Contents lists available at ScienceDirect

European Journal of Control

European
Journal
of Control

Hamiltonian path planning in constrained workspace

CrossMark

Daniele Casagrande [a,*], Gianfranco Fenu [b], Felice Andrea Pellegrino [b]

[a] Dipartimento Politecnico di Ingegneria e Architettura, Università degli Studi di Udine, Via delle Scienze, 206-33100 Udine, Italy
[b] Dipartimento di Ingegneria e Architettura, Via A. Valerio 10 34127 Trieste, Italy

ARTICLE INFO

ABSTRACT

A methodology to plan the trajectories of robots that move in an n-dimensional Euclidean space, have to reach a target avoiding obstacles and are constrained to move in a region of the space is described. It is shown that if the positions of the obstacles are known then a Hamiltonian function can be constructed and used to define a collision-free trajectory. It is also shown that the method can be extended to the case in which the target or the obstacles (or both) move. Results of simulations for a pair of planar robots and a three degrees-of-freedom manipulator are finally reported.

© 2016 European Control Association. Published by Elsevier Ltd. All rights reserved.

1. Introduction

Robot path planning is an important topic that has been extensively studied in the past three decades. Different approaches have been proposed for solving the problem of reaching a (possibly moving) target while avoiding (possibly moving) obstacles (see e.g. [24] and references therein). Among several ideas that have been developed, we recall the workspace density [9], the harmonic potential functions [20], the navigation functions [22], the decomposition approaches [6,2,1], the distance function [14], the vector field histogram [3,34], the dynamic window method [11], the rule-based approach [12], the use of the so-called Particle Swarm Optimization [32]. A well-known family of techniques is that based on artificial potential fields [19], in which the point representing the configuration of the robot behaves like a charged particle moving in a force field. In particular, the point moves along the antigradient of a potential field depending on both the obstacles and the target position and generated in such a way that a repulsive potential function is associated with each obstacle, while an attractive one is generated by the goal. As a result, a (possibly local) minimum of the potential function is placed in correspondence of the goal. By using harmonic functions [7], or restricting the admissible obstacle shapes [29], uniqueness of the minimum can be guaranteed. Methods for facing other drawbacks, such as chattering in narrow passages and mislocation of the

minimum, or to broaden the field of applicability of the method, are also available (see for example [13,28,16,35,27]).

Other approaches that have been widely used rely on sampling-based algorithms, in particular the probabilistic road maps (PRM) [18] and the rapidly exploring random tree (RRT) algorithm (introduced by [23]), which can efficiently solve complex high-dimensional problems. The main idea of such algorithms is to perform a search in the configuration space, applying a proper sampling scheme, instead to explicitly determine the sets of robot configurations causing the collision of the robot with obstacles or self-collision (see e.g. [24] for details). Variants of these methods have been proposed. In particular, in [17] the optimality of PRM and RRT algorithms is studied and modified versions, called PRM* and RRT*, that are also proven to be asymptotically optimal are described. In [21] the authors apply the RRT algorithm to the minimization of a function defined on a vector field constructed ad hoc on the basis of the environment and of the path planning problem. In [33] the extra degrees of freedom of redundant manipulators are used to satisfy geometric and kinematic constraints. In [25] a path planning strategy, based on the RRT paradigm and a static field, applied in the robot's workspace, to guarantee that the generated robot path keeps away from certain areas as much as possible is illustrated. Attention has been devoted also to improve efficiency and scalability of RRT variants, as proposed in [30,10,15], among others.

In recent works [5,4], a novel approach to solve the obstacle avoidance problem for planar robots, based on the construction of a Hamiltonian function, has been described and analysed in its simplest version. Similar to the potential field approach, the Hamiltonian is a function of the environment, namely of the configuration of the obstacles, but it is constructed by taking into

* Corresponding author.
E-mail addresses: daniele.casagrande@uniud.it (D. Casagrande), fenu@units.it (G. Fenu), fapellegrino@units.it (F.A. Pellegrino).

account only the shape and the position of the obstacles. Differently from the potential field method, instead, the information on the target is not needed to construct the Hamiltonian function. A similar approach is considered in [8], where the Hamiltonian function is the sum of an energy term depending on the momentum and an artificial potential field term depending on the configuration. The method is exploited to generate periodic orbits which are shown to remain far from the obstacles. However, the Hamiltonian function is not used to generate a trajectory directed to a target (avoiding the obstacles) which, on the contrary, is the goal of the results described herein.

This property has the advantage that the Hamiltonian function does not change its structure if the target moves. Once the Hamiltonian function is constructed, the robot is driven towards the target with a mixed strategy that exploits the level lines of the Hamiltonian function only in the vicinity of the obstacle and only to bypass it.

Herein we improve the mentioned technique to the following respects:

- we show how to impose kinematic constraints such as limiting the value of the speed and acceleration;
- we consider the case of moving target and moving obstacles;
- we show how to compress the information needed for the implementation and how it can be processed in compressed form.

2. Obstacle avoidance in the planar case

In this section we consider a Hamiltonian robot, namely a point mass, and we describe how the theory of Hamiltonian systems can be exploited to determine a trajectory avoiding obstacles. For further details we refer the reader to [5]; herein we present the idea behind the method and the basic definitions that are needed for the remainder of the paper.

We begin by recalling the structure of a planar Hamiltonian system. Its dynamics can be described by the equations

$$\dot{x} = \frac{\partial H(x,y)}{\partial y}, \quad \dot{y} = -\frac{\partial H(x,y)}{\partial x}, \tag{1}$$

where $H : \mathbb{R}^2 \to \mathbb{R}$ is a continuously differentiable function that is called the *Hamiltonian function* of the system. System (1) has two important properties.

- The value of H along the trajectories of (1) is constant.
- For almost all (x,y), the unit velocity vector of the trajectory and the unit gradient vector of H form a right-handed Cartesian reference system. In other words, a virtual observer moving along the trajectory of system (1) would have increasing values of the function H on the left and decreasing values on the right.

Definition 2.1. Consider a continuous function $H : \mathbb{R}^2 \to \mathbb{R}$. The (super) level set associated with the value H_R is the set $\mathcal{L}_{H_R} \triangleq \{\boldsymbol{\xi} \in \mathbb{R}^2 : H(\boldsymbol{\xi}) \geq H_R\}$. A connected set $\mathcal{R} \subset \mathcal{L}_{H_R}$ is a *maximal connected component* (MCC) of the level set $\mathcal{L}_{H_R}$ if for each connected set $\mathcal{R}_0 \neq \mathcal{R}$, such that $\mathcal{R} \subset \mathcal{R}_0$, there exists $\boldsymbol{\xi}_0 \in \mathcal{R}_0 \backslash \mathcal{R}$ such that $\boldsymbol{\xi}_0 \notin \mathcal{L}_{H_R}$. The boundary of each MCC of a level set is a level line. □

Suppose, now, that the continuous function $H : \mathbb{R}^2 \to \mathbb{R}$ has a local maximum at a point $\boldsymbol{\xi}_M$. Due to the continuity of H, we can always choose a level set $\mathcal{L}_{H_R}$ such that $\boldsymbol{\xi}_M \in \mathcal{L}_{H_R}$ and the MCC of $\mathcal{L}_{H_R}$ containing $\boldsymbol{\xi}_M$ is compact. Let $\mathcal{R}_M$ denote this MCC. The boundary of $\mathcal{R}_M$ is a closed level line and, due to the property explained above, a dynamical system with initial condition on the boundary of $\mathcal{R}_M$ and evolving according to (1) runs along the level
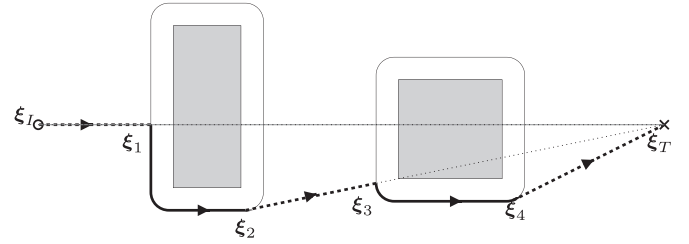

Fig. 1. An example of collision-free path. The dashed lines correspond to dynamics described by $\dot{x} = a$ and $\dot{y} = b$, while the solid line corresponds to the trajectories of system (1). The closed lines surrounding the obstacles are the level lines associated with H_R.

line in a counter-clockwise direction. This fact may be used to derive the solution of the obstacle avoidance problem.

