Energy-Minimizing Motion Design for Nonholonomic Robots Amidst Moving Obstacles

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

In this paper, a new method is introduced for computing energy-minimizing motions in two-dimensional environments cluttered with a priori known static and moving obstacles. The proposed method is based on a new four-dimensional motion-planning space represented by a Bump-Surface embedded in $\Re^4$. The energy-minimizing motion-design problem is expressed by a variational curve-design problem on the introduced Bump-Surface. The optimal motion is determined by a new algorithm for computing a B-spline curve on the Bump-Surface which is conformal to the motion-planning constrains. The proposed method is applied for designing the motion of nonholonomic robots in the presence of moving obstacles and its performance is tested in simulated 2D dynamic environments with car-like robots.

Keywords: Motion-Planning, Dynamic environment, Nonholonomic robot, Bump-Surface.

1. INTRODUCTION

The problem of determining a time-optimal path for a nonholonomic robot moving in a 2D environment cluttered with static and moving obstacles has attracted the attention of the robotics research community [12]. This interest is motivated by the large class of mechanical systems that belong in this category like the mobile robots or aircrafts and the various challenges that should be overcome to determine a time-optimal path among moving obstacles [11]. Existing motion-planning methods for computing the trajectory of a nonholonomic robot moving in a dynamic environment can be classified into two major categories: potential-field based methods and graph-search based methods.

Potential field methods [7] construct an artificial potential field where the robot is under the influence of an "attractive" potential generated by the goal position, and a dynamic "repulsive" potential generated by the static obstacles and the relative velocity and position of the robot with respect to the moving obstacles. The main limitation arises from the appearance of local minima in the combined (attractive and repulsive) field, where no descent direction exists for the robot to follow.

In the graph-search based methods [2], time is defined as an additional workspace dimension. The required path is planned into a formulated space-time of higher dimension where moving obstacles are transformed to static [5]. Thus, motion-planning among both moving and static obstacles is reduced to motion-planning in a stationary environment. Generally, current works formulate the problem as a two-stage optimization process: in the preprocessing phase a roadmap is constructed with respect to the static obstacles by taking into account the nonholonomic constraints, and then, a feasible robot trajectory is computed on that roadmap (local time-optimality). The resulted path is a sequence of line segments which is an unpleasant feature because the robot is forced to make a complete stop each time it meets a curvature discontinuity. Also, traditional path planning techniques, such as visibility graphs, cannot be directly applied to high dimensional environments [12].

On the other hand, CAGD methods for freeform curve design are able to provide direct solutions to various motion design problems like i.e., moving a robot (either mobile or arm) with respect to a fixed coordinate system [16]. Therefore the generation of one-parametric motions with given properties has been extensively studied by this research community. Another field of current interest is the variational design of curves constrained on surfaces. Large

portion of research focuses on quaternion animation on 3-sphere because of its well known relationship with rigid body motions [10] while newer work concentrates on computing energy-minimizing curves embedded in manifolds of arbitrary dimension [9].

Bump-Surfaces are recently introduced by the last two authors [1] in order to solve the problem of motion-planning in two dimensions (2D) with arbitrary static obstacles. The search method is global and it is based on an energy-minimizing algorithm applied to solve a variational curve-design problem in a subspace of $[0,1]^3$ embedded in $\Re^3$. The Bump-Surface represents the entire 2D working environment of a robot with a two-parametric surface embedded in $\Re^3$, which is everywhere flat (i.e., $z = 0$) except of the areas that correspond to obstacles. These areas have a z-coordinate larger than zero which makes the length of the robot path too long when it passes through obstacles. In this way the robot configuration space is actually a three-dimensional (3D) parametric surface in Euclidean space capturing both the free and the forbidden areas of the robot workspace. The robot motion is determined by a curve onto this surface which, in the original method, expresses the position of the robot in the 2D workspace.

The method presented in [9] constrains motion design of curves onto the six-dimensional manifold of the Euclidean Group SE(3) of rigid body motions and requires the existence of intermediate robot configurations between the start and target configuration. This method is able to handle static obstacles and barriers but it is not clear how to apply it to rotational rigid bodies or to dynamic environments with moving obstacles.

In this paper we exploit further the concept of Bump-Surfaces to address time-optimal path planning in 2D dynamic environments with moving obstacles. First we present a new motion-planning space embedded in $\Re^4$ which is able to capture the dynamic robot workspace via a three-dimensional (3D) manifold represented by a three-parametric surface embedded in $\Re^4$. An energy-minimizing strategy is then applied to the resulted 4D surface in order to determine the motion of a nonholonomic robot through a one-parametric curve that satisfies optimally the given motion-planning objectives. The search problem is solved using Genetic Algorithms [8] with variable length chromosomes [15].

