Time Optimum Motion Planning for a Set of Car-Like Robots

Elias k. Xidias*, Philip N. Azariadis,**
Nikos A. Aspragathos*

**Department of Mechanical & Aeronautics Engineering, University of Patras, Greece (Tel: +302610996248; e-mail: xidias@ mech.upatras.gr, asprag@mech.upatras.gr).*
*** Department of Product and Systems Design Engineering, University of the Aegean (e-mail: azar@aegean.gr)}*

Abstract: This paper presents a global solution for simultaneously planning of time optimum motions for a set of car-like robots, which are moving on a two-dimensional general terrain. The method obtains the geometric paths and vehicles speeds that simultaneously minimize motion time for all the robots, considering robots' kinodynamics (kinematic and dynamic constraints) and terrain geometry (terrain topography and obstacles). The proposed method is based on the Bump-Surface concept where the robots' environment is represented by a higher dimension B-Spline surface and the robots' paths consist of B-Spline curves mapped to the initial robots' environment. A global optimization problem is then formulated considering simultaneously the optimal time motion and path planning between the obstacles for all the robots. The performance of the proposed method is investigated and discussed through simulated experiments.

Keywords: Motion Planning, Car-Like Robots, Kinematics, Bump-Surface.

1. INTRODUCTION

Now days, autonomous robots are about to become an important element of the "factory of the future", Ting et al. (2002). Their flexibility and ability to react in different situations open up totally new applications, Podsedkowski et al. (2001). Thus, the problem of finding a valid path between a starting point and a target point in an environment cluttered with obstacles (either static or moving) attracted the attention of many researchers, Latombe (1991).

Several published papers focused to the problem of finding optimal motions for car-like robots. Dubins (1957) determined the shortest paths for a car-like robot that can only move forwards with constant speed under a bounded steering angle. The proposed paths were determined by a finite sequence of straight-line segments and circular arcs with minimum radius. Later, Reeds and Shepp (1990) extended Dubins' results for a car-like robot considering backward motion too. Since then, almost all proposed methods compute the shortest paths by such a sequence where the paths are C^2 along elementary components and C^0 between them. It is therefore necessary to smooth out the derived paths by using techniques such as the clothoids, Fleury et al. (1995) or by interpolating Bezier curves, Jolly et al. (2009).

The motion planning for a set of car-like robots in a dynamic environment is inherently hard. Even for a simple case in two dimensions, the problem is **NP**-hard, Lavalle (2004). Note: A problem is in **NP** (nondeterministic polynomial) if there is a polynomial-time algorithm to verify a solution to the problem. This means that a problem in **NP** can be solved in polynomial time if we have an infinite number of computers to verify in parallel all possible branches of search tree of possible solutions to the problem.

Most of the proposed methods can be classified into two major categories: centralized and decentralized approaches, Latombe (1991). In centralized approaches, the robots are considered simultaneously and therefore entail a high-dimensional composite configuration space, Egerstedt et al. (2001). In the decentralized (or decoupled) approaches a collision-free path for each robot is independently determined, followed by a "synchronization procedure" to resolve possible collisions among robots (velocity tuning), Alami et al. (1995). A variation of the decentralized approaches is proposed by Bennewitz et al. (2001) by assigning priorities to individual robots. Then, the robot paths are computed accordingly.

The trajectory planning, Lepetic et al. (2003), Qu et al. (2004), procedure for a car-like robot considers two steps: first, a distance optimal path is computed in order to obtain a feasible path. Then, the curvature profile of this feasible path is calculated (kinematical constraint). Secondly, taking into account the velocity constraints, the velocity profile is formulated along the feasible path. It must be noticed that, the robot motion must respect the dynamic constraints (velocity + acceleration) because of the motor saturation. Therefore, the original velocity profile may not be applicable under the acceleration limit and may needs to be improved if the acceleration exceeds its limit.

In this paper, a new method is presented that combines various aspects of robots' motion planning, such as velocity and path's curvature, in a unified way for a set of car-like robots. Robots' kinodynamics and the geometry of the environment are taken into account to determine simultaneously time-optimum paths for a set of car-like robots, which connects the given "start and goal" points. To minimize the motion time for each robot, the algorithm selects simultaneously the shortest collision-free path and the highest possible speeds that do not violate kinodynamic constraints on robot motion.