2.1. Obstacle avoidance

Suppose that a point mass has to reach a target in the position $\boldsymbol{\xi}_T \in \mathbb{R}^2$ starting from the initial position $\boldsymbol{\xi}_I \in \mathbb{R}^2$ avoiding some obstacles $O_1, \ldots, O_m$ at least one of which lies on the straight line connecting $\boldsymbol{\xi}_T$ and $\boldsymbol{\xi}_I$. Moreover, suppose that there exist a function[1] $H : \mathbb{R}^2 \to \mathbb{R}$ and a value H_R such that $H(\boldsymbol{\xi}_I) < H_R$, $H(\boldsymbol{\xi}_T) < H_R$, each MCC of $\mathcal{L}_{H_R}$ is simply connected and,[2] for all $i = 1, \ldots, m$,

$$O_i \subset \mathrm{Int}(\mathcal{L}_{H_R}). \tag{2}$$

This scenario is depicted in Fig. 1 where two obstacles (the grey boxes) lie between the initial position and the target and $\mathcal{L}_{H_R}$ is the union of the two "rectangles" with rounded vertices. In this scenario, any path belonging to $\mathbb{R}^2 \backslash \mathrm{Int}(\mathcal{L}_{H_R})$, namely any curve $\gamma : [0,1] \to \mathbb{R}^2$ such that $\gamma(0) = \boldsymbol{\xi}_I$, $\gamma(1) = \boldsymbol{\xi}_T$ and for all $s \in [0,1]$,

$$\gamma(s) \in \mathbb{R}^2 \backslash \mathrm{Int}(\mathcal{L}_{H_R}), \tag{3}$$

is a collision-free path connecting the initial and the final position. In particular, one of these collision-free paths may be obtained by exploiting the properties of the Hamiltonian system described above as follows. Let $\boldsymbol{\xi} = (x,y)^\top$ be the state of the Hamiltonian planar system, denote with ∇H the gradient of H, i.e. the vector

$$\nabla H = \left(\frac{\partial H}{\partial x}, \frac{\partial H}{\partial y} \right)^\top, \tag{4}$$

and let $\rho(\boldsymbol{\xi}) : \mathbb{R}^2 \to \mathbb{R}$ be defined by

$$\rho(\boldsymbol{\xi}) = \nabla H^\top (\boldsymbol{\xi}_T - \boldsymbol{\xi}). \tag{5}$$

Finally, let the dynamics of the planar system be described by the following two-modes hybrid strategy.

(D1) If $H(\boldsymbol{\xi}(t)) < H_R$ or $\rho(\boldsymbol{\xi}(t)) < 0$, then $\dot{x} = a$ and $\dot{y} = b$, with a and b such that the resulting trajectory is a straight line connecting $\boldsymbol{\xi}(t)$ to the target.

(D2) If $H(\boldsymbol{\xi}(t)) = H_R$, then $\dot{x}$ and $\dot{y}$ are as in (1).

Remark 2.1. In the case of noisy data, the strategy above may end up chattering between D1 and D2. A simple solution to avoid this chattering phenomenon is to introduce a hysteresis in the switching between the two control laws.

An example of a collision-free path generated by the above strategy is illustrated in Fig. 1.

From $\boldsymbol{\xi}_I$ to $\boldsymbol{\xi}_1$ the robot approaches the obstacle, hence the value of the Hamiltonian function increases along the trajectory and, as a consequence, $\rho(\boldsymbol{\xi}) > 0$. However, the value of H is less

---

[1] In Sections 4 and 5 two methods to construct this function are described.
[2] We denote by $\mathrm{Int}(X)$ the interior of the set X.

than H_R and the dynamics D1 are selected, for instance with $a=1$ and $b=0$ ($\boldsymbol{\xi}_I$ and $\boldsymbol{\xi}_T$ lie on a horizontal line). When the robot reaches the point $\boldsymbol{\xi}_1$ for which $H(\boldsymbol{\xi}_1)=H_R$, the dynamics switch to the Hamiltonian dynamics D2 and the robot moves along the level line corresponding to H_R. In $\boldsymbol{\xi}_2 \triangleq (x_2, y_2)^\top$ the gradient of H and the vector $\boldsymbol{\xi}_T - \boldsymbol{\xi}_2$ are orthogonal, hence $\rho(\boldsymbol{\xi}_2)=0$. If τ denotes the time instant such that $\boldsymbol{\xi}(\tau)=\boldsymbol{\xi}_2$, then $\rho(\boldsymbol{\xi}(\tau^+))<0$ and the dynamics in τ^+ switch back to D1, for instance with $a=1$ and $b=(y_T-y_2)/(x_T-x_2)$. This procedure is repeated until the target is reached.

The two-steps dynamics can be re-stated formally as follows. Let $\varepsilon(t)$ be a piecewise-constant function from $\mathbb{R}^+$ to $\{0,1\}$ such that

$$\varepsilon(t) = \begin{cases} 1 & \text{if } H(\boldsymbol{\xi}(t)) < H_R \text{ or } \rho(\boldsymbol{\xi}(t))<0, \\ 0 & \text{if } H(\boldsymbol{\xi}(t)) = H_R. \end{cases} \quad (6)$$

If $\boldsymbol{\xi}_I \notin \mathrm{Int}(\mathcal{L}_{H_R})$, then the trajectory of the system that evolves according to the dynamics

$$\dot{\boldsymbol{\xi}} = \alpha(\boldsymbol{\xi}_T - \boldsymbol{\xi})\varepsilon(t) + \beta \begin{pmatrix} 0 & (1-\varepsilon(t)) \\ -(1-\varepsilon(t)) & 0 \end{pmatrix} \nabla H \quad (7)$$

with initial conditions $\boldsymbol{\xi}(0) = \boldsymbol{\xi}_I \notin \mathrm{Int}(\mathcal{L}_{H_R})$ gives a collision-free path for all $\alpha > 0$ and all $\beta > 0$.

Remark 2.2. The parameters α and β can be used to change the velocity of the point mass.

3. Continuous and bounded velocities

In a practical application of the method it is desirable to consider the right-hand side terms in Eq. (7) as the true commands according to which the point mass is steered. However, Eq. (7), that represents the time-derivative of the motion along each generalized coordinate, makes use of the piecewise-constant signal $\varepsilon(t)$, which renders the overall dynamics discontinuous. In a real system such a discontinuity is not allowed, since velocities must be continuous.

Moreover, due to physical constraints, velocities must also be bounded. In this section we consider both these requirements, we describe a modified switching law or, more properly, a "transition law" between the linear and the Hamiltonian dynamics to render the velocities continuous and we show that the velocities are bounded.

We first modify the dynamics (7) in such a way that the right hand side term is continuous. This task can be performed by substituting the switching signal (6) between the dynamics D1 and D2 with a continuous signal. The simplest continuous transition from the value $\varepsilon(t) = 1$, corresponding to the dynamics D1, to the value $\varepsilon(t) = 0$, corresponding to the dynamics D2, is a linear function. To uniquely determine it, we use the following procedure. Instead of a single reference value H_R, we select two reference values H_R and H_S, with $H_S < H_R$, and we set

$$\varepsilon(t) = \begin{cases} 1 & \text{if } H(\boldsymbol{\xi}(t)) < H_S, \\ \dfrac{H_R - H(\boldsymbol{\xi}(t))}{H_R - H_S} & \text{if } H(\boldsymbol{\xi}(t)) \in [H_S, H_R) \text{ or } \rho(\boldsymbol{\xi}(t))<0, \\ 0 & \text{if } H(\boldsymbol{\xi}(t)) = H_R. \end{cases} \quad (8)$$

In Fig. 2 a trajectory resulting from the use of (8) is shown. Between $\boldsymbol{\xi}_I$ and $\boldsymbol{\xi}_1$, the value of the Hamiltonian function is less than H_S and the transition signal ε is equal to 1 (first equation in (8)). Between $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ the value of the Hamiltonian function is between H_S and H_R and ε decreases linearly (second equation in (8)). Between $\boldsymbol{\xi}_2$ and $\boldsymbol{\xi}_3$ the trajectory lies on the level line corresponding to H_R and $\varepsilon = 0$ (third equation in (8)). In $\boldsymbol{\xi}_3$ the gradient
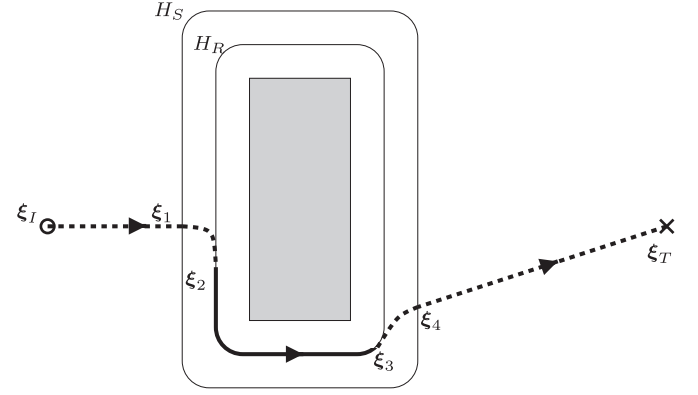

Fig. 2. A continuously differentiable trajectory obtained with the continuous dynamics (8).

of the Hamiltonian function and the straight line connecting $\boldsymbol{\xi}_3$ with $\boldsymbol{\xi}_T$ are orthogonal and $\rho(\boldsymbol{\xi}_3) = 0$; if t_3 denotes the instant such that $\boldsymbol{\xi}(t_3) = \boldsymbol{\xi}_3$, then $\rho(\boldsymbol{\xi}(t_3^+)) < 0$ and ε begins to increase linearly (see again the second equation in (8)). Finally, between $\boldsymbol{\xi}_4$ and $\boldsymbol{\xi}_T$ the value of the Hamiltonian function is again less than H_S and $\varepsilon = 1$.

We now show that, under the natural assumption that the environment in which the robot moves is bounded, the velocities expressed by (7)–(8) are bounded.

Proposition 3.1. If $R \in \mathbb{R}^+$ is such that[3] $\mathcal{B}_0(R)$ contains $\boldsymbol{\xi}_I$, $\boldsymbol{\xi}_T$ and all the level lines corresponding to H_S, then the velocities associated with the dynamics described by (7) and (8) are bounded.