In the context of this paper we concentrate on planning forward motion only, i.e., without backing-up manoeuvres, for nonholonomic robots. However, the energy-minimizing strategy introduced in this paper can be applied to similar motion design problems and it is related to research on interpolating and approximating minimizers on surfaces of arbitrary finite dimension [17], and on snakes on surfaces [13].

This paper is organized as follows. The problem description and general considerations are given in Section 2. Section 3 presents the proposed space for motion-planning, while Section 4 describes the energy-minimizing constrains. Section 5 introduces an algorithm for computing optimal motions in the new motion-planning space. Experimental results are presented and discussed in Section 6, and Section 7 gives some final conclusions and hints for future work.

2. PROBLEM STATEMENT AND GENERAL CONSIDERATIONS

Assume a nonholonomic robot with two rear wheels and two directional front wheels, which is modeled as a rectangle moving in a 2D environment cluttered with a priori known static and moving obstacles that follow known trajectories. The robot has features similar to that of a car, i.e., a limited steering angle which imposes a lower-bounded turning radius.

In this paper we search for a one-parametric curve in the 2D robot workspace which determines a collision-free robot path. The motion-planning criteria and constraints are as follows:

- **i.** The path should not intersect with the obstacles.
- **ii.** The length of the path should be minimal.
- **iii.** The path should have a lower-bounded turning radius.
- **iv.** The path should be time monotone.
- **v.** The path should be smooth without sharp corners and with optimal parameterization.

Concerning the criterion **ii**, it is assumed that the robot is moving with a velocity of constant magnitude $|\tilde{u}_{rob}|$ and therefore a time-optimal path corresponds to the shortest path.

The robot's path is represented by a B-Spline curve in order to take into advantage the local attributes and the well known numerical stability of the respective computational implementations [14]. The configuration of a nonholonomic robot in the original 2D dynamic environment is defined by a triple $(u,v,\theta) \in \Re^2 \times [0,2\pi)$, where (u,v) are the coordinates of the rear axle midpoint $\mathbf{r}$ with respect to a fixed frame, and θ is the orientation of the robot, i.e., the angle between the u-axis and the main axis of the robot. Assuming that point $\mathbf{r}$ follows a trajectory $\mathbf{r}(s)$ in the workspace (Fig.1) the steering angle $0 \le \varphi \le \varphi_{max}$, with $|\varphi| = \arctan(\frac{l}{\rho})$, is defined by the main axis of the robot and the velocity vector at the midpoint $\mathbf{F}$ of the two front wheels, where ρ is the radius of curvature at point $\mathbf{r}$ [12]. In this way one is able to incorporate the robot orientation into a "curve-design procedure" for determining a robot trajectory $\mathbf{r}(s)$ that satisfies the aforementioned motion-planning objectives.

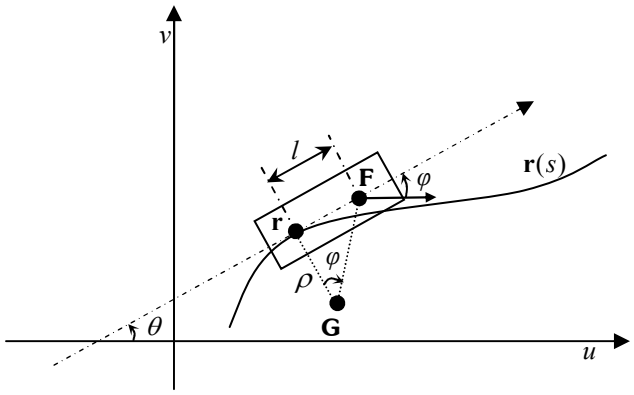

Fig. 1. A simple car-like robot, where $\mathbf{G}$ is the centre of rotation of the robot.

3. DEFINITION OF A NEW SPACE FOR MOTION-PLANNING IN THE PRESENCE OF MOVING OBSTACLES

In this section, we present the construction of the proposed space for performing motion-planning of nonholonomic robots in the presence of moving obstacles. The introduced space is represented by a four-dimensional (4D) parametric surface, called as *Bump-Surface*, which resembles the well known configuration space concept [12]. The proposed Bump-Surface is able to represent all the available configurations of the nonholonomic robot through a family of one-parametric curves lying on that surface. In other words the robot's position and rotation is determined by a curve constrained on the Bump-Surface (see also Section 4) and therefore it is not necessary to incorporate the robot's orientation as a separate surface dimension.

3.1 Incorporating Time in the Initial Workspace (or Defining the 3D Workspace)

The robot's 3D workspace $\mathcal{W}$ is the Cartesian product of the initial 2D robot environment and the time interval $\mathcal{T}$ which is bounded to yield in $[0,t_f]$, where 0 is the initial and $t_f > 0$ is the final time (without loss of the generality it is assumed henceforth that $t_f = 1$), in others words $\mathcal{W} = [0,1]^2 \times \mathcal{T} = [0,1]^3$. The planning will occur directly in $\mathcal{W}$ and it can be treated as the usual Configuration Space [12] with one critical difference: time marches forward, meaning that the path must be time-monotonic (condition **iv**).