The proposed approach is based on the Bump-Surface concept, Azariadis and Aspragathos (2005), to represent the entire robots' dynamic environment by a 3D B-Spline Surface embedded in $\Re^4$ Euclidean space. This surface is able to capture the motions of a set of car-like robots through monoparametric smooth (C^2) curves that satisfy the given trajectory planning criteria and constraints. A global combinatorial optimization problem is then solved on the resulted surface using a Genetic Algorithm (GA), in order to determine the optimum paths of the moving robots.

The rest of this paper is organized as follows. Section 2 gives the problem formulation and notations. Then the proposed motion-planning strategy is presented in Section 3. Section 4 presents the final optimization problem. Section 5 discusses indicative results, while Section 6 concludes the paper.

2. PROBLEM FORMULATION AND NOTATIONS

2.1 Problem Formulations and General Assumptions

Let a set of $\mathcal{N}$ car-like robots moving in a two-dimensional (2D) environment cluttered with arbitrary-shaped obstacles (Note: only forward motion is considered in this paper). Let also the following assumptions:

- Each robot is represented as rectangular-shaped body with two rear wheels and two directional wheels Latombe (1991).
- Each robot is moving with variable velocity.
- The obstacles have fixed and known geometry and are static or moving with known trajectories.
- The positions and robots orientations of the start (**ST**) and goal (**GL**) points are fixed and known.

Under these assumptions, the main problem addressed in this paper is to determine a set of robots paths in a given 2D environment satisfying the following motion planning criteria and constraints:

A. Each path should have a continuous curvature constraint.

B. Each robot should avoid collisions either with the obstacles or with the other robots.

C. Each path should have a lower-bounded turning radius.

D. Each path should be time monotone.

E. Each robot should reach its goal point in minimum time.

2.2 Car-Like Robot: Main Assumptions

It is assumed that, a wheeled mobile robot $\mathcal{A}$ moves like a car-like robot, it has a rectangular shape and its motion is bounded by kinodynamic constraints, Latombe (1991). The robot's configuration in the initial 2D environment is uniquely defined by the triple $(u_1, u_2, \theta) \in \Re^2 \times [0, 2\pi)$, where $(u_1, u_2) \in [0,1]^2$ are the coordinates of the rear axle midpoint **R** with respect to a fixed frame, and θ represents the orientation of the robot, as it is shown in Fig. 1. 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 **F** of the front axis of the robot, where ρ is the radius of curvature at point **R** and l is the distance between the midpoints **R** and **F**. **G** is the instantaneous centre of rotation of the robot. The orientation θ is linked to the derivative of the position of the reference point $\mathbf{R} = (u_1, u_2)$ on the robot by the equation,

$$\dot{u}_1 \sin\theta - \dot{u}_2 \cos\theta = 0 \qquad (1)$$

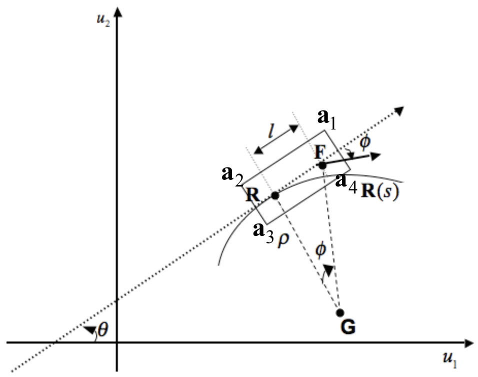

Fig. 1: A wheeled robot moving along a given path $\mathbf{R}(s)$ and $\mathbf{a}_i, i = 1,..,4$ are the robot's vertices.

It must be noticed that a path for a wheeled robot is a continuous sequences of configurations, i.e., a curve in the $u_1u_2\theta$-space, that must verify the kinodynamic constraints. However, because of Eq. (1) this path can be also defined by a u_1u_2-curve that is followed by $\mathbf{R}$, Latombe (1991). This is the approach adopted in this paper.

2.3 Dynamic Constraints of a Car-Like Robot

Following the procedure of Lavalle et al. (2001), the dynamic constraints of a car-like robot are transformed into constraints on the tangential velocity V and tangential acceleration a.