Proof. By the standard assumption that H is continuously differentiable (see Eq. (1)), the gradient of H is bounded in $\mathcal{B}_0(R)$. Hence, there exists $G > 0$ such that, for all $\boldsymbol{\xi} \in \mathcal{B}_0(R)$,

$$\left| \frac{\partial H}{\partial x}(\boldsymbol{\xi}) \right| \le G, \quad \left| \frac{\partial H}{\partial y}(\boldsymbol{\xi}) \right| \le G. \quad (9)$$

The following chain of inequalities proves the claim for $\dot{x}$.

$$|\dot{x}(t)| = \left| \alpha(x_T - x(t))\varepsilon(t) + \beta(1-\varepsilon(t))\frac{\partial H}{\partial y}(\boldsymbol{\xi}(t)) \right|$$
$$\le \alpha|x_T - x(t)| + \beta\left|\frac{\partial H}{\partial y}(\boldsymbol{\xi}(t))\right| \le 2\alpha R + \beta G. \quad (10)$$

Analogous computations prove the claim for $\dot{y}$. □

Inequality (10) may be used to find α and β such that the velocities always remain below their physical bounds. In fact, suppose that the physical constraints may be modeled as $|\dot{x}| \le V_1$ and $|\dot{y}| \le V_2$ for some $V_1 > 0$ and $V_2 > 0$. Then it is sufficient to choose α and β such that

$$2\alpha R + \beta G \le \min\{V_1, V_2\}. \quad (11)$$

4. Obstacle avoidance for an *n*-dimensional manipulator

The problem of finding a collision-free path in the case of an *n*-DOF manipulator is, from a theoretical point of view, analogous to that of the 2D navigation problem considered above. In fact, they both can be described as follows: given a set $\mathcal{G} \subset \mathbb{R}^n$ (the set of admissible configurations) and two points in $\mathcal{G}$ (the initial configuration and the final configuration) find a curve connecting the two points and belonging to $\mathcal{G}$.

---

[3] We denote with $\mathcal{B}_0(R)$ the (closed) ball of centre 0 and radius R, namely the set $\mathcal{B}_0(R) \triangleq \{\boldsymbol{\xi} : \|\boldsymbol{\xi}\| \le R\}$.

Now, let each configuration of the manipulator be identified by a point $\mathbf{x} = (q_1, \ldots, q_n)^\top$ belonging to some space $\mathcal{X} \subset \mathbb{R}^n$. If the set of all forbidden configurations, being each of them forbidden due to either a collision with an obstacle or to a self-collision [31], can be constructed, then the problem of determining a feasible path between two allowed configurations can be solved with the Hamiltonian approach. In this section we first show how to determine the set of forbidden regions and then how to build the Hamiltonian function, thus solving the path planning problem.

4.1. Self-collisions

We begin by showing how the configurations yielding to a self-collision may be represented. Any configuration $\mathbf{x} \in \mathcal{X}$ maps the points of the robot into points in the Euclidean space $\mathbb{R}^3$. More precisely, consider a feasible pose, i.e. a configuration that does not yield self-collision, and let $P \subset \mathbb{R}^3$ denote the set of points in the Euclidean space occupied by the robot when it is in that pose. Any other configuration $\mathbf{x} \in \mathcal{X}$ maps P into a region $\varphi_\mathbf{x}(P) \subset \mathbb{R}^3$. A self-collision occurs if any two different points $p_1 \in P$ and $p_2 \in P$ are mapped into the same point, namely if $\varphi_\mathbf{x}(p_1) = \varphi_\mathbf{x}(p_2)$. To avoid this occurrence it is necessary and sufficient to require that $\varphi_\mathbf{x}$ be injective. Therefore the configurations that are forbidden because of self-collisions are represented by the set

$$\mathcal{R}_A = \{\mathbf{x} \in \mathcal{X} : \varphi_\mathbf{x} \text{ is not injective}\}. \tag{12}$$

4.2. Collisions with the environment and with the obstacles

In this section we consider collisions with the objects that correspond to the environment, such as the walls and the floor, and collisions with obstacles. These two kinds of collisions may be treated analytically in the same way even though it is convenient to distinguish between the two occurrences, as discussed hereafter.

Let $\mathcal{Y} = \mathcal{X} \setminus \mathcal{R}_A$ and let $\psi : \mathcal{Y} \times P \to \mathbb{R}^3$ be defined by $\psi(\mathbf{x}, p) = \varphi_\mathbf{x}(p)$. Then any point $\mathbf{x} \in \mathcal{Y}$, namely any configuration that does not give rise to a self-collision, may be associated with the region $\psi(\mathbf{x}, P) \subset \mathbb{R}^3$, representing the set of points occupied by the robot (see Fig. 3).

Any object in the environment in which the robot moves is also associated with a region in $\mathbb{R}^3$. We denote with $\mathcal{E}$ the union of the regions corresponding to the floor, the walls and all the other objects of the workspace that may be considered part of the environmental constraints. On the other hand, we denote with $\mathcal{O}$ the union of the regions corresponding to external obstacles. This distinction is superfluous in the case of fixed obstacle but it becomes important in the case of moving obstacles.

If $\psi(\mathbf{x}, P) \cap \mathcal{E} \neq \varnothing$ then the configuration $\mathbf{x}$ yields a collision with the environment; the collection of all such configurations is

the set

$$\mathcal{R}_E = \{\mathbf{x} \in \mathcal{X} : \psi(\mathbf{x}, P) \cap \mathcal{E} \neq \varnothing\}. \tag{13}$$

If $\psi(\mathbf{x}, P) \cap \mathcal{O} \neq \varnothing$, then the configuration $\mathbf{x}$ yields a collision with an obstacle. The collection of all such configurations is the set

$$\mathcal{R}_O = \{\mathbf{x} \in \mathcal{X} : \psi(\mathbf{x}, B) \cap \mathcal{O} \neq \varnothing\}. \tag{14}$$

The set of forbidden configuration is then $\mathcal{X}_F = \mathcal{R}_A \cup \mathcal{R}_E \cup \mathcal{R}_O$.

Remark 4.1. In the following we suppose that the initial and the final configurations of the robot are feasible, namely that they do not belong to $\mathcal{X}_F$.

4.3. Characterization of the trajectory

Once the forbidden region $\mathcal{X}_F$ has been identified, a continuous Hamiltonian function H may be constructed in the following way. Let $\mathcal{B}_\sigma(\mathbf{x})$ be the open ball of centre $\mathbf{x}$ and radius σ and let

$$\mathcal{X}_G = \bigcup_{\mathbf{x} \in \mathcal{X}_F} \mathcal{B}_\sigma(\mathbf{x}). \tag{15}$$

Finally let $H(\mathbf{x}) = 1$ for all $\mathbf{x} \in \mathcal{X}_F$, $H(\mathbf{x}) = 0$ for all $\mathbf{x} \in \mathcal{X} \setminus \mathcal{X}_G$ and extend H to all the points in $\mathcal{X}_G \setminus \mathcal{X}_F$ in such a way that it is continuous for all $\mathbf{x} \in \mathcal{X}$.

The n-dimensional case is conceptually similar to the planar case. The generic structure of a Hamiltonian system of dimension n is

$$\dot{\mathbf{z}} = \mathbf{J}(\mathbf{z}) \nabla H(\mathbf{z}), \tag{16}$$

where $\mathbf{z} = (z_1, \ldots, z_n)^\top$ is the state vector and $\mathbf{J}(\mathbf{z})$ is a skew-symmetric matrix, the off-diagonal entries of which depend on the state. It is easy to check that, since J is skew-symmetric, $\dot{H}(t) = 0$ for all t along the trajectories of (16). Therefore all trajectories lie on a level surface. However, while in $\mathbb{R}^n$ a level surface is a manifold of dimension $n - 1$, the trajectory of system (16) is always a manifold of dimension 1. Therefore when $n = 2$, namely in the planar case, the trajectory support coincides with a level surface, which is a line. On the contrary, in the n-dimensional case many different Hamiltonian systems fulfill the constraint of moving along a trajectory lying on the same level surface.

The degrees of freedom can be exploited to further specify the trajectory. In particular, the trajectory of the Hamiltonian system can be constrained in any hyperplane defined by the equation $\boldsymbol{\gamma}^\top \mathbf{z} = 0$, with $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_n)^\top \in \mathbb{R}^n$. In fact, it is sufficient to impose

$$\gamma_1 \dot{z}_1 + \cdots + \gamma_n \dot{z}_n = 0 \tag{17}$$

to determine, by substitution, the corresponding algebraic conditions on the entries of J. This property is useful if, for practical reasons, some hyperplane turns out to be preferable than others.

5. Discrete implementation

In a practical implementation of the method described in the previous section it is not possible to determine exactly the regions $\mathcal{R}_A$, $\mathcal{R}_E$ and $\mathcal{R}_O$ since the set $\mathcal{X}$ of all possible configurations is an uncountable set. However, one may discretize the problem thus obtaining an approximate solution. In this section we describe a method to obtain such a solution. Let $\mathcal{X}_i$, for $i = 1, \ldots, n$, be the interval to which q_i belongs, let $\mathcal{X}_i^S$ be a set of sampled points of $\mathcal{X}_i$ and let $\mathcal{X}_S = \mathcal{X}_1^S \times \cdots \times \mathcal{X}_n^S$. Since each $\mathcal{X}_i^S$ is finite, so is $\mathcal{X}_S$; let s denote its cardinality. Finally, suppose that the environment Ω in which the robot works is a cube and divide each edge of it into L sub-edges in such a way that the cube may be though of as made of $l \triangleq L^3$ sub-cubes.
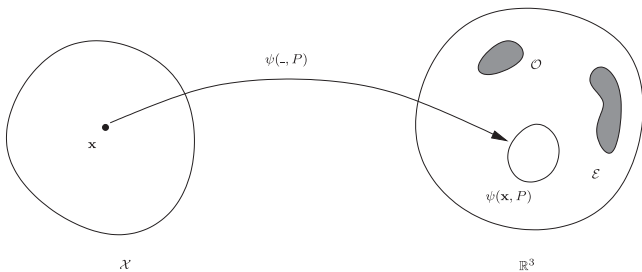

Fig. 3. Any configuration $\mathbf{x} \in \mathcal{X}$ maps the robot P into a region in the Euclidean space $\mathbb{R}^3$. If $\psi(\mathbf{x}, \_)$ is injective and if $\varphi_\mathbf{x}(P)$ is disjoint both from the environment $\mathcal{E}$ and from the obstacles $\mathcal{O}$, then the configuration $\mathbf{x}$ is feasible.