A point $\mathbf{q}=(u,v)\in[0,1]^2$ is represented in $\mathcal{W}$ by the triple $\mathbf{p}=(u,v,t)\in\mathcal{W}$ where the coordinates (u,v) denote its position in the initial 2D workspace at the time instance t. Using $\mathcal{W}$ one is able to transform the moving obstacles into static by determining the union of their instances in the $\langle u,v,t\rangle$ orthonormal frame. The resulting obstacles (or volumes) are called as *3D time-dependent obstacles*. Hence, $\mathcal{W}$ can be regarded as a unit cube with embedded volumes (3D time-dependent obstacles) expressing the forbidden motion-planning areas. Figure 2 shows such a 3D time-dependent obstacle in $\mathcal{W}$ produced by a rectangular-shaped obstacle moving in the $uv-$plane using time t as a third dimension. The obstacle follows a piecewise linear motion where: (a) at time interval $[0,t_1]$ is moving with constant velocity $|\tilde{v}_{ob}|$ in direction parallel to the $u-$axis (green-part), (b) at time interval $[t_1,t_2]$ is moving with constant velocity $|\tilde{v}_{ob}|$ in direction parallel to the $v-$axis (red-part), and (c) at time interval $[t_2,1]$ is static (grey-part).

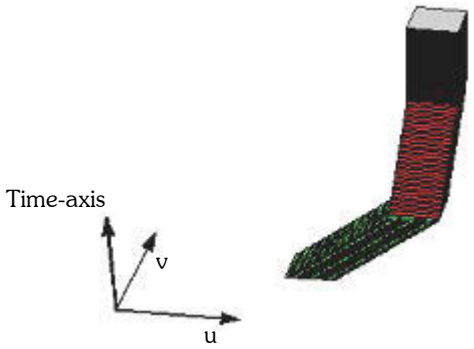

Fig. 2. A 3D time-dependent obstacle.

3.2 Defining the Proposed Motion-Planning Space (or Defining the 4D Bump-Surface)

The Bump-Surface [1] represents the aforementioned 3D workspace of a robot with a 3D manifold embedded in $\Re^4$. In this way the motion-planning is defined to be a four-dimensional parametric surface in Euclidean space capturing both the free and the forbidden areas of the robot workspace. Although the Bump-Surface is considered to extent to infinity, in the rest of this paper we will concentrate in $[0,1]^4$.

Given a 3D workspace $\mathcal{W}$ the construction of the Bump-Surface is a straightforward extension of the *Z-Value algorithm* detailed in [1] and [18]. This algorithm briefly considers that $\mathcal{W}$ is discretized into uniform subintervals along its u,v and t orthogonal directions respectively, forming a grid of points $\mathbf{p}_{ijk}=(x_{ijk},y_{ijk},z_{ijk},w_{ijk})\in[0,1]^4$, $0\le i\le N-1$, $0\le j\le N-1$ and $0\le k\le N-1$, where N denotes the grid size. The w_{ijk} coordinate of each grid point $\mathbf{p}_{ijk}$ takes a value in the interval $(0,1]$ if the corresponding grid point lies inside a 3D time-dependent obstacle and the value 0 otherwise.

The proposed Bump-Surface is represented by a tensor product B-Spline surface [14] with uniform parameterization $S:[0,1]^3\rightarrow[0,1]^4$,

$$S=\mathbf{S}(u,v,t)=\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1}N_{i,p}(u)N_{j,f}(v)N_{k,g}(t)\mathbf{p}_{ijk},\ 0\le u,v,t\le 1 \tag{1}$$

where p,f and g denote the degree in the u,v and t direction of the Bump-Surface respectively. $\mathbf{p}_{ijk}$ are the grid points, and $N_{i,p}(u)$, $N_{j,f}(v)$, $N_{k,g}(t)$ are the basis functions. Intuitively, the proposed motion-planning surface consists of "flat areas" where its fourth coordinate is zero, i.e., $S_w(u,v,t)=0$, and "bump areas" where

$S_w(u,v,t) \in (0,1]$. For simplicity, it is considered that the 3D workspace $\mathcal{W}$ is the actual parametric space of S. Therefore one is able to trace the robot's path in the original 2D dynamic environment through a one-parametric curve $\mathbf{S}(u(s),v(s),t(s))$, $s \in [0,1]$, lying on S. This fact provides the necessary motivation to perform a search for an optimal path on S in order to satisfy the aforementioned motion-planning criteria and constraints **(i)-(v)**.

4. ENERGY-MINIMIZING MOTIONS ON THE 4D BUMP-SURFACE

In the present work the motion-planning objective (**iii**) is directly connected to the path's radius of curvature ρ. Therefore all feasible configurations of the nonholonomic robot are represented onto the Bump-Surface manifold S through a family of one-parametric curves lying on S.