The following relation gives the feasible velocity range,

$$0 \prec V \leq \min\left(V_{\max}, \sqrt{\frac{T_0}{k(s)}}\right) \tag{2}$$

where $k(s)$ is the curvature, $V_{\max}$ is the maximum tangential velocity and T_0 is a constant which depends on the friction between the wheels and the ground, and the gravity constant.

The tangential acceleration boundary conditions must satisfy the following relation,

$$\mathcal{F}_{\min} / m - gk(s) \leq a \leq \mathcal{F}_{\max} / m - gk(s) \tag{3}$$

where $\mathcal{F}_{\min}$ and $\mathcal{F}_{\max}$ is the minimum and maximum braking force respectively, m is the vehicle mass and g is the gravity constant.

2.4 Definition of the Robots' Workspace $\mathcal{W}$

Since the 2D environment is cluttered by both static and moving obstacles, time is incorporated as an additional dimension formulating a 3D workspace $\mathcal{W}$. Therefore the 2D dynamic environment is transformed to a 3D static environment, Xidias et al. (2007). The constructed robots' workspace $\mathcal{W}$ is the Cartesian product of the original 2D environment and the time interval $\boldsymbol{T}$ that is bounded to yield in $[0, t_f]$, where 0 is the initial time and t_f is the overall final time. Without loss of generality it is assumed henceforth that the original 2D environment has unit length in each dimension and $t_f = 1$. Therefore the workspace $\mathcal{W}$ is defined as $\mathcal{W} = [0,1]^2 \times \boldsymbol{T} = [0,1]^3$.

A visual representation of a workspace which results from one static and one moving obstacle is depicted in Fig. 2.

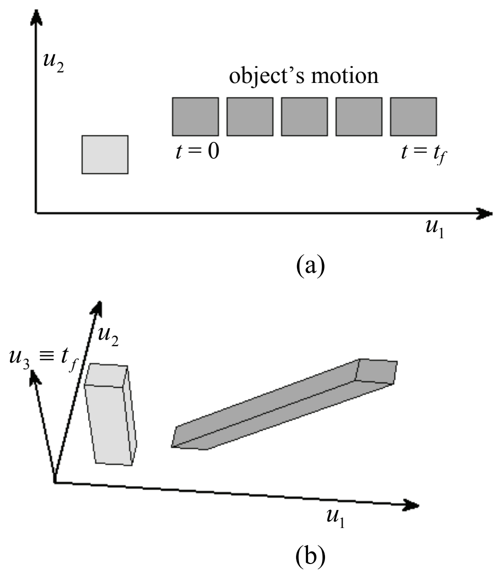

Fig. 2: (a) A 2D environment consisting of one static (light grey) and one moving obstacle (grey). (b) The resulting 3D workspace $\mathcal{W}$.

In the simple case where the initial robots' environment is cluttered with static obstacles, the corresponding workspace $\mathcal{W}$ is defined as $\mathcal{W} = [0,1]^2$.

2.5 Definition of the Motion Planning Space

The Bump-Surface represents the resulted 3D workspace $\mathcal{W}$ with a 3D manifold embedded in $\Re^4$. In this way the motion planning space is defined by a 4D surface in Euclidean space capturing both the free and the forbidden areas of the robots workspace $\mathcal{W}$.

A tensor product B-Spline surface with uniform parameterization represents the Bump-Surface $S : [0,1]^3 \rightarrow [0,1]^4$, which is given by

$$S = \mathbf{S}(u_1, u_2, u_3) = \sum_{j_1}^{M-1} \sum_{j_2}^{M-1} \sum_{j_3}^{M-1} \left(\prod_{i=1}^{3} N_{i,2}(u_i) \right) \mathbf{p}_{j_1 j_2 j_3}, \quad u_i \in [0,1], i = 1,2,3 \tag{4}$$

where, $N_{i,2}(u_i),\ i = 1,2,3$ is the B-Spline base function, $\mathbf{p}_{j_1 j_2 j_3}$ are the control points which are produced by the discretization of $\mathcal{W}$ and M is the grid size. Intuitively, the proposed motion-planning surface consists of "flat" areas where its fourth coordinate is zero and "bump" areas which make the length of the robots' paths extremely long when they pass through obstacles. In addition, since no overlap between obstacles exist, the topology of the free space is time invariant and thus the flat area on the 4D Bump-Surface represents the free area in the original 2D environment. It must be noticed that, in the simple case where the robots' initial environment is static, the robots' workspace $\mathcal{W}$ is two-dimensional and the corresponding Bump-Surface is three-dimensional.