5.1. Forbidden configurations due to self-collisions

The first step in the method is to label the configurations giving rise to self-collisions. This task can be performed by studying all the configurations in $\mathcal{X}_S$ and constructing the set

$$\mathcal{R}_A^S = \{\mathbf{y} \in \mathcal{X}_S : \psi(\mathbf{y}, P) \text{ is not injective}\}. \tag{18}$$

5.2. Forbidden configurations due to collisions with the environment and with the obstacles

To treat the collisions with the obstacles (including the environment) we first enumerate all the configurations in $\mathcal{X}_S \setminus \mathcal{R}_A^S$ from 1 to r and all the sub-cubes from 1 to l. Then, for any of the configurations we mark the sub-cubes that are occupied by the robot. More precisely, for all $i = 1, \ldots, r$, let $\mathbf{x}_i^S \in \mathcal{X}_S \setminus \mathcal{R}_A^S$ be the i-th element in the set of discretized configurations that does not cause self-collisions and for all $j = 1, \ldots, l$ let δ_j be the j-th sub-cube. Construct the matrix

$$C = \begin{pmatrix} c_{1,1} & \cdots & c_{1,l} \\ \vdots & \ddots & \vdots \\ c_{r,1} & \cdots & c_{r,l} \end{pmatrix} \tag{19}$$

with

$$c_{i,j} = \begin{cases} 1 & \text{if } \psi(\mathbf{x}_i^S, P) \cap \delta_j \neq \varnothing, \\ 0 & \text{otherwise}. \end{cases} \tag{20}$$

Note that the matrix C does not depend on the obstacles. It contains information on all the cubes occupied by each of the configurations. Each row of C is associated with a configuration; the j-th entry of the row is 1 if the configuration occupies (even partially) the j-th cube. Each column of C is associated with a cube; the i-th entry of the column is 1 if the i-th configuration occupies (even partially) the cube. Once the matrix C is constructed, it is easy to associate with a given obstacle a set of configurations giving rise to collisions with it. Let O denote the set of points (in the Euclidean space) occupied by an obstacle and let $\mathbf{o} = (o_1, \ldots, o_l)^\top \in \{0, 1\}^l$ be such that

$$o_i = \begin{cases} 1 & \text{if } O \cap \delta_i \neq \varnothing, \\ 0 & \text{otherwise}. \end{cases} \tag{21}$$

The configurations giving rise to collisions with the obstacle O are the elements of the set

$$\mathcal{R}_O^S(\mathbf{o}) = \{\mathbf{x}_i^S \in \mathcal{X}_S \setminus \mathcal{R}_A^S : C_i \mathbf{o} \neq 0\}, \tag{22}$$

where C_i is the i-th row of the matrix C. This relation may naturally be extended to the case of several obstacles. If $O_1, \ldots, O_m$ are the sets of points of m obstacles and $\mathbf{o}_1, \ldots, \mathbf{o}_m$ are the associated vectors constructed as in (21), then the configurations giving rise to collisions with any of the obstacles are the elements of the set

$$\mathcal{R}_O^S = \bigcup_{k=1}^m \mathcal{R}_O(\mathbf{o}_k) = \left\{ \mathbf{x}_i^S \in \mathcal{X}_S \setminus \mathcal{R}_A^S : C_i \sum_{k=1}^m \mathbf{o}_k \neq 0 \right\}. \tag{23}$$

5.3. Construction of a smooth Hamiltonian function

Let $\mathcal{R}_F^S$ denote the set of all forbidden sampled configurations, namely $\mathcal{R}_F^S = \mathcal{R}_A^S \cup \mathcal{R}_O^S$, and denote its cardinality with N. Then a smooth Hamiltonian function can be constructed as follows. For each $k = 1, \ldots, N$ we associate with $\mathbf{y}_k \in \mathcal{R}_F^S$ the function $H_k : \mathbb{R}^n \to \mathbb{R}$ given by

$$H_k(\mathbf{x}) = e^{-\sigma \|\mathbf{x} - \mathbf{y}_k\|^2}, \tag{24}$$

and we define the Hamiltonian function as

$$H(\mathbf{x}) = \frac{1}{N} \sum_{k=1}^N H_k(\mathbf{x}). \tag{25}$$

In Eq. (24), σ is a free parameter that affects the shape of the Hamiltonian function. A large value of σ corresponds to a narrow Gaussian "bell" while a smaller value corresponds to a wider bell. As a consequence, the trajectory avoiding the obstacle generated by a large value of σ is closer to the obstacle than a trajectory corresponding to a smaller value. This property can be exploited to obtain the desired degree of precision (or of safety) of the trajectory.

If the size of the sampling is sufficiently small, then the points are close to each other and it is possible to choose a reference level such that the corresponding connected components represent with a good approximation the forbidden regions.

5.4. Boundedness of the hybrid trajectories

In the case of an n-DOFs robot, the hybrid dynamics (partly linear and partly Hamiltonian) have a structure similar to the one shown in Section 3 for the planar systems. In particular, we define the transition signal as

$$\varepsilon(t) = \begin{cases} 1 & \text{if } H(\mathbf{z}(t)) < H_S, \\ \frac{1}{2} + \frac{1}{2} \cos\left(\frac{H(\mathbf{z}(t)) - H_S}{H_R - H_S} \pi \right) & \text{if } H(\mathbf{z}(t)) \in [H_S, H_R] \text{ or } \rho(\mathbf{z}(t)) < 0, \\ 0 & \text{if } H(\mathbf{z}(t)) = H_R, \end{cases} \tag{26}$$

and we steer the robot according to the hybrid dynamics

$$\dot{\mathbf{z}}(t) = \alpha(\mathbf{z}_T - \mathbf{z}(t))\varepsilon(t) + \beta(1 - \varepsilon(t))\mathbf{J}(\mathbf{z}(t))\nabla H(\mathbf{z}(t)). \tag{27}$$

For the system described by Eqs. (26) and (27) a result similar to Proposition 3.1 can be proven.

Proposition 5.1. *Let $R \in \mathbb{R}^+$ be such that the ball of centre zero and radius R, namely the set*

$$\mathcal{B}_0(R) \triangleq \{\mathbf{z} : |\mathbf{z}| \leq R\}, \tag{28}$$

contains $\mathbf{z}_I$, $\mathbf{z}_T$ and all the level lines corresponding to H_S. If $\mathbf{J}(\mathbf{z})$ is chosen in such a way that there exists $K_1 > 0$ such that, for all $\mathbf{z} \in \mathcal{B}_0(R)$,

$$\max_{i,j=1,\ldots,n} |J_{i,j}(\mathbf{z})| \leq K_1, \tag{29}$$

then there exists M_1 such that

$$|\dot{z}_i(t)| \leq M_1, \tag{30}$$

for all $i = 1, \ldots, n$ and for all $t \geq 0$. □

Proof. Since H is smooth and $\mathcal{B}_0(R)$ is compact, there exists $G_1 > 0$ such that for all $i = 1, \ldots, n$ and for all $\mathbf{z} \in \mathcal{B}_0(R)$,

$$\left| \frac{\partial H}{\partial z_i}(\mathbf{z}) \right| \leq G_1. \tag{31}$$

Hence

$$|\dot{z}_i(t)| = \left| \alpha(z_{T,i} - z_i(t))\varepsilon(t) + \beta(1 - \varepsilon(t)) \sum_{j=1}^n J_{i,j}(\mathbf{z}(t)) \frac{\partial H}{\partial z_j}(\mathbf{z}(t)) \right|$$

$$\leq \alpha|z_{T,i} - z_i(t)| + \beta G_1 \sum_{j=1}^n |J_{i,j}(\mathbf{z}(t))| \leq 2\alpha R + \beta G_1 n K_1. \tag{32}$$

The claim is proven by setting $M_1 = 2\alpha R + \beta G_1 n K_1$. □

Proposition 5.2. *If, in addition to the hypotheses of Proposition 5.1, $\mathbf{J}(\mathbf{z})$ is chosen in such a way that there exists $K_2 > 0$ such that, for all*

$\mathbf{z} \in \mathcal{B}_0(R)$,

$$\max_{i,j,k=1,\ldots,n} \left| \frac{\partial J_{i,j}}{\partial z_k}(\mathbf{z}) \right| \le K_2, \tag{33}$$

then there exists M_2 such that

$$|\ddot{z}_i(t)| \le M_2, \tag{34}$$

for all $i=1,\ldots,n$ and for all $t \ge 0$. $\qquad \square$

Proof. The second time-derivative of z_i is

$$\ddot{z}_i = -\alpha \dot{z}_i \varepsilon + \alpha(z_{T,i} - z_i)\dot{\varepsilon} - \beta \dot{\varepsilon} \sum_{j=1}^{n} J_{i,j}(\mathbf{z}) \frac{\partial H}{\partial z_j}(\mathbf{z})$$
$$+ \beta(1-\varepsilon) \sum_{j=1}^{n} \sum_{k=1}^{n} \left(\frac{\partial J_{i,j}}{\partial z_k}(\mathbf{z}) \dot{z}_k \frac{\partial H}{\partial z_j}(\mathbf{z}) + J_{i,j}(\mathbf{z}) \frac{\partial^2 H}{\partial z_j \partial z_k}(\mathbf{z}) \dot{z}_k \right). \tag{35}$$