Let $\mathbf{R}$ denote the image of the rear wheels midpoint $\mathbf{r}$ in $\mathcal{W}$. Then $\mathbf{R}$ traces a path given by $\mathbf{R}(s) = (u(s),v(s),t(s)) = (\mathbf{r}(s),t(s))$ in $\mathcal{W}$ which is represented by a B-Spline curve

$$R = \mathbf{R}(s) = \sum_{i=0}^{N_C-1} N_{i,\gamma}(s)\mathbf{p}_i, 0 \le s \le 1 \tag{2}$$

defined in the parametric space of S. Here, N_C is the number of control points, $N_{i,\gamma}(s)$ is the basis function and γ is the curve degree. The image of the path R on S is given by $\mathbf{S}(\mathbf{R}(s))$. The goal of the motion-planning strategy presented in this paper is the definition of the (N_C-2) control points such that curve $\mathbf{R}(s)$ satisfies the criteria and constraints **(i)-(v)**. In fact, we search for a path $\mathbf{S}(\mathbf{R}(s))$ on S such that its inverse image in $\mathcal{W}$ satisfies the motion-planning objectives. Cleary, the image of the first and last control point of R on S at the time instances $t=0$ and $t=t_g \in (0,t_f]$ respectively, should not be in collision with obstacles. The orientation of the robot at a point $\mathbf{R}(s) \in \mathcal{W}$ equals the direction of the tangent vector $\mathbf{r}'(s)$ at that point, where $r=\mathbf{r}(s)$ is the projection of $R=\mathbf{R}(s)$ on the $uv-$plane.

4.1 Satisfying Objectives (i) and (ii)

Following the results from [1] and [18] a valid path which avoids the time-dependent obstacles should be searched in the "flat" areas of the Bump-Surface S. A path that "climbs" the "bumps" of the Bump-Surface results to a path in $\mathcal{W}$ (and hence in the 2D dynamic environment) that penetrates the 3D time-dependent obstacles. By construction, the arc-length of $\mathbf{R}(s)$ approximates the length of its image $\mathbf{S}(\mathbf{R}(s))$ on S, as long as $\mathbf{R}(s)$ does not penetrate the time-dependent obstacles. Therefore it is reasonable to search for "flat" curve $\mathbf{R}(s)$ on S such that $S_w(\mathbf{R}(s))=0$ in order to satisfy both objectives (**i**) and (**ii**).

The arc length of the image of $\mathbf{R}(s)$ onto S is given by [3]

$$L = \int_0^1 \left(S_u^2\left(\frac{du}{ds}\right)^2 + S_v^2\left(\frac{dv}{ds}\right)^2 + S_t^2\left(\frac{dt}{ds}\right)^2 + 2S_uS_v\left(\frac{du}{ds}\frac{dv}{ds}\right) + 2S_vS_t\left(\frac{dv}{ds}\frac{dt}{ds}\right) + 2S_tS_u\left(\frac{dt}{ds}\frac{du}{ds}\right)\right)^{1/2} ds \tag{3}$$

where, S_u, S_v and S_t are the partial derivatives of S with respect to u, v and t, respectively.

In order to take into account the shape of the nonholonomic robot, which should not intersect with the obstacles, a set of vertices $\mathbf{A}_k \in [0,1]^2$, $k = 1...n$ is selected on its boundary according to its shape and the requested accuracy (e.g., we set $n=4$ for a rectangular-shaped robot). Thus, similarly with midpoint $\mathbf{R}$, each vertex $\mathbf{A}_k$ follows a curve $\mathbf{A}_k = \mathbf{A}_k(s)$ on $\mathcal{W}$ where its projection on $uv-$plane gives a curve $\mathbf{a}_k = \mathbf{a}_k(s) \in [0,1]^2$, while its image on S is

given by $\mathbf{S}(\mathbf{A}_{\mathrm{k}}(s))$. The distance between every vertex $\mathbf{a}_{\mathrm{k}}(s)$ and the corresponding midpoint point $\mathbf{r}(s)$ of the rear axle is constant. The same holds also for the angle between vectors $\overrightarrow{\mathbf{a}_{\mathrm{k}}(s)\mathbf{r}(s)}$ and the tangent vector $\overrightarrow{\mathbf{r}'(s)}$ as illustrated in Fig. 3 for $\mathrm{k}=1$. Therefore the location of every vertex $\mathbf{A}_{\mathrm{k}}(s)=\left(\mathbf{a}_{\mathrm{k}}(s),t(s)\right)$ in $\mathcal{W}$ can be easily determined with respect to $\mathbf{R}(s)=\left(\mathbf{r}(s),t(s)\right)$, meaning that it is not necessary to employ a separate expression for determining the path $\mathbf{A}_{\mathrm{k}}\left(s\right)$ on $\mathcal{W}$.

