Geodesic motion planning on 3D-terrains satisfying the robot's kinodynamic constraints

Arvanitakis, I., Tzes, A. & Thanou, M.

Original citation & hyperlink:
Arvanitakis, I, Tzes, A & Thanou, M 2013, Geodesic motion planning on 3D-terrains satisfying the robot's kinodynamic constraints. in 39th Annual Conference of the IEEE Industrial Electronics Society. IEEE, pp. 4144-4149, 39th Annual Conference of the IEEE Industrial Electronics Society, Vienna, Austria, 10/11/13.
https://dx.doi.org/10.1109/IECON.2013.6699800

DOI 10.1109/IECON.2013.6699800
ISBN 978-1-4799-0224-8

Publisher: IEEE

Geodesic motion planning on 3D-terrains satisfying the robot's kinodynamic constraints

Ioannis Arvanitakis
Department of Electrical and Computer Engineering
University of Patras
Rio, Achaia–26500, Greece
Email: arvanitakis@ece.upatras.gr

Anthony Tzes
Department of Electrical and Computer Engineering
University of Patras
Rio, Achaia–26500, Greece
Email: tzes@ece.upatras.gr

and Michalis Thanou
Department of Electrical and Computer Engineering
University of Patras
Rio, Achaia–26500, Greece
Email: mthanou@ece.upatras.gr

***Abstract*—In this article, a robot motion planning scheme for 3D-terrains is developed. Given the terrain profile and various obstacles on it, a navigation function is created. A geodesic based shortest path algorithm is developed to find the optimal lengthwise path towards the goal position. The path is then converted into a continuous smooth trajectory via an optimization scheme relying on a Bézier curve parametrization that satisfies the robot's kinodynamic constraints. The efficacy of the proposed method is tested in various simulation studies.**

I. Introduction

Area surveillance and patrolling, exploration of unknown and/or hazardous environments, search and rescue missions, are applications of mobile robots that demand a motion planning [1] algorithm. The area of motion planning has been extensively researched over the recent decades and extensive results have appeared [2]. Khatib in [3] proposed the idea of Artificial Potential Fields (APFs) created from the environment's obstacles and the target area that steer the robot towards a desired position; within the APF-theory the notion of a steering function for guiding a mobile robot is encountered. Rimon and Koditschek [4] focus on the navigation function and Borenstein and Koren [5], propose the use of vector field histogram which steers the robot towards the direction of low obstacle density areas. Kavraki et al. [6] propose the creation of a graph whose nodes are collision free configurations and its edges are paths between these configurations followed by the use of a query method to find a feasible path between the start and goal node. Kuffner and LaValle [7] propose the creation of two Rapidly-exploring Random Trees (RRT) both at start and goal position, which explore free space and advance towards each other with the help of a greedy heuristic.

Most of the encountered motion planning works have been implemented for navigation on planar (2D) terrains. To account for the robot's velocity and acceleration constraints the notion of Kinodynamic Motion Planning [8] is introduced.

Recently, there has been an effort to extend these theories for 3D-terrains to account for other issues such as slippage, maximum inclination that ought to be considered. Page et al. in [9] propose the use of Artificial Potential Fields method on 3D-terrains to avoid: a) ridges in the covert path case, and/or b) valleys in the surveillance path case. Iagnema et al. [10] propose a layered control strategy with a high level layer creating the desired trajectory represented via waypoints and a lower level using a potential-field based control method to track it. Eathakota et al. [11] use a variable of RRT's to produce a slipping free trajectory for a wheeled mobile robot. Waheed and Fotouhi [12] use a path planning method based on cubic splines with a trajectory generation method to calculate the velocity profile for each spline segment for indoor environments. Singh et al. in [13] use a two-stepped control strategy to navigate a wheeled robot in a free configuration space, where they create a trajectory by ignoring the vehicle dynamics, and then employ a non-linear scaling transformation to satisfy stability constraints such as no-slip and permanent ground contact. Tahirovich and Magnani in [14] propose a passivity-based non-linear model predictive controller as an extension of the Dynamic Window Approach for navigation on rough terrains. Miro et al. [15] propose a fast marching algorithm to utilise with a kinodynamic metric to calculate a stable path. In most of these cases, a distinguished vehicle design is utilised, resulting into complicated motion planning algorithms.

