Computer-Aided Design and Applications

http://www.cadanda.com

Path Planning of Holonomic and Non-Holonomic Robots Using Bump-Surfaces

Elias K. Xidias[1], Philip N. Azariadis[2] and Nikos A. Aspragathos[1]

[1]University of Patras, {xidias, asprag}@mech.upatras.gr
[2]University of the Aegean, azar@aegean.gr

ABSTRACT

This paper presents a global solution to a general problem of planning an optimum path of a robot moving in two dimensional environments. The proposed method is able to deal with both holonomic and non-holonomic robots moving into dynamic 2D environments with static and/or moving obstacles. The proposed method is based on the Bump-Surfaces concept which is used to represent the entire robot's environment. The introduced method is general and easy to implement and can be applied virtually to any dimensional or point robot. The performance of the proposed method is investigated and discussed through selected simulated experiments.

Keywords: motion-design, path-planning, bump-surfaces.
DOI: 10.3722/cadaps.2008.497-507

1. INTRODUCTION

Path planning in the presence of obstacles is an important problem in robotics with applications in other areas such as simulation and computer aided design [1]. This problem is of particular interest to many fields of research, including global positioning systems (GPS) applications and autonomous robot navigation.

No global solution is known to the general problem of planning collision-free paths for a robot (either holonomic or non-holonomic) moving in 2D environments cluttered with arbitrary shaped obstacles static or dynamic with a priori known trajectories. However, due to the complexity of the problem, several sub-problems dealing with robot arms [2], car-like robots [3], multiple robots [4] and manipulation tasks [5] of this general problem have been addressed in the literature [6].

The path planning problem is formulated in the Configuration Space [7]. The Configuration Space (C-Space) is the space of all possible configurations of the robot. Thus, each pose (position and orientation) of a robot is represented by a point in the C-Space while the obstacles are expanded appropriately. Then, the path-planning problem becomes equivalent to the path planning of a point robot in C-Space.

Path-planning approaches can globally be classified in three classes: roadmap methods, cell decomposition and potential fields. Roadmap methods [8]-[11] generally construct a graph or *roadmap* (representing the robot's configuration space) and search this graph for the shortest path between start and goal configurations. The constructed roadmap is modified by taking into account the non-holonomic constraints of the robot and the nature of the environment (dynamic or static). A big disadvantage of this kind of methods is the fact that computing an effective roadmap is usually a difficult task. Potential field methods [12] force the robot to move according to the influence of an artificial potential field produced by the goal configuration and the obstacles. The goal configuration generates an "attractive" potential and the obstacles generate a "repulsive" potential. The main limitation arises from the existence of local minima in the resulted field, where no descent direction exists for the robot to follow. Cell Decomposition

approaches [1] divide the free space of the robot's workspace into a number of cells. Motion is then planned through these cells. Unfortunately, when the dimension of the C-Space is high or when the complexity of the scene is large, the number of cells becomes too large rendering this kind of methods practically limited.

Recently, Azariadis and Aspragathos [13] introduced a new method for solving the path-planning problem for a robot (either mobile or manipulator) moving in 2D static terrains. The key-element of this method is the representation of the robot's workspace through a single mathematical entity called as *Bump-Surface* (or *B-Surface* for short). The path planning solution is searched on the 3D B-Surface in such a way that its inverse image into the original 2D environment satisfies the given objectives and constraints. Later, Xidias et al. [14]-[18] have demonstrated the capacity of that method to solve a number of different motion design problems involving a variety of robots such as point robots, car-like robots and multiple car-like robots in complicated two and three dimensional environments cluttered with static and/or moving obstacles. In this paper we present a first approach towards unifying the different "Bump-Surfaces' solutions" under a global one with the capacity to handle either dimensional or point robots moving in a dynamic 2D environment.

The proposed approach takes into account the information of the entire robot's environment to solve the given motion-design problem and searches for an optimum or near-optimum solution (if one exists) on the entire problem space represented by a B-Surface. The proposed method utilizes the familiar in the CAD community concept of B-Spline curves and surfaces in order to take advantage of their local-control properties and the underlying fast and robust computational algorithms. This approach yields robot paths which are smooth without sharp corners and self loops. Contrarily to other approaches [1] smoothing can be incorporated as an integral part of the motion-design algorithm, and not as a final improvement to a path solution found by other techniques.

The paper is structured as follows: Section 2 states the motion-design problem under consideration. Section 3 defines the new path-planning space, while Section 4 presents a formal representation of the different types of robots used in this paper. The path-planning problem is described in Section 5 and it is solved in Section 6. Finally, Section 7 demonstrates and discusses the efficiency of the proposed method through multiple experiments applied on vehicle(s) moving in 2D industrial environments.

2. PROBLEM FORMULATION AND NOTATIONS

Let a robot ***A*** moving in a 2D dynamic environment cluttered with arbitrary-shaped obstacles. Let also the following assumptions:

- The robot is represented as a point or as rectangular-shaped body with two rear wheels and two directional front wheels and a limited steering angle [1].
- The robot is moving with constant velocity.
- The obstacles have fixed and known geometry and are static or dynamic (with known trajectories).
- The positions and robot orientations of the start (**ST**) and goal (**GL**) points are fixed and known.
- It is possible to have more than one robot moving at the same time in the same environment (multiple robots).