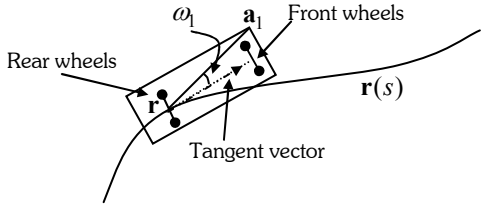

Fig. 3. The robot's position and orientation in the 2D initial environment with respect to a given path $\mathbf{r}(s)$.

We measure the "flatness" of $\mathbf{S}(\mathbf{A}_{\mathrm{k}}(s))$ by

$$H_{\mathrm{k}}=\int_0^1 S_w\left(\mathbf{A}_{\mathrm{k}}(s)\right)ds,\ \mathrm{k}=1...n \text{ and } n\geq 2 \tag{4}$$

where, $S_w(\mathbf{A}_{\mathrm{k}}(s))$ denotes the w -coordinate of $\mathbf{S}(\mathbf{A}_{\mathrm{k}}(s))$.

Therefore, the minimizer of

$$E_{ext.}=\sum_{\mathrm{k}=1}^{n} H_{\mathrm{k}}+L \tag{5}$$

with respect to the control points $\mathbf{p}_i$, corresponds to the minimum external energy of $\mathbf{R}(s)$ and leads to a collision free path which satisfies the conditions **(i)** and **(ii)** .

4.2 Satisfying Objective (iii)

The third objective expresses the requirement for a minimum turning radius ρ_{min} in the path. The curvature $k(s)$ of the $\mathbf{R}(s)$ path should be smaller than or equal to $\frac{1}{\rho_{min}}$, i.e.,

$$k(s)\leq \frac{1}{\rho_{min}}=k_{max} \text{ or } \frac{\left\|\mathbf{R}'(s)\times\mathbf{R}''(s)\right\|}{\left\|\mathbf{R}'(s)\right\|^3}\leq k_{max} \tag{6}$$

By computing the roots of the derivative of $k(s)$ one is able to determine the curve points $\mathbf{R}\left(s_i\right), i=1,...,\gamma$ with maximum curvature. Solving

$$k'(s)=0 \tag{7}$$

using the Regula-Falsi method [4] between the inflection points [14] of $\mathbf{R}(s)$, one obtains a new expression

$$\max\left(k(s_1),...,k(s_\gamma)\right)\leq k_{max} \tag{8}$$

for satisfying objective **(iii)**.

4.3 Satisfying Objective (iv)

Objective **(iv)** expresses the requirement for a time-monotone path $\mathbf{R}(s)$ with increasing monotone in the 3D workspace $\mathcal{W}$ to ensure that the robot does not go back in time. In other words,

$$R_s > 0 \tag{9}$$

where R_s is the derivative of $\mathbf{R}(s)$ with respect to s.

When the sign of R_s between consecutive roots $\left(R_s(s)=0\right)$ is positive, then curve $\mathbf{R}(s)$ has increasing monotone in the 3D workspace $\mathcal{W}$ [6]. The roots of R_s are again computed using the Regula-Falsi method.

4.4 Satisfying Objective (v)

Objective **(v)** expresses the requirement for a smooth path $\mathbf{R}(s)$ without sharp corners and loops and with an optimal parameterization, allowing, thus, for a smooth robot motion with constant velocity. These requirements are expressed by:

$$E_{int} = \int_0^1 \left|\mathbf{R}'(s)\right|^2 ds + \int_0^1 \left|\mathbf{R}''(s)\right|^2 ds \tag{10}$$

The minimizer of E_{int} with respect to control points $\mathbf{p}_i$ subject to the constraints (8), (9) results to the minimization of the internal energy of $\mathbf{R}(s)$ and leads to a path satisfying optimally the objectives **(iii)** and **(v)**.

4.5 The Overall Motion-Planning Problem

Taking the above into consideration the final global optimization problem is written as:

$$\min_{\mathbf{p}_i} E_{comb} = \min_{\mathbf{p}_i} \left(a_1 E_{ext} + a_2 E_{int}\right) \tag{11}$$

Subject to $\max\left(k(s_1),...,k(s_\gamma)\right) \le k_{max}$

$$R_s > 0$$

where, E_{comb} is the combined energy and its minimization with respect to $\mathbf{p}_i$, $i=1,\ldots,N_C-2$ leads to a continuous collision-free path which satisfies the motion-planning objectives $(\mathbf{i})-(\mathbf{v})$. The scalars $a_1+a_2=1,\ 0\le a_1,a_2\le 1$ are weight factors which means that if for example we want to search for shorter paths ignoring to some degree their smoothness we must put $a_1 > a_2$.