Moreover

$$|\dot{\varepsilon}| = \left| \frac{1}{2} \sin\left(\frac{H(\mathbf{z}) - H_S}{H_R - H_S} \pi \right) \frac{\pi}{H_R - H_S} \sum_{i=1}^{n} \frac{\partial H}{\partial z_i} \dot{z}_i \right| \le \frac{\pi n G_1 M_1}{2(H_R - H_S)} \tag{36}$$

and, since H is smooth and $\mathcal{B}_0(R)$ is compact, there exists $G_2 > 0$ such that for all $j=1,\ldots,n$, for all $k=1,\ldots,n$ and for all $\mathbf{z} \in \mathcal{B}$,

$$\left| \frac{\partial^2 H}{\partial z_j \partial z_k}(\mathbf{z}) \right| \le G_2. \tag{37}$$

Hence

$$|\ddot{z}_i| \le \alpha |\dot{z}_i| + \alpha |z_{T,i} - z_i| \, |\dot{\varepsilon}| + \beta |\dot{\varepsilon}| \sum_{j=1}^{n} |J_{i,j}(\mathbf{z})| \left| \frac{\partial H}{\partial z_j}(\mathbf{z}) \right|$$
$$+ \beta \sum_{j=1}^{n} \sum_{k=1}^{n} \left(\left| \frac{\partial J_{i,j}}{\partial z_k}(\mathbf{z}) \right| \left| \frac{\partial H}{\partial z_j}(\mathbf{z}) \right| + |J_{i,j}(\mathbf{z})| \left| \frac{\partial^2 H}{\partial z_j \partial z_k}(\mathbf{z}) \right| \right) |\dot{z}_k|$$
$$\le \alpha |\dot{z}_i| + 2\alpha R |\dot{\varepsilon}| + n\beta |\dot{\varepsilon}| K_1 G_1 + \beta n \sum_{k=1}^{n} (K_2 G_1 + K_1 G_2) |\dot{z}_k|$$
$$\le \alpha M_1 + \beta n^2 (K_2 G_1 + K_1 G_2) M_1 + (2\alpha R + n\beta K_1 G_1) \frac{\pi n G_1 M_1}{2(H_R - H_S)}, \tag{38}$$

which proves the claim. $\qquad \square$

As a consequence of Proposition 5.1, the design parameters α and β can be chosen in such a way that the velocities and the accelerations of the trajectory defined by (26) and (27) are bounded. In fact, suppose that the physical constraints on the velocities and on the accelerations are such that, for all $i=1,\ldots,n$, $|\dot{z}_i| \le V_i$ for some $V_i > 0$ and $|\ddot{z}_i| \le W_i$ for some $W_i > 0$. Then it is sufficient to choose α and β such that $M_1 \le \min_i V_i$ and $M_2 \le \min_i W_i$.

6. Discussion on the implementation

In the present section we discuss some implementation strategies that can be applied when the configuration space is uniformly sampled. It is well-known that uniform sampling is not scalable because the computational requirements grow exponentially. However, two notes are in order. First, the method itself does not require uniform sampling since the Hamiltonian function can have a "sparse" support, therefore ideas from previous work dealing with nonuniform sampling could be employed. Moreover, as explained below, with a proper design we managed to apply a uniform sampling to a 3DOF manipulator with consumer-level hardware (see Section 8).

Now, in the method described in the previous section the construction of the entries of C is a step having a high computational cost, since it is performed by searching, for any possible sampled configuration, all the sub-cubes that are occupied by the robot. Clearly, the finer the sampling of the space $\mathcal{X}$, and the

smaller the edges of the sub-cubes, the larger the dimensions of C and the number of operations needed to determine its entries.

However, the matrix C is a feature of the robot and not of the environment in which it is installed or of the obstacles that are around it. Therefore its determination needs to be performed only once; it could be performed by the robot manufacturer. A similar operative context, where a large amount of data has to be collected off-line only once, arises in other approaches to the real-time path planning problem (see for instance [26]).

Now, despite the fact that the size of C is rl, i.e. the cardinality r of the configuration set $\mathcal{X}_S$ times the cardinality l of the set of sub-cubes, the effective size of the data structure representing the matrix C may be highly reduced by implementing a simple compression algorithm like the classical "run length encoding" (RLE) method.[4] In fact, suppose that the set $\mathcal{X}$ is discretized using a sufficiently fine step; if the i-th configuration occupies the j-th sub-cube, so that $c_{i,j} = 1$, then with high probability the configuration occupies also the $(j-1)$-th and the $(j+1)$-th cubes. Analogously, if the i-th configuration does not occupy the j-th sub-cube, then with high probability it does not occupy the $(j-1)$-th and the $(j+1)$-th cubes. This feature allows one to expect that a compression algorithm, applied row-by-row, could effectively reduce the storage memory of the collision matrix. As a matter of fact, in the experiments reported in Section 8, a compression rate of about 99% has been achieved.

As explained in Section 5.3 to construct the Hamiltonian function one needs to determine the set of forbidden configurations $\mathcal{R}_F^S$ and, in particular, the set $\mathcal{R}_O^S$ of configurations yielding collisions with the environment or with the obstacles. In turn, $\mathcal{R}_O^S$ can be determined by computing the product $C\mathbf{o}$ (see Eq. (23)). As a consequence, if the matrix C is stored in the compressed form described above, each row should be de-compressed before performing the multiplication. However, the de-compression is not necessary since we actually do not need to know the result of the multiplication but only if the result is different from zero. The following algorithm, which avoids the de-compression of the vectors, can therefore be used.

6.1. Fast product algorithm

Suppose that $\mathbf{A} = (a_0, a_1, \ldots)$ and $\mathbf{B} = (b_0, b_1, \ldots)$ represent the compressed form (provided by the run length encoding) of two Boolean vectors $\mathbf{a}$ and $\mathbf{b}$ belonging to $\{0,1\}^{n+1}$ and suppose that we want to know if $\mathbf{a}^\top \mathbf{b}$ is different from zero without de-compressing $\mathbf{A}$ and $\mathbf{B}$. We may use the following algorithm.

1 If $b_0 > a_0$ switch $\mathbf{A}$ and $\mathbf{B}$.
2 If $a_0 = n+1$ the answer is NO. STOP.
3 If $a_0 < b_0 + b_1$ the answer is YES. STOP.
4 Set $\mathbf{B} \leftarrow (b_2 + 1, b_3, b_4, \ldots)$. Set $n \leftarrow n - b_0 - b_1$. Set $a_0 \leftarrow a_0 - b_0 - b_1$. With the updated value of a_0, set $A \leftarrow (a_0 + 1, a_1, a_2, \ldots)$. Go back to step 1.

As an example, consider $\mathbf{A} = (3, 5, 4)$ and $B = (1, 2, 3, 6)$, corresponding to

$$\mathbf{a} = (0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0), \quad \mathbf{b} = (0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1) \tag{39}$$

and $n = 11$. The result of the multiplication is 2 which is different from zero. The algorithm works as follows.

---

[4] To explain the result of a RLE method, it is convenient to consider a Boolean sequence $\mathbf{a}$ corresponding to a sequence of a_0 zeros, followed by a sequence of a_1 ones, followed by a sequence of a_2 zeros, and so on. In this case, in fact, the output of the run length encoding is the sequence $\mathbf{A} = a_0, a_1, a_2, a_3 \ldots$. As an example, if $\mathbf{a} = 00011110000111$, then $\mathbf{A} = 3, 4, 4, 3$. Without loss of generality, one may suppose that $\mathbf{a}$ starts with a zero.
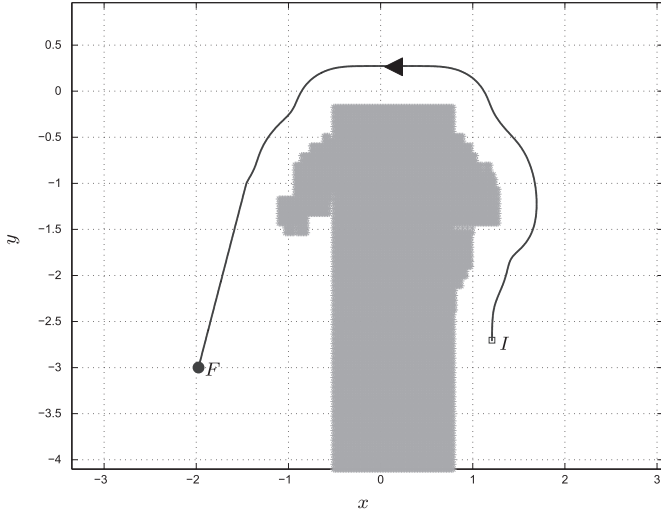
Fig. 4. Trajectory of a single vehicle in 2D space. The gray area represents an obstacle.

Iteration 1

1.1 $b_0 = 1$ and $a_0 = 3$, thus $b_0 > a_0$ is false, hence there is no switch between **A** and **B**. Go to step 2.

1.2 $a_0 = 3$ and $n = 11$, thus $a_0 = n + 1$ is false (hence the answer may not be NO). Go to step 3.

1.3 $a_0 = 3$ and $b_0 + b_1 = 1 + 2 = 3$, thus $a_0 < b_0 + b_1$ is false (hence the answer may not be YES). Go to step 4.

1.4 Set **B** $\leftarrow (b_2 + 1, b_3) = (4, 6)$. Set $n \leftarrow n - b_0 - b_1 = 11 - 1 - 2 = 8$. Set $a_0 \leftarrow a_0 - b_0 - b_1 = 3 - 1 - 2 = 0$. Set **A** $\leftarrow (a_0 + 1, a_1, a_2) = (1, 5, 4)$. Go back to step 1.