Under these assumptions, the main problem addressed in this paper is to determine a robot path in a given 2D dynamic environment satisfying the following motion-design constrains:

A. The path should not intersect with the obstacles and should result to a collision-free robot motion.
B. The path should have minimum length, starting at **ST** and ending at **GL.**
C. Each robot should avoid collisions with the other ones (case with multiple robots).
D. The path should be time monotone (dynamic environments).

3. DEFINING A NEW SPACE FOR PATH PLANNING

In this section, we present the construction of the proposed space for performing path planning for robots with either holonomic or non-holonomic constraints, moving in a 2D dynamic environment. The introduced space is represented by a parametric surface, called as Bump-Surface [13]. In the next section, we show how the proposed Bump-Surface is able to represent all the available configurations of the robot through a family of one-parametric curves lying on that surface.

3.1 Defining the Robot's Workspace $\mathcal{W}$

Since the original 2D environment is cluttered with both static and dynamic obstacles we incorporate time as an additional dimension formulating the 3D workspace $\mathcal{W}$. Under this way the original 2D environment is transformed to a 3D "static" workspace [19]. The constructed robot's 3D workspace $\mathcal{W}$ is the Cartesian product of the original 2D environment and the time interval T which is bounded to yield in $[0,t_g]$, where 0 is the initial and $t_g > 0$ is the final time. Without loss of generality it is assumed henceforth that the 2D environment has unit length in each dimension and $t_g = 1$ therefore $\mathcal{W} = [0,1]^2 \times T = [0,1]^3$.

A point $\mathbf{q} = (u_1, u_2) \in [0,1]^2$ is represented in $\mathcal{W}$ by the triple $\mathbf{p} = (u_1, u_2, u_3) \in \mathcal{W}$ where the coordinates (u_1, u_2) denote its position in the original 2D environment at the time instance $t = u_3$. In 3D workspace $\mathcal{W}$, both moving and static obstacles are transformed into "static" by determining the union of their instances in the orthogonal directions $\langle u_1, u_2, u_3 \rangle$. The resulting "static" obstacles are called *3D time-dependent obstacles*. Hence, $\mathcal{W}$ is considered as a unit cube cluttered with 3D time-dependent obstacles. Fig. 1(a) shows a 2D dynamic environment cluttered with one static obstacle and one moving obstacle, while Fig. 1(b) shows the corresponding 3D time-dependent obstacles in $\mathcal{W}$. The light-grey obstacle is static, while the grey obstacle follows a linear motion in the time interval $[0,1]$ with constant velocity $|\tilde{v}_{ob}|$ along a line parallel to the u_1-axis.

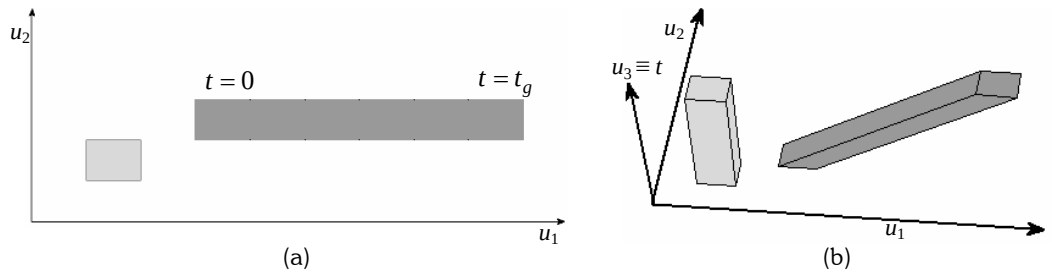

Fig. 1: (a) A simple 2D dynamic environment. (b) The corresponding 3D workspace $\mathcal{W}$.

3.2 Defining the Path Planning Space

The Bump-Surface represents the aforementioned three-dimensional workspace $\mathcal{W}$ of a robot with a 3D manifold embedded in $\Re^4$. In this way the path planning space is defined to be a 4D surface in Euclidean space capturing both the free and the forbidden areas of the robot workspace $\mathcal{W}$.

The required Bump-Surface is obtained by a straightforward extension of the *Z-Value algorithm* [13]. Briefly, this algorithm considers that $\mathcal{W}$ is discretized into uniform subintervals along its $\langle u_1, u_2, u_3 \rangle$, orthogonal directions respectively, forming a grid of points $\mathbf{p}_{j_1,j_2,j_3} = \left(v^1_{j_1,j_2,j_3}, v^2_{j_1,j_2,j_3}, v^3_{j_1,j_2,j_3}, v^4_{j_1,j_2,j_3}\right) \in [0,1]^4$, $0 \le j_1, j_2, j_3 \le M_g - 1$, where M_g denotes the grid size. The $v^4_{j_1,j_2,j_3}$ coordinate of each grid point $\mathbf{p}_{j_1,j_2,j_3}$ takes a value in the interval $(0,1]$, if the corresponding grid point lies inside an obstacle and the value 0 otherwise.