5. COMPUTING THE OPTIMAL ROBOT MOTION

Equation (11) corresponds to a nonlinear optimization problem with nonlinear constrains. We propose the use of Genetic Algorithms (GAs) to solve this problem because of their ability to search exhaustively large and complex spaces, reaching a global near- optimal solution in the presence of multiple local minima. In order to apply GAs one has to select the "genetic operators" which include crossover, mutation and selection [8]. In the proposed method we use uniform crossover, a Gaussian mutation operator and the Roulette Wheel selection. Each chromosome represents a possible path for the robot as a sequence of the unknown control points $\mathbf{p}_i$ which define the B-Spline curve $\mathbf{R}(s)$ (Eqn.(2)). We use the following *fitness function* in order to minimize E_{comb}:

$$E_{ga} = \frac{1}{E_{comb}+Q+X} \tag{12}$$

Subject to $R_s > 0$

where, Q is given by

$$Q = \frac{L}{D_E} e^{-\frac{D_E}{L} N_{\max}} \tag{13}$$

$N_{\max}$ is the maximum number of control points of $\mathbf{R}(s)$, L the arc length of the image of $\mathbf{R}(s)$ onto S and D_E is the Euclidean length of the straight line connecting the starting and the destination point defined in $\mathcal{W}$. Function X is given by:

$$X = \frac{\max\left(k(s_1),...,k(s_\nu)\right)}{k_{max}} \tag{14}$$

Eqn.(13) allows the GA to determine the optimal number of control points that define the robot's path, while Eqn.(14) defines the curvature constraint. Each chromosome represents a possible path for the robot as a sequence of control points which define the B-Spline curve of Eqn.(2). Each control point consists of coordinates $u, v, t \in [0,1]$ which are represented by the *genes* of the chromosome.

Finally, paths $\mathbf{R}(s)$ with a decreasing monotone ($R_s < 0$) are excluded from being selected by the GA by assigning them a very small fitness value. In this way, these paths do not take part in the population of the next generation of solutions.

6. EXPERIMENTAL RESULTS

The performance of the proposed method is investigated with a car-like robot moving in various 2D dynamic environments. The results indicate that the new method determines a near time-optimal path whenever it exists. Due to the page limits of this paper, we present the results obtained by the proposed method only in two scenes (corresponding video files (avi) can be found in http://www.mech.upatras.gr/~Robotics).

The overall method is implemented on a Pentium IV 3.2 GHz PC using Matlab. The size of the car-like robot is 0.075 by 0.025 environment units (the distance between its two axes is 0.07 units). The grid size is set to $50\times50\times50$, the maximum number of the control points is set to $N_{\max}=10$ and the general GA's control parameters are defined as follow: *population size* $=100$, *number of generations* $=90$, *crossover rate* $=0.7$ and *probability of boundary mutation* $=0.0075$. Finally, in all experiments we set $a_1=0.7$, $a_2=0.3$ in order to increase the importance of the first two motion-planning objectives. Also, a 2-degree B-Spline curve and a (2, 2, 2)-degree B-Spline surface are used to represent the robot path and the Bump-Surface respectively.

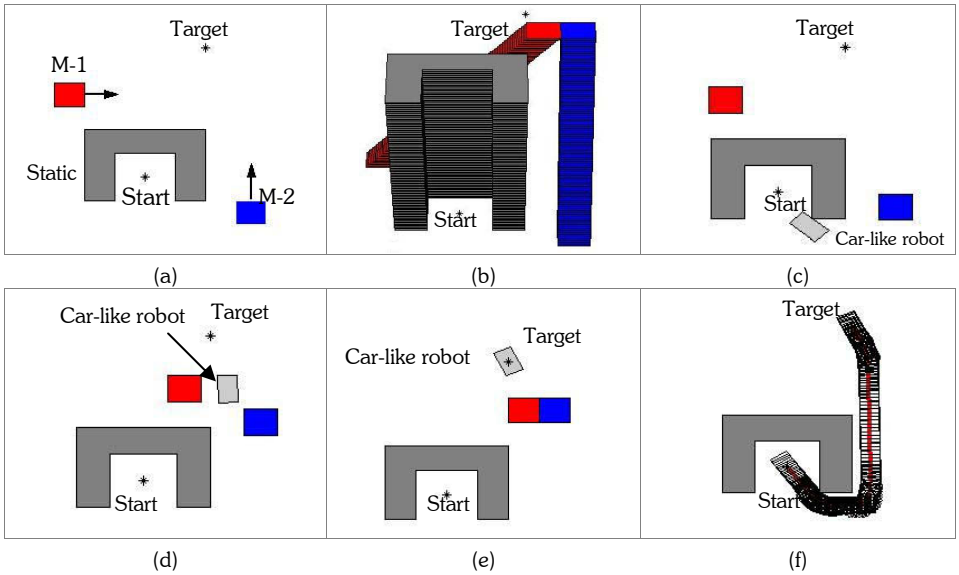

Fig. 4. (a) The initial 2D dynamic environment. (b) The corresponding 3D workspace $\mathcal{W}$. (c)-(e) Three time instances of the robot motion in the dynamic environment. (f) The trace of the final robot's path and the path $\mathbf{r}(s)$ (red curve).

