubbles (right)

VI. Conclusions and Future Works

A novel approach to path planning for robotic manipulators is presented. The planner searches for a collision-free path via bubbles of free C-space while trading off the path length with a defined degree of danger. The algorithm is based on bidirectional A^*-search technique with a heuristic function that makes account of the danger. We estimated the danger degree using the danger field - a proposed quantity that captures the complete kinematic behavior of the manipulator and is easily computable. We have proved the completeness of the proposed algorithm if a more conservative variant of bubbles is used. The proposed algorithm is supported with several numerical examples.

The future work will investigate the possibilities to replace the A^*-search algorithm with another search technique like iterative deepening A^* in order to decrease the memory space requirements. From a control point of view, we will utilize the concept of danger field to implement the low level reactive control for dealing with dynamic environments.

VII. Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 - Challenge 2 - Cognitive Systems, Interaction, Robotics - under grant agreement No 230902 - ROSETTA.

References

[1] K. Ikuta, M.Nokata, H.Ishii, Safety Evaluation Method of Design and Control for Human-Care Robots, *The International Journal of Robotics Research*, 2003, Vol. 22, No. 5, pp 281-297

[2] M. Nokata, K.Ikuta, H. Ishii, Safety-optimizing Method of Human-care Robot Design and Control, *Proc. of IEEE Conf. on Rob. and Autom.*, 2002.

[3] J. Heinzmann and A. Zelinsky, Quantitative safety guarantees for physical human-robot interaction, *The International Journal of Robotics Research*, 2003, vol. 22, no. 7/8, pp. 479-504

[4] A. Bicchi, G. Tonietti, Fast and Soft Arm Tactics - Dealing with the Safety-Performance Trade-Off in Robot Arms Design and Control, *IEEE Robotics and Automation Magazine, Special issue on Dependability in Human-Friendly Robots*, June 2004, Vol. 11, No. 2

[5] D. Henrich, S. Kuhn, Modeling Intuitive behavior for safe human/robot coexistence cooperation, *Proc. of IEEE Conf. on Rob. and Autom.*, vol., no., pp.3929-3934, 15-19 May 2006

[6] J. Guiochet, D. Powell, E. Baudin, J.-P. Blanquart, Online safety monitoring using safety modes, *6th IARP/IEEE-RAS/EURON Workshop on Technical Challenges for Dependable Robots in Human Environments*, Pasadena, CA, USA (May 2008)

[7] R. Alami, A. Albu-Schaeffer, A. Bicchi, R. Bischoff, R. Chatila, A. De Luca, A. De Santis, G. Giralt, J. Guiochet, G. Hirzinger, F. Ingrand, V. Lippiello, R. Mattone, D. Powell, S. Sen, B. Siciliano, G. Tonietti, and L. Villani, Safe and Dependable Physical Human-Robot Interaction in Anthropic Domains: State of the Art and Challenges, *Proc. IROS'06 Workshop on pHRI - Physical Human-Robot Interaction in Anthropic Domains*, 2006. IEEE

[8] A. Pervez, J. Ryu, Safe physical human robot interaction-past, present and future *Journal of Mechanical Science and Technology*, Vol. 22, No. 3. (5 March 2008), pp. 469-483.

[9] D. Kulic, E. Croft, Safe Planning for Human-Robot Interaction, *Journal of Robotic Systems* 2005, Vol. 22, No. 7, pp 383 - 396

[10] D. Kulic, E. Croft, Real-time safety for human-robot interaction, *Robotics and Autonomous Systems*, Jan 31, 2006, v54, n1, pp 1-12

[11] O. Brock, O. Khatib, Elastic Strips: A Framework for Motion Generation in Human Environments *International Journal of Robotics Research*, Vol. 21, No. 12, Dec 2002, pp. 1031-1052

[12] D. Bertram, J. Kuffner, R. Dillmann, and T. Asfour, An integratedapproach to inverse kinematics and path planning for redundant manipulators, *Proc. of IEEE Conf. on Rob. and Autom.*, 2006.

[13] J. F. Canny, *The Complexity of Robot Motion Planning*, Cambridge, MA: MIT Press, 1988.

[14] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H.Overmars, Probabilistic roadmaps for path planning in high-dimensional configuration spaces, *IEEE Transactions on Robotics and Automation*, 1996, 12(4), pp 566580.

[15] S. M. Lavalle , J. J. Kuffner, Rapidly-Exploring Random Trees: Progress and Prospects, *Proceedings Workshop on the Algorithmic Foundations of Robotics*, 2000.

[16] J. J. Kuffner and S. M. LaValle, RRT-connect: An efficient approach to single-query path planning, *Proc. of IEEE Conf. on Rob. and Autom.*, 2000, pp 995-1001.

[17] J. C. Latombe, *Robot Motion Planning*, Boston, MA: Kluwer Academic Publishers, 1991.

[18] S. M. LaValle, *Planning Algorithms*, Cambridge University Press 2006,

[19] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*, MIT Press, Boston, 2005.

[20] D. Sent and M.H. Overmars, Motion planning in an environment with dangerzones, *Proc. of IEEE Conf. on Rob. and Autom.*, 2001, pp. 1488-1493.

[21] P. Melchior, B. Orsoni, O. Lavialle, A. Poty, and A. Oustaloup, Consideration of obstacle danger level in path planning using A* and fast-marching optimisation: comparative study, *Signal Process.*, 2003, 83, 11, pp 2387-2396.

[22] B. Lacevic and P. Rocco, Sampling-based Safe Path Planning for Robotic Manipulators, *ETFA 2010 - 15th IEEE International Conference on Emerging Techonologies and Factory Automation*.

[23] S. Quinlan, *Real-time modification of collision-free paths*, PhD dissertation, Stanford University, 1994.

[24] B. Lacevic and P. Rocco, Kinetostatic danger field - a novel safety assessment for human-robot interaction, *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS2010)*.

[25] J. W. Burdick, On the Inverse Kinematics of Redundant Manipulators: Characterization of the Self-Motion Manifolds, *Proc. of IEEE Conf. on Rob. and Autom.*, 1989, pp 264-270.

[26] H. Kaindl, G. Kainz, Bidirectional Heuristic Search Reconsidered, *J. Artif. Intell. Res. (JAIR)*, 1997, 7, pp 283-317.

[27] P.I. Corke, A Robotics Toolbox for Matlab, *IEEE Robotics and Automation Magazine*, 1996, Vol. 3, No 1, pp 24-32.