The proposed Bump-Surface is represented by a tensor product B-Spline surface with uniform parameterization $S : [0,1]^3 \rightarrow [0,1]^4$, given by,

$$S = \mathbf{S}(u_1,u_2,u_3) = \sum_{j_1=0}^{M_g-1}\sum_{j_2=0}^{M_g-1}\sum_{j_3=0}^{M_g-1}\left(\prod_{i=1}^{3} M_{j_i,d_i}(u_i)\right)\mathbf{p}_{j_1,j_2,j_3},\ 0 \le u_1,u_2,u_3 \le 1 \quad (1)$$

where d_i, $i=1,2,3$ denote the degree in the $\langle u_1,u_2,u_3\rangle$ directions of the B-Spline Surface respectively and $M_{j_i,d_i}(u_i)$ $i=1,2,3$ correspond to the B-Spline basis functions. Intuitively, the proposed path planning surface consists of "flat" areas where its four coordinate is zero and "bump" areas which make the length of the robot path extremely long when it passes through obstacles. In addition, since the obstacles never intersect each other we know that the topology of the free space is invariant with time thus the free area in the original 2D dynamic environment is represented by the "flat" area on the 4D Bump-Surface.

4. FORMULATION OF DIFFERENT ROBOT REPRESENTATIONS

In this section we formulate the different types of robots mentioned in Section 2 in such a way that their motion in the original environment can be represented by a certain path on the Bump-Surface S.

4.1 Point Robots

A point-robot $\mathcal{A}$ is considered to trace a path $R = \mathbf{R}(s) = (u_1,u_2,u_3)$ in $\mathcal{W}$ (as illustrated in Fig. 2) which is given as a B-Spline curve

$$\mathbf{R}(s) = \sum_{\delta=0}^{K-1} M_{\delta,d}(s)\mathbf{p}_\delta, 0 \le s \le 1 \quad (2)$$

Here, K is the number of the control points $\mathbf{p}_\delta \in [0,1]^3, \delta = 0,\ldots,K-1$, that define the robot's path R, $M_{\delta,d}(s)$ is the B-Spline basis function and d is the curve's degree.

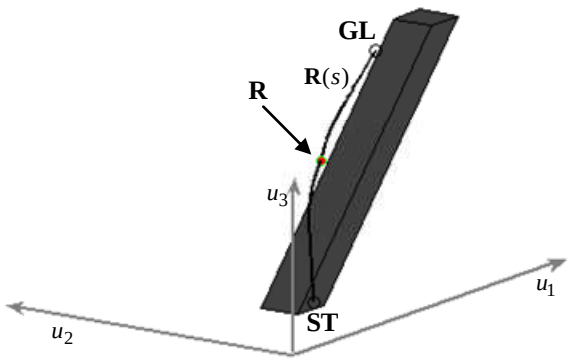

Fig. 2: A point-robot (red dot) moving in $\mathcal{W}$.

4.2 Non-Holonomic Robots

The robot $\mathcal{A}$ has a rectangular shape and is bounded with special, so called *non-holonomic constraints*, than restrict its motion [1]. The configuration of the robot in the original 2D environment is uniquely defined by the triple $(u_1,u_2,\theta) \in [0,1]^2 \times [0,2\pi)$, where (u_1,u_2) are the coordinates of the rear axle midpoint $\mathbf{R}$ with respect to a fixed frame, and θ represents the orientation of the robot, i.e., the angle between the u_1-axis and the main axis of the robot, as it is shown in

Fig. **3**. The *steering angle* $0 \le \phi \le \phi_{\max}$, where $|\phi| = \arctan\left(\frac{l}{\rho}\right) \prec \frac{\pi}{2}$, is defined by the main axis of the robot and the velocity vector of the midpoint $\mathbf{F}$ of the two front wheels, where ρ is the radius of curvature at point $\mathbf{R}$ and ℓ is the distance between $\mathbf{F}$ and $\mathbf{R}$. The orientation θ is linked to the derivative of the position of a reference point $\mathbf{R} = (u_1,u_2)$ on the robot by the equation,

$$\dot{u}_1 \sin\theta - \dot{u}_2 \cos\theta = 0 \tag{3}$$

It is noted that robot's dynamics is not considered in this study.

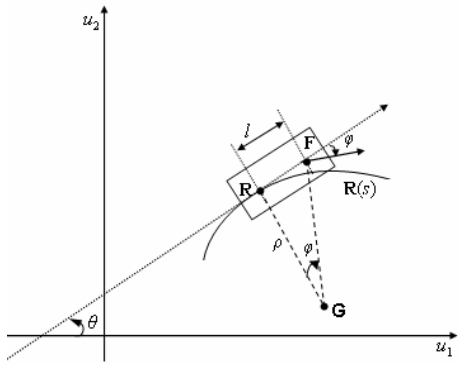

Fig. 3: A non-holonomic robot moving along $\mathbf{R}(s)$. Here $\mathbf{G}$ is the instantaneous centre of rotation of the robot.

We take advantage of the fact that the robot has a maximum steering angle which can be directly connected to the path's radius of curvature ρ [1], as it is shown later in this section.