Scene I: Figure 4 shows a car-like robot with steering angle 90° moving in a 2D dynamic environment cluttered with one non-convex static obstacle (grey-obstacle) and two moving rectangular-shaped obstacles. The moving obstacle M-1 (red-obstacle) has constant velocity $|\tilde{v}_{ob1}| = 0.5$ in the direction parallel to u-axis and the moving obstacle M-2 (blue-obstacle) is moving with constant velocity $|\tilde{v}_{ob2}| = 0.5$ in the direction parallel to v-axis while the robot moves with $|\tilde{u}_r| = 1$. Figure 4(a) shows the initial 2D dynamic environment with the start and the target location. Figure 4(b) shows the 3D workspace $\mathcal{W}$ cluttered with the three time-dependent obstacles. Figures 4(c)-(e) show three snapshots of the computed motion of the car-like robot within the dynamic environment. In Fig. 4(c) the robot is moving near to the static obstacle, in Fig. 4(d) it is passing through the moving obstacles and, finally, in Fig. 4(e) it reaches the target location. The robot's path is defined by five control points and it is shown in Fig.4(f). The processing time is 7.98 min.

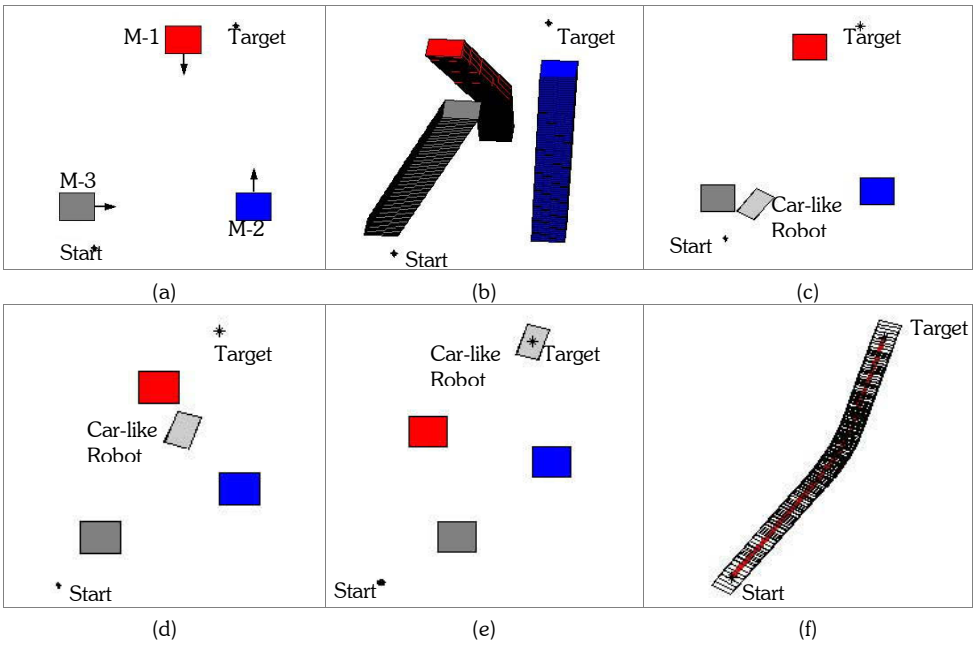

Fig. 5. (a) The initial 2D dynamic environment. (b) The corresponding 3D workspace $\mathcal{W}$. (c)-(e) Three time instances of the robot motion in the dynamic environment. (f) The trace of the final robot's path and the path $\mathbf{r}(s)$ (red-curve).

Scene II: Figure 5 represents a car-like robot with steering angle 40° moving in a 2D dynamic environment cluttered with three moving rectangular-shaped obstacles. The moving obstacle M-1 (red-obstacle) has a piecewise linear motion: (a) in the time interval $[0, 0.5]$ has constant velocity $|\tilde{v}_{ob1}| = 0.25$ in the direction parallel to v-axis, and (b) in the time interval $(0.5, 1]$ has constant velocity $|\tilde{v}_{ob1}| = 0.25$ with a slope 225° with respect to the u-axis. The moving obstacle M-2 (blue-obstacle) has constant velocity $|\tilde{v}_{ob2}| = 0.25$ in the direction parallel to v-axis, and the moving obstacle M-3 (grey-obstacle) has constant velocity $|\tilde{v}_{ob3}| = 0.25$ in the direction parallel to u-axis. The robot moves with $|\tilde{u}_r| = 1$. Figure 5(a) shows the initial 2D dynamic environment with the start and the target location. Figure 5(b) shows the 3D workspace $\mathcal{W}$ with the three time-dependent obstacles. Figures 5(c)-(e) show three snapshots of the motion of the car-like robot in the 2D dynamic environment. In Fig. 5(c) the robot is moving near to obstacle M-3, in Fig. 5(d) the robot is moving near to obstacle M-1 and, finally, in Fig. 5(e) the robot reaches the target location. Finally, the robot's path is defined by three control points and it is shown in Fig.5(f). The processing time is 8.43 min.