3. MOTION PLANNING

The objective **A** implies that each path must be a curve of class at least C^2. The C^2 property ensures that a path is smooth and thus a car-like robot can follow it without any changes of the motion direction (no cusp points). We adopt the B-Spline curves as the best option to represent the robots paths for the following reasons:

- The mathematical model is simple.
- It is possible to guarantee C^2 continuity in $u_1 u_2$-plane.
- They provide high control flexibility allowing the derivation of solutions in very difficult environments that may be cluttered by narrow passages with a limited number of control points.
- Computation algorithms based on the B-Spline curves are fast and robust.

It is assumed that the midpoint $\mathbf{R}^n$ of the rear wheels of each n-robot traces a path $\mathbf{R}^n(s)=(u_1^n(s),u_2^n(s),u_3^n(s))$, $n=1,...,\mathcal{N}$ given as a B-Spline curve

$$\mathbf{R}^n(s)=\sum_{i=0}^{K-1} N_{i,d}(s)\mathbf{p}_i^n,\ s\in[0,1],\ n=1,...,\mathcal{N} \quad (5)$$

defined in the parametric space of S, which is the actual workspace $\mathcal{W}$. Here, K is the number of control points $\mathbf{p}_i^n$, $N_{i,d}$ is the B-Spline basis function and d is the degree of the B-Spline curve. In all our experiments we set $d=4$. The first and last control point, namely $\mathbf{p}_0^n$ and $\mathbf{p}_{K-1}^n$ are fixed to the initial and final position of the midpoint of the n-robot's rear wheels. In addition, the orientation of the n-robot in each point of the n-path has the same direction with the tangent vector of $\mathbf{R}^n(s)$ at that point.

The goal of the proposed motion-planning strategy is the determination of the $K-2$ control points, which define the B-Spline curve $\mathbf{R}^n(s)$. In general, increasing the number of control points provides more flexibility and better local control to the produced curve. Therefore, in workspaces with narrow passages and/or large number of moving obstacles it is better to use more control points to provide more degrees of freedom to the underlying path.

*3.1 Constraints **B**-**C***

Following the results from Xidias et al. (2007), the $\mathcal{N}$ robots should move in the flat areas of the Bump-Surface. An n-robot that "climbs" the bumps of the Bump-Surface results to an invalid motion in the original 2D environment that penetrates the obstacles. By construction, the arc length of each $\mathbf{R}^n(s)$ approximates the length $\boldsymbol{L}^n(s)$ of its image $S\left(\mathbf{R}^n(s)\right)$ on S as long as $\mathbf{R}^n(s)$ does not penetrate any obstacle. Therefore, it is reasonable to search for a "flat" n-path on S in order to satisfy the objective **B**.

In addition, in order to take into account the geometry of each n-robot we select a set of feature points $A_f^n\in[0,1]^3, f=1,...,4, n=1,...,\mathcal{N}$, which correspond to the vertices of the n-robot. Then we measure the "flatness" H_f^n. Furthermore, in order to take into account the requirement that each robot should avoid collisions with the other robots we follow the approach which is described in detail in Xidias et al. (2007). Briefly, each n-robot is encircled with a circle of diameter d_n, where d_n is the distance between the "diagonal" vertices $\mathbf{a}_1^n$ and $\mathbf{a}_3^n$. If the distance $\boldsymbol{d}$ between the centres of two robots n and n' satisfies $\boldsymbol{d}\geq\frac{d_n}{2}+\frac{d_{n'}}{2}$, then no collision exist. Otherwise, collision detection requires searching for intersecting edges between the two robots.

The requirement for an n-path $\mathbf{R}^n(s)$ with lower bounded turning radius is given by,

$$\max(k_1^n(s),k_2^n(s),....,k_g^n(s))\leq k_{\max}^n,\quad n=1,...,\mathcal{N} \quad (6)$$

where $k_1^n(s),\ldots,k_g^n(s)$ are the roots of the equation $\frac{dk^n(s)}{ds}=0$, $k_{\max}^n$ is the maximum allowed curvature for the n-path and $k^n(s)$ is the curvature profile of the n-path Azariadis and Aspragathos (2005). The objective **D** is expressed by,

$$\dot{\mathbf{R}}^n(s)\succ 0, n=1,...,\mathcal{N} \quad (7)$$

where $\dot{\mathbf{R}}^n(s)$ is the first derivative of $\mathbf{R}^n(s)$ with respect to s. When the sign of $\dot{\mathbf{R}}^n(s)$ between consecutives roots ($\dot{\mathbf{R}}^n(s)=0$) is positive, then the curve $\mathbf{R}^n(s)$ has increasing monotony in $\mathcal{W}$.