Iteration 2

2.1 $b_0 = 4$ and $a_0 = 1$, thus $b_0 > a_0$ is true, hence switch **A** and **B**: **A** $\leftarrow (4, 6)$ and **B** $\leftarrow (1, 5, 4)$. Go to step 2.

2.2 $a_0 = 4$ and $n = 8$, thus $a_0 = n + 1$ is false (hence the answer may not be NO). Go to step 3.

2.3 $a_0 = 4$ and $b_0 + b_1 = 1 + 5 = 6$, thus $a_0 < b_0 + b_1$ is true, hence the answer is YES. STOP.

The performance of this algorithm has been tested and it shows to be about three times faster than the process of de-compressing the vectors and computing the standard product (besides, of course, reducing the memory consumption).

6.2. Use of a local Hamiltonian

An additional way to further speed up the implementation of the method is to use, in the computation of the value of the Hamiltonian function, only the forbidden points lying in a neighborhood of the current state. In fact, the value of the k-th Gaussian function (Eq. (24)) is close to 1 if **x** is close to $\mathbf{y}_k$ while it decreases rapidly when **x** is far from $\mathbf{y}_k$. For instance, if $\|\mathbf{x} - \mathbf{y}_k\| = 2$ and $\sigma = 1$, then $H_k(\mathbf{x}) \simeq 0.018$ which is negligible if compared to 1. Therefore, one may fix a threshold θ and compute the Hamiltonian function at the time t as

$$H(\mathbf{x}(t)) = \frac{1}{M(t)} \sum_{k \in V(t)} H_k(\mathbf{x}(t)), \qquad (40)$$

where

$$V(t) = \{k \in \{1, \ldots, N\} : \|\mathbf{x}(t) - \mathbf{y}_k\| < \theta\} \qquad (41)$$

and $M(t)$ is the cardinality of $V(t)$. The function defined by (40) is not continuous (and not differentiable) for all t since $V(t)$ changes instantaneously when points "enter" or "leave" the neighborhood.
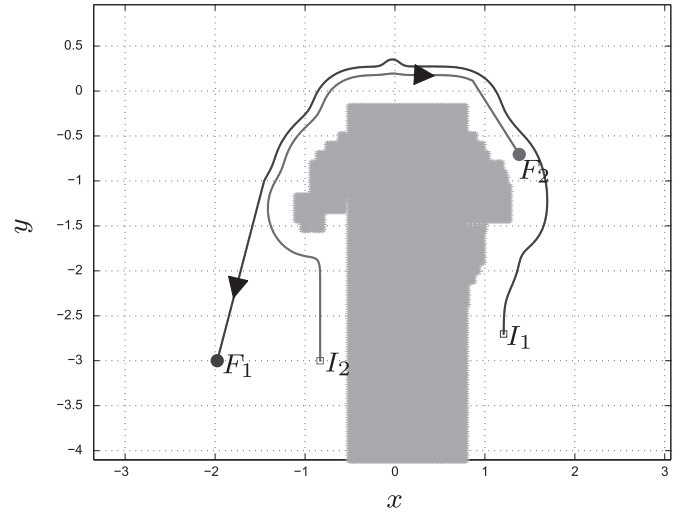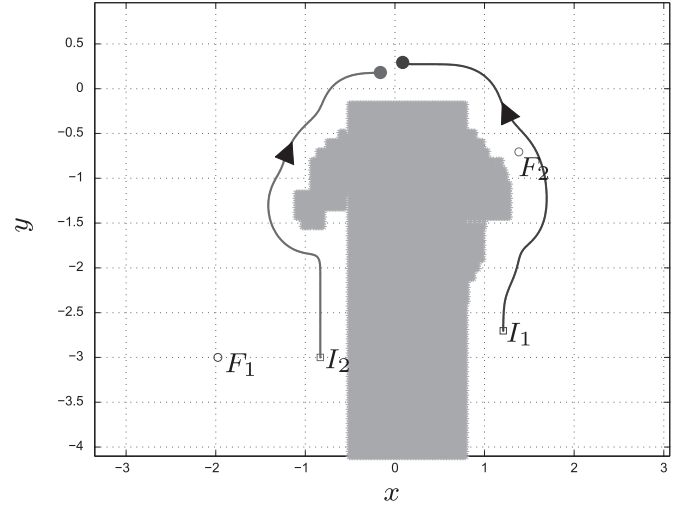

Fig. 5. Trajectory of two vehicles in the 2D space. The gray area represents an obstacle.

However, in a practical implementation, time intervals of constant length T may be considered and the set V updated at time instants that are multiple of T, thus obtaining, for all $i \in \mathbb{N}$ and for all $t \in [iT, (i+1)T)$

$$H(\mathbf{x}(t)) = \frac{1}{M(iT)} \sum_{k \in V(iT)} H_k(\mathbf{x}(t)). \qquad (42)$$

This variation of the method, which yields a piece-wise smooth Hamiltonian function, has been used to obtain the results shown in Section 8.

7. Moving obstacles

The proposed approach allows us to deal also with the case of moving obstacles. If an obstacle is not in a fixed position but moves, namely if the set of points O is not constant but time-varying, then the vector associated with it is also a time-dependent vector $\mathbf{o}(t)$. The method can be applied also in this case, the only difference being that all the quantities are time-varying. In particular,

$$o_i(t) = \begin{cases} 1 & \text{if } O(t) \cap \delta_i \neq \varnothing, \\ 0 & \text{otherwise,} \end{cases} \qquad (43)$$
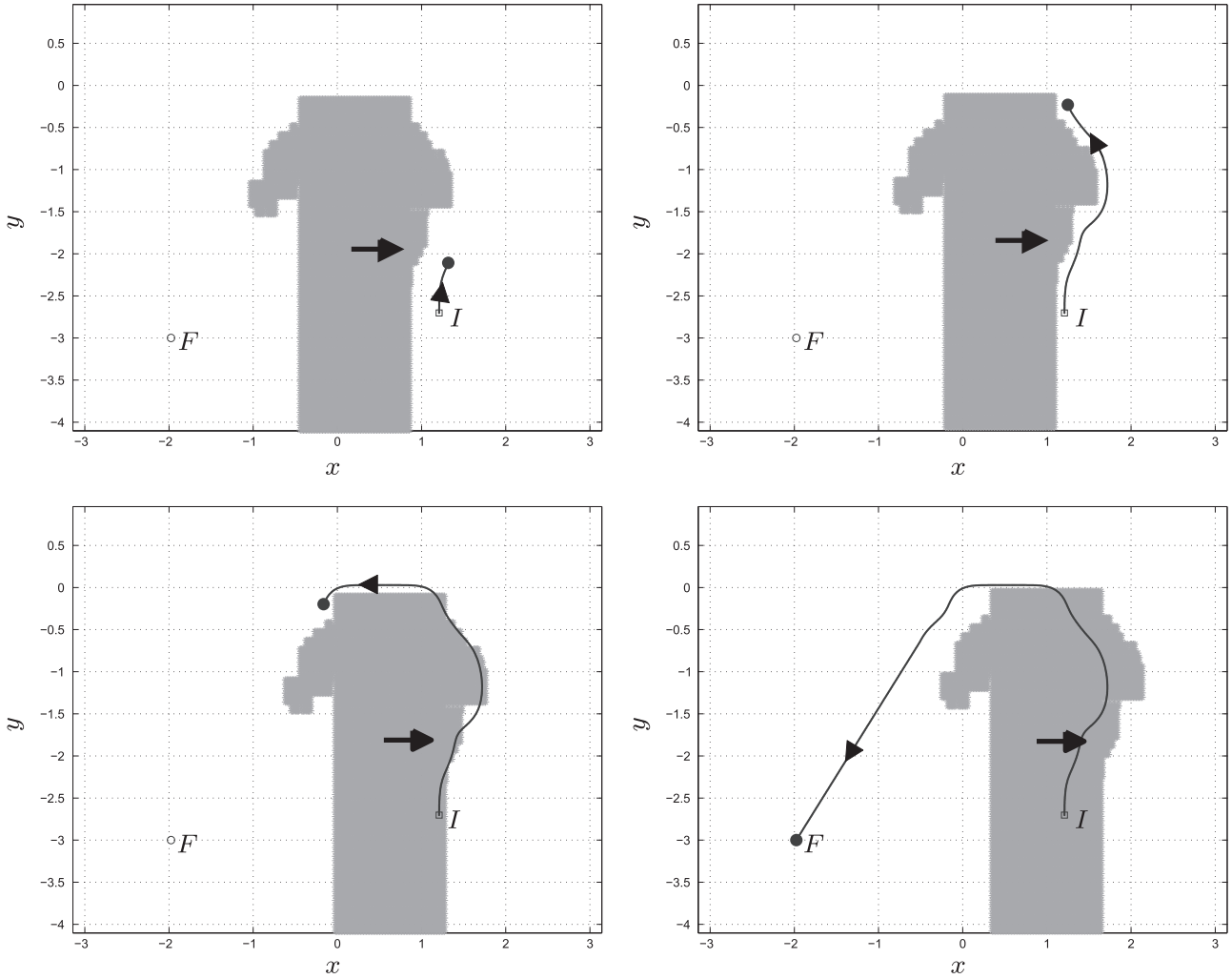
Fig. 6. Trajectory of a vehicle in the 2D space with moving obstacle. The obstacle moves from left to right and the order of the panels is from top to bottom and from left to right.