The midpoint $\mathbf{R}$ traces a path $\mathbf{R}(s) = (u_1(s), u_2(s), u_3(s))$, which has the form given by Eq.(2) [16]. In order to take into account the shape of the robot, a set of feature points $\mathbf{A}_f \in [0,1]^3$, $f = 1,...,V$ is selected on its boundary according to its shape and the requested accuracy (e.g., we set $V = 4$ for a rectangular-shaped robot which correspond to its four corner vertices). Thus, similarly with midpoint $\mathbf{R}$, each feature point $\mathbf{A}_f$ follows a curve $\mathbf{A}_f = \mathbf{A}_f(s)$ in $\mathcal{W}$. By means of Euclidean geometry we can determine easily the path $\mathbf{A}_f(s)$ in $\mathcal{W}$ with respect to $\mathbf{R}(s)$, taking into account that for every $s \in [0,1]$ the distance between every point $\mathbf{A}_f(s)$ and the corresponding midpoint point $\mathbf{R}(s)$ of the rear axle is constant, while the corresponding angles ω_f between vectors $\overrightarrow{\mathbf{R}(s)\mathbf{A}_f(s)}$ and the main axis of the robot are fixed (see Fig. 4).

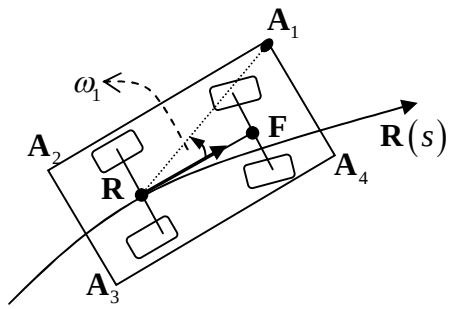

Fig. 4: The robot's position with respect to a given path $\mathbf{R}(s)$.

We measure the "flatness" of the image $S(\mathbf{A}_f(s))$ of $\mathbf{A}_f(s)$ on S by

$$H_f = \int_0^1 S_{v^4}(\mathbf{A}_f(s))ds, \quad f = 1,...,V \tag{4}$$

where, $S_{v^4}\left(\mathbf{A}_f(s)\right)$ denotes the fourth coordinate of $S\left(\mathbf{A}_f(s)\right)$.

Since, the orientation of the robot's front wheels is mechanically limited a turning radius constraint is incorporated in $\mathbf{R}(s)$ which results to a path with a lower bounded turning radius $\rho \geq \rho_{min}$. Therefore, the resulted curvature k of $\mathbf{R}(s)$ is smaller than or equal to $\frac{1}{\rho_{min}}$, i.e.,

$$k(s) \leq k_{max} \Leftrightarrow \frac{\left\|\dot{\mathbf{R}}(s) \times \ddot{\mathbf{R}}(s)\right\|}{\left\|\dot{\mathbf{R}}(s)\right\|^3} \leq \frac{1}{\rho_{min}} \tag{5}$$

where $\dot{\mathbf{R}}(s)$ and $\ddot{\mathbf{R}}(s)$ is the first and second derivative of $\mathbf{R}(s)$ with respect to s, respectively.

Note: In order to determine the points of $\mathbf{R}(s)$ with maximum curvature, we first compute the curvature profile $k(s)$ and then, using the Regula-Falsi method [17], we determine the points where $\frac{dk(s)}{ds} = 0$. At these points, the function $k(s)$ either reaches a local maximum or a local minimum.

4.3 Multiple Robots

When there are multiple moving robots in the same environment we need to coordinate their motions. Assume a set of N non-holonomic robots moving (forwards only) in a 2D dynamic environment. Each robot knows its start and goal position and moves forward with constant magnitude speed $\left|\tilde{\mathrm{u}}^n_{rob}\right|$ $n = 1,\ldots,N$ where the superscript n labels the n^{th} robot (n-robot) [18].

Again we consider that the midpoint $\mathbf{R}^n$ of the rear axle of each n-robot traces a path $\mathbf{R}^n(s) = \left(u_1^n, u_2^n, u_3^n\right)$ in $\mathcal{W}$ given as a B-Spline curve,

$$\mathbf{R}^n(s) = \sum_{\delta=0}^{K-1} M_{\delta,d}(s)\mathbf{p}_\delta^n, 0 \leq s \leq 1,\ n = 1,\ldots,N \tag{6}$$

Here, K is the number of the control points $\mathbf{p}_\delta^n \in \left[0,1\right]^3, \delta = 0,\ldots,K-1$, which define the path R^n. Cleary, the images of the first and last control point of each R^n on S at the time instances $t = 0$ and $t_g = 1$ respectively should not be in collision with obstacles.

In order to take into account the shape of each n-robot we select a set of feature points $\mathbf{A}_f^n \in [0,1]^3,\ f = 1,\ldots,V$ and $n = 1,\ldots,N$, on the boundary of each n-robot. The flatness of the image $S\left(\mathbf{A}_f^n(s)\right)$ of $\mathbf{A}_f^n(s)$ of each n- robot on S is given by:

$$H_f^n = \int_0^1 S_{v^4}\left(\mathbf{A}_f^n(s)\right)ds,\quad f = 1,\ldots,V \text{ and } n = 1,\ldots,N \tag{7}$$

where, $S_{v^4}\left(\mathbf{A}_f^n(s)\right)$ denotes the fourth coordinate of $S\left(\mathbf{A}_f^n(s)\right)$ for each n- robot.