3.2 Velocity Function

An optimum velocity profile must be generated for each robot to travel along an assigned path $\mathbf{R}^n(s)$, whose length is $\boldsymbol{L}^n(s)$. Since our main constraint is planning forward motions only, the velocity $V^n(s)$ is constraint by the relation (2).

Velocity can never become negative and can be equal to zero only at the starting point $\left(\mathbf{ST}^n\right)$ and ending point $\left(\mathbf{GL}^n\right)$. The velocity function $V^n(s)$ is defined by,

$$V^n(s)=\begin{cases} V_{\max}^n, if\ k^n(s)=0 \\ \min\left(V_{\max},\sqrt{\dfrac{T_0}{k(s)}}\right), if\ k^n(s)\neq 0 \end{cases},n=1,...,\mathcal{N} \quad (8)$$

It must be noticed that, in order to decrease the problem's complexity we assume that, the tangential acceleration takes the values $a=\{-a_{\max},0,a_{\max}\}$. Takes the value $-a_{\max}$ when the velocity at point $\mathbf{R}_j^n(s)$ is higher than the velocity at point $\mathbf{R}_{j+1}^n(s)$, the value 0 when both velocities are equal and the value $a_{\max}$ when the velocity at point $\mathbf{R}_j^n(s)$ is smaller than the velocity at point $\mathbf{R}_{j+1}^n(s)$.

3.3 Time function

Let $\mathbf{R}_j^n(s)$ and $\mathbf{R}_{j+1}^n(s)$ two points on the n-path $\mathbf{R}^n(s)$. We assume that the n-robot is moving from the point $\mathbf{R}_j^n(s)$ to the point $\mathbf{R}_{j+1}^n(s)$ in an infinitesimal time Δt, and $\Delta\boldsymbol{L}$ is the corresponding displacement along the curved path. The average velocity of the n-robot during this period is represented by $\mathbf{V}_j(s)$. The infinitesimal travel time Δt from the position $\mathbf{R}_j^n(s)$ to the position $\mathbf{R}_{j+1}^n(s)$ of the robot is given by,

$$\Delta t = \frac{\Delta \boldsymbol{L}(s)}{\mathbf{V}_j(s)} \quad (9)$$

Then, the time required for the n-robot to travel from its initial position $\left(\mathbf{ST}^n\right)$ to the target position $\left(\mathbf{GL}^n\right)$ along the path $\mathbf{R}^n(s)$ is calculated by,

$$\boldsymbol{t}^n = \sum_{j=0}^{N_{c-2}} \Delta t_j = \sum_{j=0}^{N_{c-2}} \frac{\left\| S(\mathbf{R}_{j+1}^n) - S(\mathbf{R}_j^n) \right\|}{\mathbf{V}_j(s)} \quad (10)$$

where N_{c-1} denotes the number of equally-sized sequential chords which approximate the n-path $\mathbf{R}^n(s)$, and $S(\mathbf{R}_j^n)$ is the image of each point $\mathbf{R}_j^n(s)$ on the Bump-Surface *S*.

The minimization of the Eq.(10) with respect to the control points $\mathbf{p}_i^n \in [0,1]^3$ leads to a time optimum n-path for the n-robot.

3.4 Objective Function

Taking the above analysis into consideration, the overall multi-objective optimization problem is given by,

$$E_{obj} = \mathcal{A}_1 \max\left(t^1, \ldots, t^{\mathcal{N}}\right) + \mathcal{A}_2 \sum_{n=1}^{\mathcal{N}} \left(\sum_{f=1}^{4} H_f^n \right) \quad \text{subject to}$$

$$\begin{cases} \max(k_1^n(s), k_2^n(s), \ldots, k_g^n(s)) \le k_{\max}^n \\ \dot{\mathbf{R}}^n(s) \succ 0, & , n = 1, \ldots, \mathcal{N} \\ a = \{-a_{\max}, 0, a_{\max}\} \end{cases} \quad (11)$$