and the configurations yielding collisions at time t with the obstacle $O(t)$ are the elements of the set

$$\mathcal{R}_O(\mathbf{o}(t)) = \left\{ \mathbf{x}_i^S \in \mathcal{X}_S \setminus \mathcal{R}_A^S : C_i \mathbf{o}(t) \neq 0 \right\}. \tag{44}$$

Analogously, if $O_1(t), \ldots, O_m(t)$ are the sets of points of m obstacles and $\mathbf{o}_1(t), \ldots, \mathbf{o}_m(t)$ are the time-dependent associated vectors, then the configurations yielding collisions at time t with any of the obstacles are the elements of the set

$$\mathcal{R}_O(t) = \bigcup_{k=1}^{m} \mathcal{R}_O(\mathbf{o}_k(t)) \tag{45}$$

$$\mathcal{R}_O(t) = \left\{ \mathbf{x}_i^S \in \mathcal{X}_S \setminus \mathcal{R}_A^S : C_i \sum_{k=1}^{m} \mathbf{o}_k(t) \neq 0 \right\}. \tag{46}$$

Finally, the Hamiltonian function is

$$H(\mathbf{x}, t) = \frac{1}{N(t)} \sum_{k=1}^{N(t)} e^{-\|\mathbf{x} - \mathbf{y}_k(t)\|^2}. \tag{47}$$

Note that in this case the argument of the Hamiltonian function is $(n+1)$-dimensional, since the function depends also on time. However, if the velocities of the obstacles are slow with respect to the parameters α and β, the method can be applied successfully. The velocity bound for the obstacle depends on the shape of the Hamiltonian function and on the angle between the velocity

vectors of the obstacle and of the robot. Its precise determination is beyond the scopes of this work.

8. Simulation results

In this section the proposed approach is first applied to holonomic vehicles navigating in the 2D space with possibly moving obstacles and then to an industrial manipulator moving in the 3D space.

8.1. Vehicles in 2D environment

In this section we return to the planar case for which it is easier to pictorially show the results of simulations in the presence of moving obstacles. Consider a holonomic vehicle in a 2D environment the configuration of which is given by the pair (x, y), namely by a point in the Cartesian space.[5] In this case, there is no need for mapping the obstacles into the configuration space to find the forbidden configurations: the forbidden configurations are simply those pairs (x, y) belonging to an obstacle. The Hamiltonian function is constructed by (25) and the vehicle moves along a curve

---

[5] Since we assume that the vehicle is holonomic, the direction of motion is not constrained by the vehicle orientation which is therefore neglected.

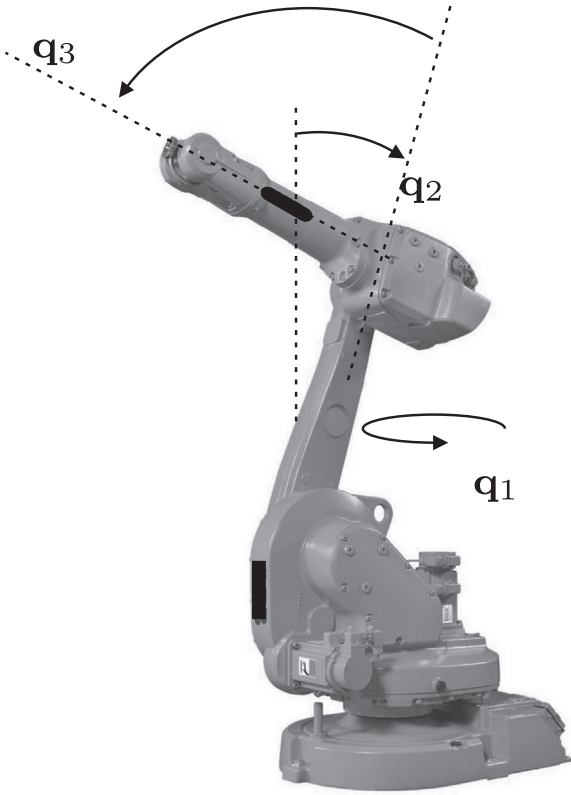Fig. 7. Picture of the robot considered in the simulations.

Fig. 8. The robot in the test environment comprising the floor, a wall and a box. The shown configuration corresponds to the first pose of the trajectory illustrated in Fig. 10.
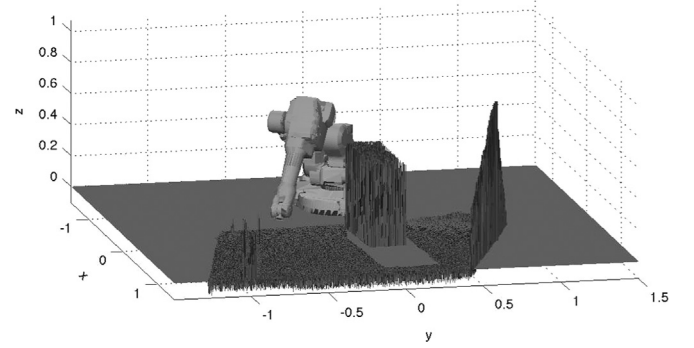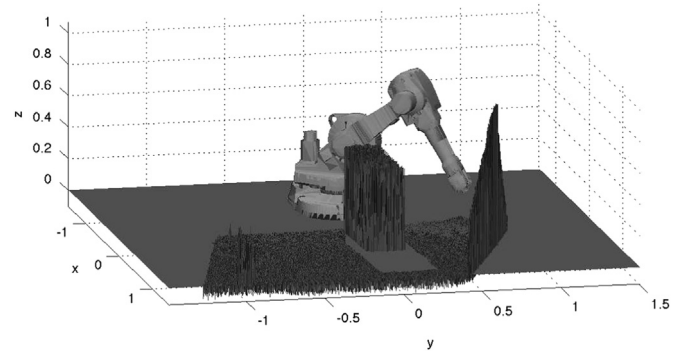

Fig. 9. The robot in the test environment comprising the floor, a wall and a box. The shown configuration corresponds to the last pose of the trajectory illustrated in Fig. 10.
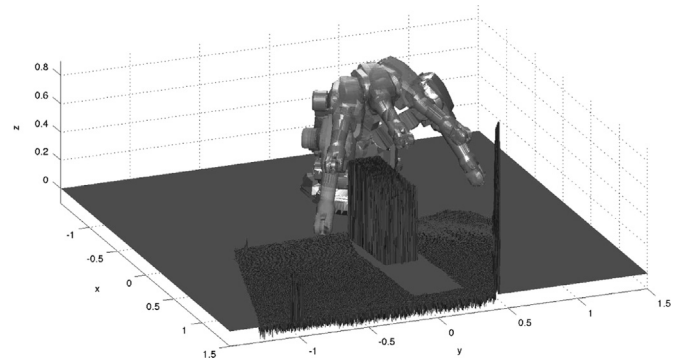

Fig. 10. Five poses belonging to the trajectory (the leftmost pose is the starting one, the rightmost the target).

belonging to the Cartesian space. As a first example, we consider a single vehicle in a static environment, namely with fixed obstacles. In Fig. 4 a trajectory from the initial position (labelled "I") to the final position (labelled "F") is shown. The first part of the trajectory corresponds to a level line of the Hamiltonian function. The last part, due to the fact that the goal becomes visible, is a straight line.

The method can be easily adapted to the case in which there are several vehicles. It is sufficient that each vehicle considers all the other vehicles as obstacles. For instance, in Fig. 5 the case of two vehicles is considered and the corresponding trajectories are shown. To ease the comparison with the previous example, the initial and the final configurations of vehicle 1, as well as the obstacle, are the same as that of Fig. 4; however, the final trajectory is slightly different, since vehicle 1 has to avoid not only the obstacle but also vehicle 2.

Finally, we consider the case of a single vehicle with a moving obstacle (Fig. 6). Also in this case the initial and the final positions, as well as the shape and the initial position of the obstacle, are the same as those used in Fig. 4. Comparing the last part of the trajectory (the straight line) in Fig. 4 with the last part of the trajectory in the case of a moving obstacle (the 4th frame in Fig. 6), it can be noted that in the latter case the straight part of the vehicle trajectory is longer than the corresponding part of the trajectory in the first scenario. This difference is due to the fact that in the situation depicted in Fig. 6 the obstacle is moving and the goal becomes visible earlier than in the former case.

8.2. Industrial manipulator

In the second example we focus on the first 3 DOF of a 6 DOF industrial manipulator, namely an ABB IRB 1600-1.2 (see Fig. 7).

As described above, the first step is to construct the matrix C given the geometry of the links and the kinematics of the robot. The robot workspace can be enclosed in a $3.10 \times 3.10 \times 2.36$ m^3 box. By partitioning the workspace into sub-cubes of sides 0.06 m, we obtain $l = 108\,160$ sub-cubes which correspond to the number of columns of C. The configuration space has been sampled with a resolution of 0.0175 rad, 0.035 rad and 0.0524 rad for q_1, q_2 and q_3, respectively. Given the range of each joint variable, a total of $361 \times 88 \times 98 = 3\,113\,264$ points belonging to $\mathcal{X}_S$ are obtained. Hence C is a $3\,113\,264 \times 108\,160$ Boolean matrix. This step is rather time-consuming[6] but it has to be performed only once since the results depends only on the robot, not on the environment. As explained in Section 5, the memory requirements may result prohibitive; in fact, without compression, the matrix C would occupy as much as 330 GB (in some environments logic variables are stored in a byte). By compressing the matrix with the RLE algorithm, we have obtained a memory consumption of about 3.1 GB, that nowadays can be processed by consumer-level hardware.