Taking the above into account, Eq. (5) is written for the multiple-robots case as

$$k^n(s) \leq k^n_{max} \Leftrightarrow \frac{\left\|\dot{\mathbf{R}}^n(s) \times \ddot{\mathbf{R}}^n(s)\right\|}{\left\|\dot{\mathbf{R}}^n(s)\right\|^3} \leq \frac{1}{\rho^n_{min}} \tag{8}$$

where $\dot{\mathbf{R}}^n(s)$ and $\ddot{\mathbf{R}}^n(s)$ denote the first and second derivative of $\mathbf{R}^n(s), n = 1,\ldots,N$ with respect to s, respectively.

4.4 Unknown Path Parameters

It must be noticed that in all applications we concentrate on the definition of the $K-2$ control points such that the curve $\mathbf{R}(s)$ (or $\mathbf{R}^n(s)$ for multiple robots) satisfies the path-planning conditions **A**-**D**. The first and last control points, namely $\mathbf{p}_0$ and $\mathbf{p}_{K-1}$ are fixed to the initial and final position of the robot's midpoint $\mathbf{R}$ (or $\mathbf{R}^n$ for multiple robots). Furthermore, the orientation of the robot is linked to the tangent vector (first derivative) of the path $\mathbf{R}(s)$ (or $\mathbf{R}^n(s)$ for multiple robots). In fact, the orientation of the robot at a point $\mathbf{R}(s) \in \mathcal{W}$ equals to the direction of the tangent vector $\dot{\mathbf{r}}(s)$ at that point where $\mathbf{r}(s)$ is the projection of $\mathbf{R}(s)$ on the initial u_1u_2-plane (similarly for multiple robots). The number of the unknown control points depends on the difficulty and complexity of the environment. Using more control points a path with higher flexibility is derived.

5. PATH PLANNING ON THE 4D BUMP-SURFACE

Clearly, based on the above, the most complicated path-planning problem consists of multiple non-holonomic robots which move in a 2D environment with static and dynamic obstacles. The rest of the cases correspond to a limited and special case of this problem. In fact, the case of only one robot is handled by setting $n = 1$, while point-robots require only the determination of path $\mathbf{R}^n(s)$. Since, all path-planning problems described in this paper can be formulated using as unknown parameters the control points of $\mathbf{R}^n(s)$, in this section we present a technique for computing these control points $\mathbf{p}_\delta^n$.

5.1 Shortest Collision Free Path

Generally, a collision-free path should be searched in the "flat" area of the Bump-Surface. A path that "climbs" the bumps of the Bump-Surface results to an invalid path in the original 2D environment that penetrates the obstacles. By construction, the arc length of $\mathbf{R}^n(s)$ approximates the length of its image $S(\mathbf{R}^n(s))$ on S as long as $\mathbf{R}^n(s)$ does not penetrate the obstacles [13]. Therefore, it is reasonable to search for a "flat" path on S in order to satisfy the objectives **A** and **B**.

The arc length L^n of the $S(\mathbf{R}^n(s))$ onto S is given by differential geometry as [20]:

$$L^n = \int_0^1 \sqrt{\left(S_{u_1^n}\frac{d_{u_1^n}}{ds}, S_{u_2^n}\frac{d_{u_2^n}}{ds}, S_{u_3^n}\frac{d_{u_3^n}}{ds}\right)\bullet\left(S_{u_1^n}\frac{d_{u_1^n}}{ds}, S_{u_2^n}\frac{d_{u_2^n}}{ds}, S_{u_3^n}\frac{d_{u_3^n}}{ds}\right)}\,ds \tag{9}$$

where $S_{u_i^n}$ is the partial derivative of S with respect to u_i^n, $i = 1,2,3$ and $n = 1,\ldots,N$.

In addition, in order to take into account that each robot should avoid collisions with the rest ones, we use the predetermined set of feature points $\mathbf{A}_f^n$ in order to designate the regions $CL^n, n = 1,\ldots,N$ of $\mathcal{W}$ which are occupied by robots. Thus the above requirement is expressed by:

$$\bigcap_{n=1}^{N} CL^n = \varnothing \tag{10}$$

The minimization of the following objective function

$$\begin{aligned} &E^n = L^n + \sum_{f=1}^{V} H_f^n, n = 1,\ldots,N \\ &\text{Subject to } \bigcap_{n=1}^{N} CL^n = \varnothing \end{aligned} \tag{11}$$

with respect to the control points $\mathbf{p}_\delta^n \in [0,1]^3, \delta = 1,...,K-2$ and $n = 1,...,N$, leads to a collision free n-path for the n-robot which satisfies the conditions **A** - **C**.

5.2 Time Monotone

Objective **D** expresses the requirement for deriving a *strictly time monotone* $n-$path $\mathbf{R}^n(s)$, which should satisfy the following condition:

$$\dot{\mathbf{R}}^n(s) \succ 0, n = 1,...,N \tag{12}$$

When the sign of each $\dot{\mathbf{R}}^n(s)$ between consecutive roots $\dot{\mathbf{R}}^n(s) = 0$ is positive, then the curve $\mathbf{R}^n(s)$ has an increasing monotone in $\mathcal{W}$. The roots of each $\dot{\mathbf{R}}^n(s) = 0$ are computed using the Regula-Falsi method.