The main novelty of the present case study is a layered method to solve the motion planning problem of a differential drive robot in 3D-terrains. Initially, a simple geodesic based navigation function is utilised that solves the path planning problem and creates a collision free path. Then a trajectory extraction method based on Bézier curves [16] generates the trajectory (from the path), ensuring that the resulting (angular/linear) velocities are well within the velocity feasible area of the robot, while remaining close to the original path.

The paper is structured as follows: In Section II the path planning algorithm is presented followed by the Bézier-curve based trajectory generation method in Section III. In Section IV multiple simulation results that prove the efficacy of the proposed scheme are offered, while in Section V, conclusions are drawn.

II. Geodesic-Metric Path Planning

A. Mathematical Preliminaries

Consider a smooth Riemannian 2D-manifold in $\Omega \subset \Re^3$, described by:

$$\mathcal{H}(x, y, z) = 0 \ . \tag{1}$$

Rather than using the Euclidean metric to define the distance between two points on $\mathcal{H}$, the more appropriate Geodesic metric, $d_g(X_1, X_2)$ can be utilized, defined as the infimum of the

length L_g taken over all continuous, piecewise continuously differentiable curves $\gamma(s) : [a,b] \rightarrow \Omega$ such that $\gamma(a) = X_1$ and $\gamma(b) = X_2$. The length L_g in this case can be defined as:

$$L_g(\gamma) = \int_a^b \sqrt{g_\gamma(\dot{\gamma}(t), \dot{\gamma}(t))} dt \ , \tag{2}$$

where g_γ is a Riemannian metric. A Riemannian metric g_γ is a differentiable map $\gamma \mapsto g_\gamma : T_\gamma\Omega \times T_\gamma\Omega \rightarrow \Re$, where $T_\gamma\Omega$ is Ω's tangential space, such that g_γ is: 1) Bilinear: $g_\gamma(\alpha Y_1 + \beta Y_2, Y_3) = \alpha g_\gamma(Y_1, Y_3) + \beta g_\gamma(Y_2, Y_3)$, 2) Symmetric, $g_\gamma(Y_1, Y_2) = g_\gamma(Y_2, Y_1)$, and 3) Positive Definite, $g_\gamma(Y, Y) > 0$ for $Y \neq 0$.

A manifold is considered compact in a topological sense; in our case in a simplified interpretation let the projection of Ω on the xy-plane be $\text{proj}_{xy}(\Omega) \subset \Re^2$, then the projection of any minimum length geodesic curve on Ω between any two points $X_1, X_2 \in \Omega$ should belong to $\text{proj}_{xy}(\Omega) \subset \Re^2$. Two distinct cases of a compact and a non compact manifold appear in Figure 1.

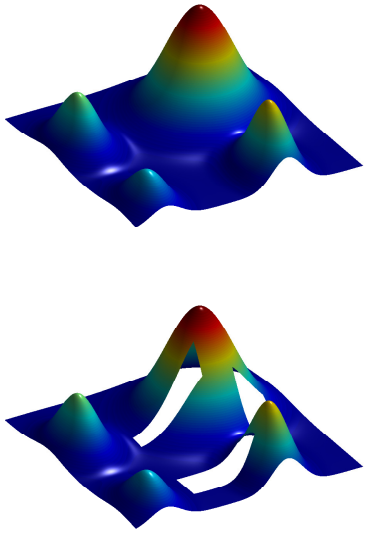

Fig. 1. Compact [top] and non-compact [bottom] manifolds

B. Geodesic Based Path Planning

A closed (with boundary) compact smooth Riemannian $2D$-manifold Ω is a typical configuration space of mobile robots that move on 3D-terrains. The robot's motion on this manifold can be occluded by: a) obstacles, or b) areas where the slope of the surface becomes too steep thus increasing the slippage and possible overturning. These areas are removed from the manifold thus possibly resulting in a non-convex non-compact manifold, $\Psi = \Omega \setminus \Sigma$, where Σ corresponds to the union of all restricted regimes.