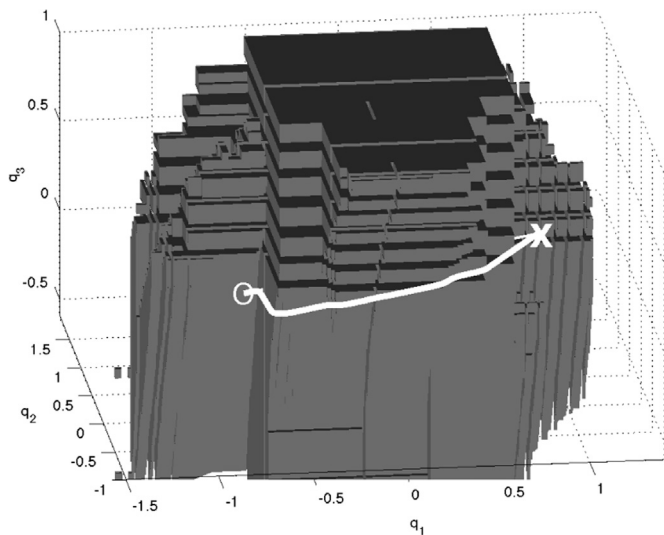[6] It took about 40 h in a Intel-I7-equipped machine.

Fig. 11. A view of the forbidden regions and of the resulting trajectory for the system described in Section 8.2.

In Figs. 8 and 9, the initial and the final pose of the robot in the Euclidean space are depicted, together with some static obstacles (the floor, a wall and a box). The sub-cubes occupied by the obstacles are assumed to be known. This is easy to achieve by using a 3D camera, as it has been done to acquire the obstacles shown in the figure, the flat zones representing the area not covered by the sensor. In Fig. 10 five positions of the robot, belonging to the trajectory determined by the method, are depicted.

The same scenario, but interpreted in the space of the joints, is reported in Fig. 11 where the gray volume corresponds to the set $\mathcal{R}_O^S$, namely the set of forbidden configurations computed by the algorithm described in Section 5, while the small white circle and the small white cross represent the initial and the final pose, respectively. The white line represents the resulting trajectory determined by the hybrid control law.

To test the navigation algorithm the initial pose q^{start} and the final pose q^{goal} have been chosen in such a way that the straight line connecting them intersects the set $\mathcal{R}_O^S$ (see again Fig. 11). The Hamiltonian function is constructed by using (25) with $\sigma = 0.05$.

The matrix J in (16) has been computed as explained at the end of Section 4.3 in order to force the trajectory to lie in the plane determined by q^{start}, q^{goal} and $\bar{q} = (1, 2, 0)^T$ (the latter chosen arbitrarily).

9. Conclusions

The recently developed Hamiltonian-based path planning method [5] has been refined by imposing fulfillment of kinematic constraints. Moreover it has been extended to the case of mobile target and obstacles. It has been applied to the case of an *n*-DOF robot showing that it can comply with the requirements of the speed and acceleration limits. The precision of the method is associated with the dimension of the matrices used in the computations. The more precise the requirements on the final trajectories are, the larger the dimension of the matrices that have to be stored is. Future developments will consider non-uniform sampling of the configuration space and applications to non-holonomic vehicles.

References

[1] C. Belta, V. Isler, G.J. Pappas, Discrete abstractions for robot motion planning and control in polygonal environments, IEEE Trans. Robotics 21 (5) (2005) 864–874.
[2] F. Blanchini, F.A. Pellegrino, L. Visentini, Control of manipulators in a constrained workspace by means of linked invariant sets, Int. J. Robust Nonlinear Control 14 (13–14) (2004) 1185–1205.
[3] J. Borenstein, Y. Koren, The vector field histogram – fast obstacle avoidance for mobile robots, IEEE J. Robotics Autom. 7 (1991) 278–288.
[4] D. Casagrande, G. Fenu, F.A. Pellegrino, A. Astolfi, Application of hamiltonian dynamics to manipulator control in constrained workspace, in: Proceedings of the 2013 Conference on Decision and Control, 2013.
[5] D. Casagrande, F. Pellegrino, A. Astolfi, Hamiltonian dynamics for robot navigation, in: Proceedings of the 9th International Conference on Control and Automation (ICCA), 2011, pp. 118–123.
[6] D.C. Conner, A.A. Rizzi, H. Choset, Composition of local potential functions for global robot control and navigation, in: Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 4, IEEE, 2003, pp. 3546–3551.
[7] C. Connolly, J. Burns, R. Weiss, Path planning using Laplace's equation, in: Proceedings of the 1990 IEEE International Conference on Robotics and Automation, 1990, pp. 2102–2106.
[8] C. Connolly, K. Souccar, R. Grupen, A Hamiltonian framework for kinodynamic planning and control, in: Proceedings of the IEEE International Conference on Robotics and Automation, 1995, pp. 2746–2751.
[9] H. Dong, Z. Du, Obstacle avoidance path planning of planar redundant manipulators using workspace density, Int. J. Adv. Robotic Syst. 12 (9) (2014). http://dx.doi.org/10.5772/59973.
[10] B. Feng, Y. Liu, An improved RRT based path planning with safe navigation, Appl. Mech. Mater. 494–495 (2014) 1080–1083.
[11] D. Fox, W. Burgard, S. Thrun, The dynamic window approach to collision avoidance, IEEE Robotics Autom. Mag. 4 (1) (1997) 23–33.
[12] K. Fujimura, Motion Planning in Dynamic Environments, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1992.
[13] S.S. Ge, Y.J. Cui, New potential functions for mobile robot path planning, IEEE Trans. Robotics Autom. 16 (5) (2000) 615–620.
[14] E.G. Gilbert, D.W. Johnson, Distance functions and their application to robot path planning in the presence of obstacles, IEEE J. Robotics Autom. 1 (1) (1985) 21–30.
[15] S.A. Jacobs, N. Stradford, C. Rodriguez, S. Thomas, N.M. Amato, A scalable distributed RRT for motion planning, in: Proceedings of the 2013 IEEE International Conference on Robotics and Automation, 2013, pp. 5088–5095.
[16] Q. Jia, An improved potential field method for path planning, in: Proceedings of the 2010 IEEE Chinese Control and Decision Conference, 2010, pp. 2265–2270.
[17] S. Karaman, E. Frazzoli, Sampling-based algorithms for optimal motion planning, Int. J. Robotics Res. 30 (7) (2011) 846–894.
[18] L.E. Kavraki, P. Svestka, J.-C. Latombe, M.H. Overmars, Probabilistic roadmaps for path planning in high-dimensional configuration spaces, IEEE Trans. Robotics Autom. 12 (5) (1996) 566–580.
[19] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, Int. J. Robotics Res. 5 (1) (1986) 90–98.
[20] J.-O. Kim, P. Khosla, Real-time obstacle avoidance using harmonic potential functions, IEEE Trans. Robotics Autom. 8 (3) (1992) 338–349.
[21] I. Ko, B. Kim, F.C. Park, Randomized path planning on vector fields, Int. J. Robotics Res. 33 (13) (2014) 1664–1682.
[22] D. Koditschek, E. Rimon, Robot navigation functions on manifolds with boundary, Adv. Appl. Math. 11 (4) (1990) 412–442.
[23] S.M. La Valle, Rapidly-exploring random trees: a new tool for path planning, Technical Report 98-11, Computer Science Department, Iowa State University, October 1998.
[24] S.M. La Valle, Planning Algorithms, Cambridge University Press, Cambridge, UK, 2006,, also available at ⟨http://planning.cs.uiuc.edu/⟩.
[25] B. Lacevic, P. Rocco, M. Strandberg, Safe motion planning for articulated robots using RRTs, in: Proceedings of the XXIII International Symposium on Information, Communication and Automation Technologies, 2011, pp. 1–7.
[26] P. Leven, S. Hutchinson, A framework for real-time path planning in changing environments, Int. J. Robotic Res. 21 (12) (2002) 999–1030.
[27] A. Nagata, K. Nonaka, K. Sekiguchi, Model predictive obstacle avoidance control for omnidirectional mobile robots based on fuzzy potential methods, in: Proceedings of the 2014 European Control Conference, 2014, pp. 352–357.
[28] J. Ren, K.A. Mclsaac, R.V. Patel, Modified Newton's method applied to potential field-based navigation for mobile robots, IEEE Trans. Robotics 22 (2) (2006) 384–391.
[29] E. Rimon, D. Koditschek, Exact robot navigation using artificial potential function, IEEE Trans. Robotics Autom. 8 (5) (1992) 501–518.
[30] A. Shkolnik, R. Tedrake, Path planning in 1000+ dimensions using a task-space Voronoi bias, in: Proceedings of the 2009 IEEE International Conference on Robotics and Automation, 2009, pp. 2061–2067.
[31] B. Siciliano, O. Khatib, Springer Handbook of Robotics, Springer-Verlag, New York, 2008.
[32] L. Song, Y. Mao, Z. Xiang, Y. Zhou, K. Du, A study on path planning algorithms based upon particle swarm optimization, J. Inf. Comput. Sci. 12 (2) (2015) 673–680.
[33] M. Stilman, Global manipulation planning in robot joint space with task constraints, IEEE Trans. Robotics 26 (3) (2010) 576–584.
[34] I. Ulrich, J. Borenstein, VFH+: reliable obstacle avoidance for fast mobile robots, in: Proceedings of the 1998 International Conference on Robotics and Automation, 1998, pp. 1572–1577.
[35] Y. Wang, G.S. Chirikjian, A new potential field method for robot path planning, in: Proceedings of the 2000 IEEE International Conference on Robotics and Automation, vol. 2, 2000, pp. 977–982.