5.3 The Overall Path Planning Problem

Taking the above analysis into consideration, the overall constrained optimization problem is given by,

$$E = \sum_{n=1}^{N}\left(L^n + \sum_{f=1}^{V} H_f^n\right)$$

$$\text{subject to}\begin{cases} \max\left(k^n(s_1),...,k^n(s_\gamma)\right) \le k_{\max}^n, \text{ for robots with nonholonomic constraints} \\ \dot{\mathbf{R}}^n(s) > 0 \\ \bigcap_{n=1}^{N} CL^n = \varnothing \end{cases} \tag{13}$$

The minimization of Eq. (13) with respect to the control points $\mathbf{p}_\delta^n \in [0,1]^3, \delta = 1,...,K-2$, leads to N optimum collision-free paths which satisfy the path-planning objectives **A**-**D**.

6. SEARCHING FOR A NEAR-OPTIMUM ROBOT'S PATH USING A GENETIC ALGORITHM

Genetic Algorithms (GAs) have many features that make them attractive for the solution of combinatorial NP-hard problems. They are theoretically and empirically proven to provide robust search in large and complex search spaces, which can be multimodal and nonlinear [21]. In addition, GAs are well known for their ability of reaching a near-optimal solution for constraint optimization problems. Based on the above we consider that GAs are well adapted in this paper in order to solve problem (13).

The fitness function provides the mechanism for determining the direction of the search on the Bump-Surface. In this paper the fitness function is given by,

$$fitness = \begin{cases} \dfrac{1}{E}, \ if \ \ \max\left(k^n(s_1),...,k^n(s_\gamma)\right) \le k_{\max}^n, \text{ for robots with nonholonomic constraints} \\ \quad and \ \dot{\mathbf{R}}^n(s) > 0, \quad \bigcap_{n=1}^{N} CL^n = \varnothing \\ 0, \ otherwise \end{cases} \tag{14}$$

Each chromosome represents a possible path for the robot as a sequence of control points which define the B-Spline curve of Eq. (6). Each control point consists of coordinates $(u_1, u_2, u_3) \in [0,1]^3$, which are represented by the genes of the chromosome.

7. EXPERIMENTAL RESULTS

The performance of the proposed method is investigated through multiple simulation experiments for a robot moving in 2D environments. These characteristic experiments are presented and discussed in section. The simulations were implemented in *Matlab* and run on a Core 2 Duo 2.13 GHz PC.

The grid size M_g was set equal to 100. The suitable settings for the GA's control parameter were experimentally determined and defined as follow: *population size*=150, *maximum number of generations* =220, *crossover rate* =0.75 and *boundary mutation rate* =0.004. Furthermore, in all the experiments a (2, 2) B-Spline surface was used to represent the Bump-Surface and a 2-degree B-Spline curve is used to represent the robot's path.

Test Case I: Fig 5(a) corresponds to a 2D dynamic environment where the robot has to avoid two moving rectangular-shaped obstacles (dark-grey colour) and seven static obstacles (light-grey colour). The obstacles MO_1 and MO_2 are moving with constant velocity $\left|\tilde{u}_{ob}\right| = 0.25$ in the direction parallel to u_2-axis. The robot is moving with constant velocity $\left|\tilde{u}\right| = 0.5$. Figure 5(b) shows the corresponding 3D workspace $\mathcal{W}$. Figures 5(c)-(d) show two snapshots of the motion of the robot in the original 2D dynamic environment. The solution path (Fig.5(d)) is defined by six control points (including the points **ST** and **GL**).