Let the robot's position be $X = (x, y, z) \in \Psi$, and its target position P_T, then the robot's navigation function is:

$$\phi = d_g(X, P_T) \ . \tag{3}$$

To compute the desired path with the shortest length, the utilization of the deepest gradient descent method generates the *j*-th path point, X_j, as:

$$X_j = X_{j-1} - \alpha \frac{\nabla\phi}{\|\nabla\phi\|} \ , \ j = 1, \ldots, \tilde{N} \ , \tag{4}$$

where $0 < \alpha$ corresponds to a small step-size.

III. Kinodynamic Trajectory Generation

The path points, X_j, found from the geodesic path planning algorithm provide a discrete description of the path. A trajectory generation algorithm must be then utilised, that will meet the kinodynamic constraints imposed by the wheeled robot and that the resulting trajectory will be a smooth continuous curve.

A. Robot's Kinodynamic Model

A differential drive robot is assumed with its kinematic equations be defined as:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \upsilon \\ \omega \end{bmatrix} \ , \tag{5}$$

where $\upsilon(\omega)$ is the tangential(angular) velocity of the robot, $x(y)$ is the horizontal(vertical) displacement, and $\theta \in [-\pi, \ \pi]$ is the heading of the robot. In a differential drive robot, its motion is directly controlled by the right υ_R and left υ_L wheel velocities of the vehicle's two motor wheels, which affect the following expressions for υ and ω, as:

$$\upsilon = \frac{\upsilon_L + \upsilon_R}{2} \ , \tag{6}$$

$$\omega = \frac{\upsilon_R - \upsilon_L}{\lambda} \ , \tag{7}$$

where λ is the distance between the two wheels. The velocity constraints depend on the maximum allowable velocities of the two wheels. Given that $\upsilon_L(\upsilon_R) \in [-\upsilon_{\max}, \upsilon_{\max}]$, hard velocity bounds are:

$$\upsilon \in [-\upsilon_{\max}, \upsilon_{\max}] \ , \tag{8}$$

$$\omega \in [-\omega_{\max}, \omega_{\max}] = [-\frac{2\upsilon_{\max}}{\lambda}, \frac{2\upsilon_{\max}}{\lambda}] \ . \tag{9}$$

The feasible kinodynamic area $(\omega \times \upsilon)$ within which a differential–drive motor can operate is $(\omega \times \upsilon) = \pm\frac{2}{\lambda}(\upsilon \pm \upsilon_{\max}$; depending on the sign, the area's four boundary edges can be computed.

B. Bèzier Curve Based Trajectory Optimization

Based on the kinodynamic model of the robot (5-7) and its constraints, the trajectory generation method can be derived relying on Bèzier curves [17, 18]. Given $n+1$ $3D$-control points $(P_0, P_1 ... P_n)$, the n-th order Bèzier curve [19] is given from:

$$P(t) = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i} t^i P_i, \ t \in [0,1] \ . \tag{10}$$

If the curve must be traversed in time $\tau \in [\tau_0, \tau_f]$ then t is given from $t = \frac{\tau - \tau_0}{\tau_f - \tau_0}$. Bèzier curves have two distinct properties: 1) the starting and ending segment of the curve

is tangent to the first (P_0P_1) and last $(P_{n-1}P_n)$ edges, and 2) $P(t) \in$ convex hull$(P_0, P_1 ... P_n)$.

The created trajectory must be accurate relative to the initial path whilst smoothing out any sharp turns caused by the navigation function. For this reason, a segmented trajectory is created, consisting of sequential third order Bèzier curves, an extension in $3D$-space of the proposed method introduced in [20].