Finally, we have compared the proposed motion-design method with the potential-fields method reported in [7] using the same environment reported in that work which consists of three moving and three stationary polygonal obstacles. The proposed method gave a solution in 187 seconds while the method in [7] required 212 seconds. The derived path is again smooth and collision free. Contrarily to [7] the new method is able to take into account the actual shape of the moving robot and the corresponding nonholonomic constraints.

7. CONCLUSIONS

In this paper, a new method for computing planar energy-minimizing motions in the presence of moving obstacles has been proposed. The introduced method is based on a new motion-planning space which is defined by a four-dimensional Bump-Surface incorporating both static and moving obstacles. The computation of the optimal energy-minimizing motion is searched on the new 4D surface using a genetic algorithm with variable length chromosomes.

The proposed methodology has been applied for designing the motion of nonholonomic robots moving in a dynamic environment. Simulations with a car-like robot have shown that the new method works effectively in complicated situations. The computed path was always smooth, conformal to the motion-planning objectives in contrast to existing methods which result to polygonal paths. Future work will be concentrated on applying the Bump-Surface concept in the state × time space including the magnitude of the robot velocity as a further motion-design variable.

8. ACKNOWLEDGMENTS

This work is financed by the Research Committee of the University of Patras as a part of the research project "An Optimal Motion Planning for a Robot based on Computational Geometry" under the K. Karatheodoris frame. University of Patras is a partner in the EU-funded FP6 Innovative Production Machines and Systems (I*PROMS) Network of Excellence.

9. REFERENCES

[1] Azariadis, P. and Aspragathos, N., Obstacle Representation by Bump-Surface for Optimal Motion-Planning, *Journal of Robotics and Autonomous Systems*, Vol. 51, No. 2-3, 2005, pp 129-150.

[2] Berg, J. P. van den Overmars, M.H., Roadmap-based motion planning in dynamic environments, *IEEE Trans. on Robotics and Automation* Vol. 21, 2005, pp 885–897.

[3] Bishop, R. and Goldberg, S., *Tensor Analysis on Manifolds*, Dover, 1980.

[4] Cooper, L. and Steinberg, D., Introduction to Methods of Optimization, W. B. Saunders Company, 1970.

[5] Erdmann, M. and Lozano-Perez, T., On multiple moving objects, *Algorithmica*, Vol. 2, No. 4, 1987, pp 477-521.

[6] Flett, M. -J., *Mathematical Analysis*, Mc Graw-Hill, London, 1966.

[7] Ge, S.-S. and Cui, J.-Y., Dynamic Motion Planning for Mobile Robots using Potential Fields Method, *Autonomous Robots*, Vol. 13, 2002, pp 207-222.

[8] Goldberg, D.-E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley Publishing Company, 1989.

[9] Hofer, M. and Pottmann, H., Energy-Minimizing Splines in Manifolds, *Transactions on Graphics*, Vol. 23, No. 3, 2004, pp 284-293.

[10] Jüttler, B., and Wagner, M., *Kinematics and animation*, In Handbook of Computer Aided Geometric Design, G. Farin, M. S. Kim, and J. Hoschek, Eds. Elsevier, 2002, pp 723–748.

[11] Kolmanovsky, I. and McClamroch, H.-N., Developments in nonholonomic control problems, *IEEE Control Systems*, 1995, pp 20-36.

[12] LaValle, M.-S., *Planning Algorithms*, University of Illinois, 2004.

[13] Lee, Y. and Lee, S., Geometric snakes for triangular meshes, *Computer Graphics Forum*, Vol. 21, No. 3, 2002, pp 229–238.

[14] Piegl, L. and Tiller, W., *The NURBS Book*, Springer-Verlag Berlin Heidelberg, 1997.

[15] Nearchou, A., Adaptive Navigation of Autonomous Vehicles using Evolutionary Algorithms, *Artificial Intelligence in Engineering*, Vol. 13, No. 2, 1999.

[16] Otto Röschel, Rational motion design—a survey, *Computer-Aided Design*, Vol. 30, No. 3, 1998, pp 169-178.

[17] Pottmann, H. and Hofer, M., *A variational approach to spline curves on surfaces*. Tech. Rep. 115, TUWien, Inst. Geometry, http://www.geometrie.tuwien.ac.at/ig/papers/vscs.pdf, 2003.

[18] Xidias, E. and Aspragathos N., Bump-Hypersurfaces for optimal motion planning in 3D spaces, *Proceedings of RAAD'04*, 2004, pp 172-177.