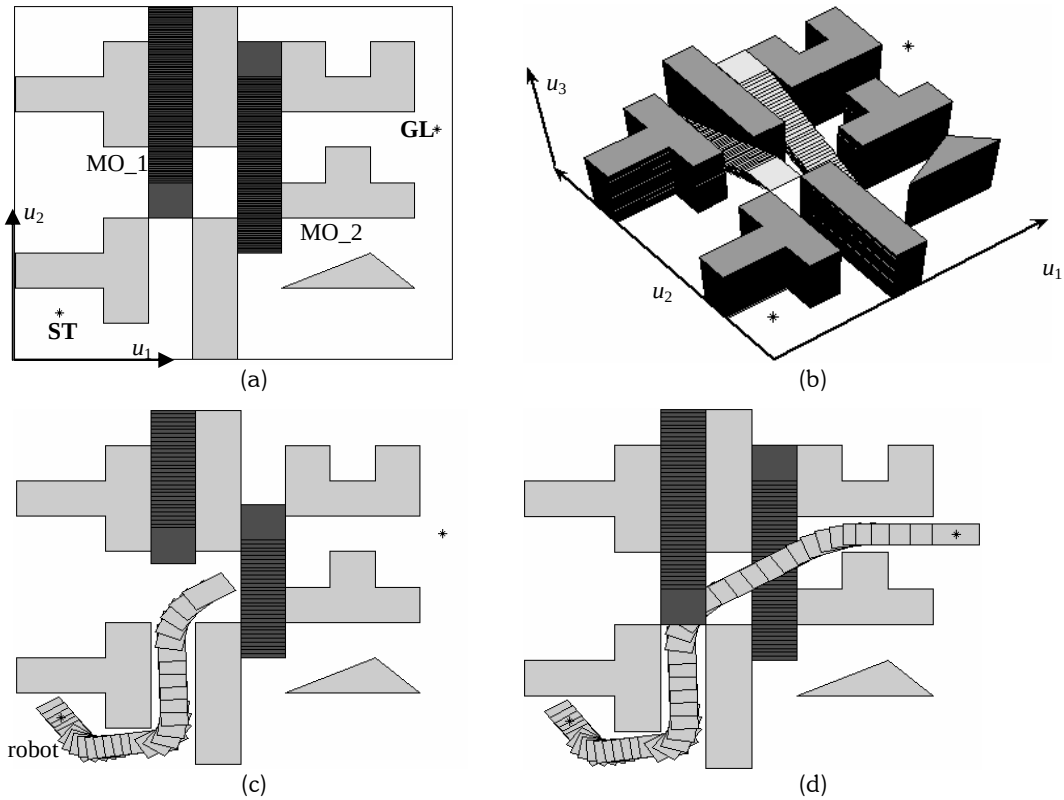

Fig. 5: (a) The original 2D environment. (b) The corresponding 3D workspace. (c) One snapshot of the robot's motion and (d) the solution path.

As it is shown in Fig.5, the robot reaches the GL point without colliding with the obstacles and pass closely to both moving obstacles in order to approach its final position under an optimum path.

Test Case II : Figure 6(a) corresponds to a 2D dynamic environment where two robots have to avoid two moving rectangular-shaped obstacles (dark-grey colour) and seven static obstacles (light-grey colour). The obstacles MO_1 and MO_2 are moving with constant velocity $\left|\tilde{u}_{ob}\right| = 0.25$ in the direction parallel to u_1-axis and u_2-axis, respectively. The robots are moving with constant velocity $\left|\tilde{u}\right| = 0.5$. Figure 6(b) shows the corresponding 3D workspace $\mathcal{W}$. Figures 6(c)-(d) show two snapshots of the motion of the robots in the original 2D dynamic environment. The solution paths

(Fig. 6(d)) are defined by ten control points (including the points **ST** and **GL**). In this experiment we chose an increased number of control points in order to provide more flexibility to the underlying robots' path.

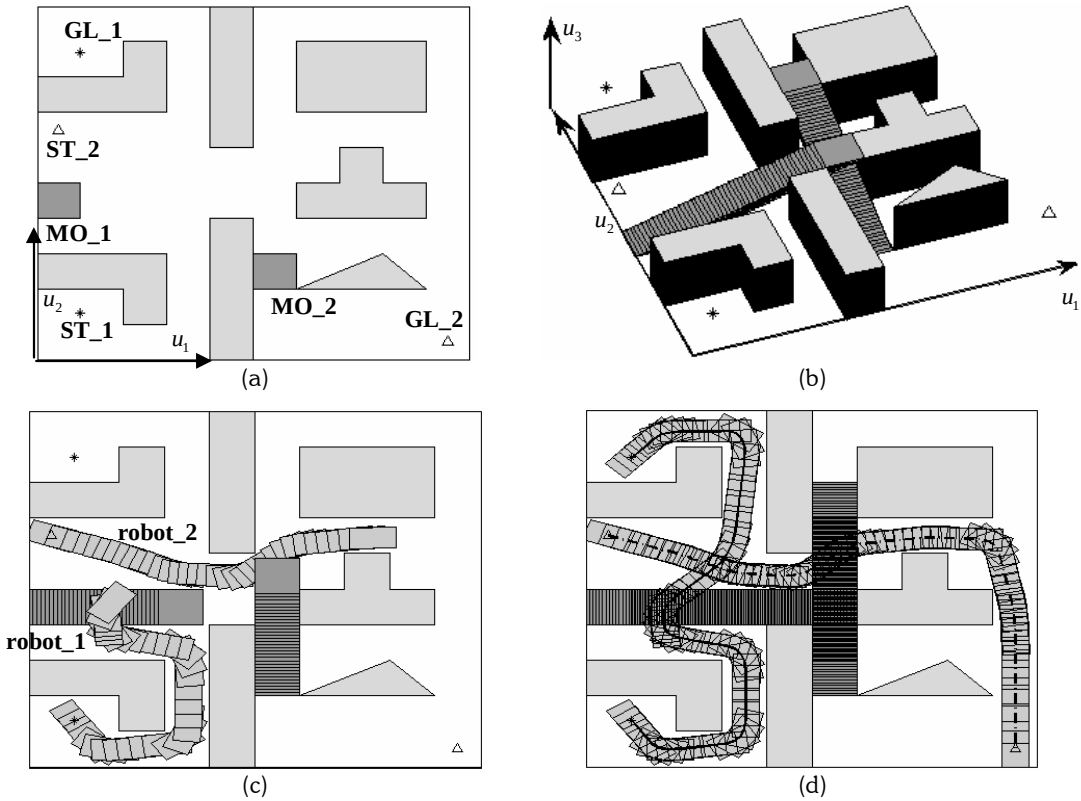

Fig. 6: (a) The 2D environment. (b) The corresponding 3D workspace $\mathcal{W}$. (c) One snapshot of the robots' motion and (d) the solution paths.