Given the path's points $X_i,\ i = 1, \ldots, N_i\ (N_i \leq \tilde{N})$, two control points A_i^{i+1}, B_i^{i+1} are introduced for each X_iX_{i+1} segment, as shown in Figure 2. These control points are selected as,

$$A_i^{i+1} = X_i + l_i^1 (X_{i+1} - X_i) \ , \quad (11)$$
$$B_i^{i+1} = X_{i+1} + l_i^2 (X_{i+1} - X_{i+2}) \ . \quad (12)$$

These four control points define a specific plane $\mathit{\Pi}_i$ in $3D$ space. Whilst this selection ensures that each path point is G^1 continuous (maintains first order geometric continuity), the resulting trajectory does not lay on the manifold. The final trajectory is created in a two-step process; 1) project the trajectory $\tilde{N}$-$P(t)$ segments on the xy-plane, and 2) project the 2D-trajectory back to the manifold.

Equation (10) for $n = 3$ can be rewritten:

$$P(t) = (1-t)^3 X_i + 3(1-t)^2 t(X_i + l_i^1(X_{i+1} - X_i)) + 3(1-t)t^2(X_{i+1} - l_i^2(X_{i+2} - X_{i+1})) + t^3 X_{i+1}. \quad (13)$$

To simplify the trajectory generation procedure, the normalised length $\ell_i = \frac{\|X_iA_i^{i+1}\|}{\|X_iX_{i+1}\|}$ is utilised, and the assumption that $\|X_iA_i^{i+1}\| = \|B_i^{i+1}X_{i+1}\|$, $(l_i^1 = l_i^2)$ is made. Also, the projection of the B_i^{i+1} control point should be at the semi–axis $A_i^{i+1}X_{i+1}$, or

$$\overrightarrow{A_i^{i+1}\text{proj}(B_i^{i+1})} \cdot \overrightarrow{A_i^{i+1}X_{i+1}} \geq 0 \ , \quad (14)$$

in order to avoid self–loops of the Bèzier.

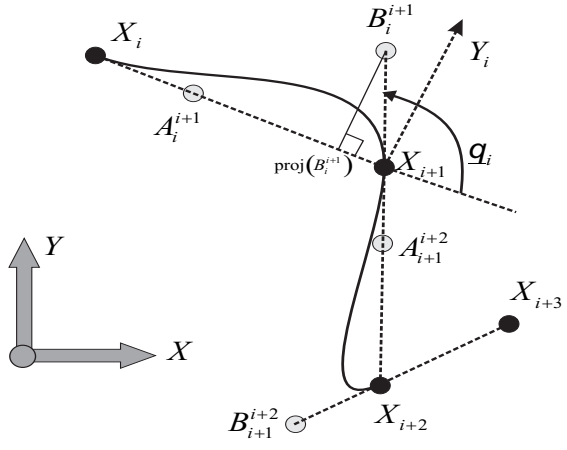

Fig. 2. Bèzier curve's control points on plane $\mathit{\Pi}_i$

From the resulting Bèzier, the $x(t)$ and $y(t)$ coordinates are taken into account (xy-plane projection) and the $z(t)$ coordinate is calculated then from equation (1) (projection of the 2D-trajectory back to the manifold).

The length of each dual-projected Bezier-based curve is given from,

$$L_i = \int_0^1 \sqrt{\dot{x}_i^2(t) + \dot{y}_i^2(t) + \dot{z}_i^2(t)}dt \ , \quad (15)$$

while its curvature is:

$$\kappa_i = \left\| \frac{d\overrightarrow{\mathcal{T}}_i}{ds} \right\| \ , \quad (16)$$

where $\overrightarrow{\mathcal{T}}$ is the unit tangent vector of the curve and s is the arc length parametrization.

Given the time-parametrization of the curve, (16) is:

$$\kappa_i = \left\| \frac{d\overrightarrow{\mathcal{T}}_i}{dt} \right\| \frac{1}{\|\overrightarrow{v}\|} \ , \text{ where } \overrightarrow{v} = \frac{d\overrightarrow{r_i}}{dt} \ . \quad (17)$$