The minimization of the Eq. (11) with respect to the control points $\mathbf{p}_i^n \in [0,1]^3$ and with the maximum possible instantaneous velocity leads to optimum collision free $\mathcal{N}$ paths for the set of robots, which satisfies the conditions **A-D**. The scalars $\mathcal{A}_1 + \mathcal{A}_2 = 1$, $\mathcal{A}_1, \mathcal{A}_2 \in [0,1]$ are weight factors, which means that if for example we want to search for shorter time paths ignoring to some degree the collisions we must put $\mathcal{A}_1 \succ \mathcal{A}_2$. It must be noticed that the total travel time is depended by the slowest n-robot.

4. SEARCHING FOR OPTIMAL PATHS

Taking into account that Eq.(11) corresponds to a nonlinear optimization problem with nonlinear constraints and high complexity, the Genetic Algorithms (GAs) are adopted in this paper. The 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 non-linear. In addition, GAs are well known for their ability of reaching a near-optimal solution for constrained optimization problems.

The proposed GA utilises a uniform crossover, a Gaussian mutation operator and a Roulette wheel selection. Finally, the following fitness function is used in order to minimize E_{obj}:

$$F = \begin{cases} \dfrac{1}{E_{obj}}, if \max(k_1^n(s), k_2^n(s), \ldots, k_g^n(s)) \le k_{\max}^n \\ \quad \dot{\mathbf{R}}^n(s) \succ 0, \\ \quad a = \{-a_{\max}, 0, a_{\max}\}, \quad n = 1, \ldots, \mathcal{N} \\ 0, otherwise \end{cases} \quad (12)$$

Each chromosome represents $\mathcal{N}$ possible paths for the set of robots as a sequence of control points, which define the B-Spline curves of Eq.(5). The coordinates of each control point $\mathbf{p}_i^n \in [0,1]^3$ are the genes of the chromosome. The initial population of the proposed GA consists of a number of chromosomes having genes with coordinates randomly selected within $[0,1]^3$.

5. EXPERIMENTAL RESULTS AND DISCUSSION

The performance of the proposed method is investigated through multiple simulation experiments for a set of car-like robots moving in 2D environments. Due to the space limitations we present and discuss only one example. The simulation is implemented in Matlab and on a Core 2 Duo 2.13 GHz PC.

The grid size *M* is 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=250, crossover rate=0.75 and boundary mutation rate=0.004. Furthermore, in all the experiments a (2,2,2)-degree B-Spline surface is used to represent the Bump-Surface.

Test Case: Figure 3 shows two car-like robots with steering angle 80° moving in a 2D static environment cluttered with four polygonal obstacles. The robots have the same value of maximum velocities and are set equal to 2. Figure 3 shows the solution paths $\mathbf{R}^1(s)$ and $\mathbf{R}^2(s)$. The velocity profiles of the robots are shown in Figure 4, where the red curve represents the velocity profile for robot-1 and the blue curve represents the velocity profile for robot-2.

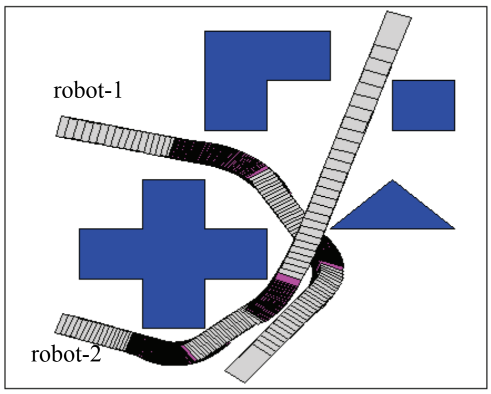

Fig. 3 The overall robots' motions.

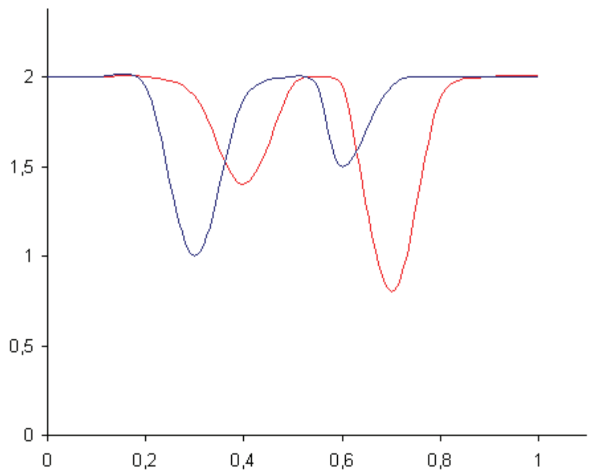

Fig. 4 The maximum allowable velocities for the robots.