The results of all experiments show that the method is effective for determining a near-optimum solution in complicated environments cluttered with obstacles of arbitrary shape, size and location. It should be noticed that with the proposed method the robot is able to take full advantage of its rotational and positional degrees of freedom in order to pass close by the moving and static obstacles. In addition, Fig.6(d) shows that the optimum path is not always the shorter one since in this case robot_1 had to make a small detour in order to avoid the moving obstacle and to maintain its constant speed. Future work will be focused on providing speed control to the moving robots in order to allow velocity variations during the calculation of the optimum path-solution.

8. CONCLUSION

In this paper a general method for determining an optimum (or near-optimum) path for either holonomic or non-holonomic robots moving in 2D dynamic environments is presented. The proposed method consists of two phases: first, the Bump-Surface concept is used to represent the 3D robot's workspace $\mathcal{W}$ by a 3D manifold Bump-Surface embedded in the 4D Euclidean space. The generated Bump-Surface is being searched for a near optimum path satisfying all motion-design constraints by a genetic algorithm. Simulated experiments have shown that the proposed method is able to derive high quality solutions with all types of robots referred in this paper.

9. ACKNOWLEDGMENTS

University of Patras is a partner in the EU-funded FP6 Innovative Production Machines and Systems (I*PROMS) Network of Excellence.

10. REFERENCES

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

[2] Kavraki, L. E.; Svestka, P.; Latombe, J. C.; Overmars, M. H.: Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces, IEEE Transactions on Robotics and Automation, 12(4), 1996, 566-580.

[3] Lamiraux, F.; Laumond, J. P.: Smooth Motion Planning for Car-Like Vehicles, IEEE Trans. of Robotics and Automation, 17(4), 2001, 498-502.

[4] Sánchez, G.; Latombe, J. C.: On delaying collision checking in PRM planning: application to multi-robot coordination, Int. J. of Robotics Research, 21(1), 2002, 5-26.

[5] Saha, M.; Sanchez, G.; Latombe, J. C.: Planning Multi-Goal Tours for Robots Arms, Proc. IEEE International Conference on Robotics and Automation, 3, 3797-3803.

[6] LaValle, S. M.: Planning Algorithms, University of Illinois, 2004.

[7] Lozano-Perez: Spatial Planning: A Configuration Space Approach, IEEE Transactions On Computers, 32(2), 1983, 108-120.

[8] Nearchou, A. C.: Path planning of a mobile robot using genetic heuristics, Robotica, 16, 1998, 575–588.

[9] Pruski, A.; Rohmer S.: Robust Path Planning for Non-Holonomic Robots, Journal of Intelligent and Robotic Systems, 18(4), 1997, 329-350.

[10] Song, G.; Amato, N. M.: Randomized Motion Planning for Car-like Robots with C-PRM, In Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 1, 2001, 37-42.

[11] Vougioukas, S. G.: Optimization of Robots Paths Computed by Randomized Planners, In Proc. IEEE Int. Conf. on Robot Automat., Barcelona, Spain, 2005, 2160-2165.

[12] Rimon, E.; Koditschek, E.: Exact robot navigation using artificial potential functions, IEEE Trans. Robot Automat., 8, 1992, 501-518.

[13] Azariadis P.; Aspragathos, N.: Obstacle Representation by Bump-Surface for Optimal Motion-Planning, Journal of Robotics and Autonomous Systems, 51(2-3), 2005, 129-150.

[14] Xidias E.; Aspragathos, N.: Bump-Hypersurfaces for optimal motion planning in 3D spaces, In Proceedings of International Workshop on Robotics in Alpe-Adria-Danube Region Brno, June 2-5, 2004, 172-177.

[15] Xidias E.; Zacharia P.; Aspragathos, N.: Task scheduling with obstacle avoidance for industrial manipulators operating in 3D environments, In Proceedings I*PROMS NoE Virtual International Conference on Intelligent Production Machines and Systems, 2007, to appear (http://conference.iproms.org/forums/iproms_2007/production_automation_and_control_0/feed).

[16] Xidias E.; Azariadis, P.; Aspragathos, N.: Energy-Minimizing Motion Design for Nonholonomic Robots Amidst Moving Obstacles, Journal of Computer Aided Design and Applications, 3(1-4), 2006, 165-174.

[17] Cooper L.; Steinberg D.: Introduction to Methods of Optimization, W. B. Saunders Company, 1970.

[18] Xidias E.; Aspragathos N.: Motion Planning for Multiple Nonholonomic Robots Using the Bump-Surface Concept, Robotica, accepted for publication, 2007.

[19] Erdmann M.; Lozano-Perez, T.: On Multiple Moving Objects, Algorithmica, 2(1), 1987, 477–521.

[20] Bishop R.; Goldberg S.: Tensor Analysis on Manifolds, Dover, 1980.

[21] Goldberg D. E.: Genetic Algorithm in Search, Optimization and Machine Learning, Addison Wesley, 1989.