In the above example, the robots are requested to travel a long distance avoiding each other and the obstacles. Due to the local-control property of the B-Spline curves, the robots are moving close to the obstacles and to each other with safety. The computed paths are smooth and satisfy optimally all the motion planning criteria. It must be noticed that due to the probabilistic nature of GAs, it is not possible to guarantee optimality even when it may be reached. However, they are likely to be close to the global optimum. This probabilistic nature of the solution is also the reason that they are not contained by local optima.

6. CONCLUSIONS

In this paper, a new method is presented for time-optimum motion planning for a set of car-like robots. The proposed method generates simultaneously time-optimum collision-free paths for a set of wheeled robots, which are moving in 2D environments cluttered with both static and moving obstacles. Its efficiency and reliability has been proven by means of a series of test experiments. The final paths are always smooth and satisfy the objectives of the underlying motion-planning problem. Considering the future research work, the proposed method could be applied to 2D environments using mobile robots with full manoeuvring capabilities (back and forth motions). It could also be extended to environments cluttered with pickup and delivery stations where a set of mobile robots is requested to serve these stations in optimum time.

ACKNOWLEDGMENT

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

REFERENCES

Alami, R., Robert, F., Ingrand, F., Suzuki, S., (1995). Multi-robot cooperation through incremental plan-merging. *In Proc. IEEE Int. Conf. on Robotics and Automation*, 2573-2678.

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

Bennewitz, M., Burgard, W., Thrun, S., (2001). Optimizing schedules for prioritized path planning of multi-robot systems. *In Proceedings of the Int. Conf. on Robotics and Automation*, 271- 276.

Dubins, L.E., (1957). On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents. *Amer. J. Math.*, 497-516.

Egerstedt, M., Hu, X., (2001). Formation constrained multi-agent control. *In Proceedings of the IEEE Conference on Robotics and Automation*, 3961–3966.

Fleury, S., Soueres P., Laumond, J.P., Chatila, R., (1995). Primitives for smoothing mobile robot trajectories. *IEEE Trans. of Robotics and Automation*, 441-448.

Jolly, K.G., Kumar, R.S., Vijayakumar, R., (2009). A Bezier curve based path planning in a multi agent robot soccer system without violating the acceleration limits. *On Robotics and Autonomous Systems*, Volume (57), 23-33.

Latombe, J. C., (1991). Robot Motion Planning, Kluwer Academic Publishers, Boston.

LaValle, M.S., (2004). Planning Algorithms. *University of Illinois*.

LaValle, M.S., Kuffner, J.J., (2001). Randomized kinodynamic planning. *International Journal of Robotics Research*, Volume (20), 278-400.

Lepetic, M., Klancar, G., Skrjanc, I., Matko, D., Potocnik, B., (2003). Time optimal path planning considering acceleration limits. *International Journal of Robotics and Autonomous Systems*, Volume (45), 99-110.

Podsedkowski, L., Nowakowski, J., Idzikowski, M., Vizvary, I., (2001). A new solution for path planning in partially known or unknown environment for non holonomic mobile robots. *Robotics and Autonomous Systems*, Volume (34), 145–152.

Qu, Z., Wang, J., Plaisted, C.E., (2004). A new solution to mobile robot trajectory generation in the presence of moving obstacles. *IEEE Transactions on Robotics*, Volume (20), 978-993.

Reeds, J.A., Shepp, L.A., (1990). Optimal paths for a car that goes both forwards and backwards. *Pacific J. Math.*, 367-393.

Ting, Y., Lei, W.I., Jar, H.C., (2002). A path-planning algorithm for industrial robots. *Computers and Industrial Engineering*, Volume (42), 299–308.

Xidias, E.K., Aspragathos, N.A., (2007). Motion Planning for Multiple Nonholonomic Robots Using the Bump-Surface Concept. *Robotica Journal,* Volume (26), 525-536.