The unit tangent vector can be calculated as:

$$\overrightarrow{\mathcal{T}}_i = \frac{\overrightarrow{v}}{\|\overrightarrow{v}\|} \ , \quad (18)$$

and its time derivative:

$$\frac{d\overrightarrow{\mathcal{T}}_i}{dt} = \frac{\frac{d\overrightarrow{v}}{dt}\|\overrightarrow{v}\| - \overrightarrow{v}\frac{d\|\overrightarrow{v}\|}{dt}}{\|\overrightarrow{v}\|^2} \ , \text{ where} \quad (19)$$
$$\frac{d\|\overrightarrow{v}\|}{dt} = \frac{\overrightarrow{v} \cdot \frac{d\overrightarrow{v}}{dt}}{\|\overrightarrow{v}\|} \ .$$

Equations (17) and (19) provide the the curve's curvature in $3D$-space. This curvature κ_i of the 3D-curve lying on the 2D-manifold can be further split into

$$\kappa_i = \sqrt{(\kappa_i^g)^2 + (\kappa_i^n)^2} \ , \text{ or} \quad (20)$$

1) the normal curvature κ^n which corresponds to the curvature's segment due to the manifold

$$\kappa_i^n = \frac{\nabla\mathcal{H}}{\|\nabla\mathcal{H}\|} \cdot \frac{d\overrightarrow{\mathcal{T}}_i}{dt} \quad (21)$$

and 2) the geodesic curvature κ^g which is the projection on the tangent plane.

From equations (20) and (21), the geodesic curvature can be calculated, parametrized by parameter ℓ_i. The projection of a geodesic curve onto the tangent plane is a straight line with zero geodesic curvature. Given the robot's velocity constraints, a kinodynamic planning must be used for the selection of parameter ℓ_i and the time T_i needed to traverse each curve.

In general, the requirements for the resulting trajectory are: 1) force the robot in traversing each segment with constant translational speed, 2) smoothing out any sharp turns while staying as close as possible to a geodesic curve. These requirements amount to: a) $0 \leq \upsilon = \frac{L_i}{T_i} \leq \upsilon_{\max}$, and b) decrease the geodesic curvature of the path. However the curvature along a segment varies and the worst case corresponds to a min-max problem, where the objective is to compute a path that minimizes the maximum value of the geodesic curvature. This

amounts to the following optimization problem for the ℓ_i - selection

$$\ell_i = \arg\min\{\max|\kappa_i^g(t;\ell_i)|\} \tag{22}$$

$$\text{subject to } 0 \le \ell_i \le \frac{1}{1+\cos\theta_i} \text{ , where} \tag{23}$$

$$\theta_i = \frac{\overrightarrow{X_iX_{i+1}}}{\|X_iX_{i+1}\|} \cdot \frac{\overrightarrow{X_{i+1}B_i^{i+1}}}{\|X_{i+1}B_i^{i+1}\|} \ . \tag{24}$$

The constraint for this non–linear optimization problem is posed in order to avoid self–loop in the Bèzier curve.

The angular velocity of the robot is the rate of rotation of the heading of the robot around the normal vector of the surface it moves on. Taking into account that its motion lies on the tangent plane of this surface, the part of the curvature that relates to this rotation is the geodesic curvature (since it is the projection of the curvature on the tangent plane), and the angular velocity can be calculated as:

$$\omega_i(t) = \upsilon_i\kappa_i^g(t) \ . \tag{25}$$

Hence the velocity constraints of the robot

$$\begin{aligned} \max_t|\omega_i(t)| &= \upsilon_i \max_t|\kappa_i^g(t)| \\ &= \frac{L_i}{T_i}\max_t|\kappa_i(t)| \le -\frac{2}{\lambda}\left(\frac{L_i}{T_i}-\upsilon_{\max}\right), \end{aligned} \tag{26}$$

need to be accounted for the derivation of the lower bound on T_i.

Let the path's length from (4) be L_p and assume N_i control points for the Bèzier curves. If these control points are selected in an equidistant manner along the path then $L_p = N_iL_{pi}$. Large N_i–values lead to many segments, thus increasing the complexity of the transformation from a path to a trajectory satisfying the kinodynamic constraints. On the contrary, small N_i lead to an inaccuracy of the resulting trajectory relative to the initial path. Given that the initial path is a geodesic one, a good measure of its approximation is from the resulting curvature. In this work the normalized average curvature in the jth segment is considered as a penalty term, as

$$\xi_j = \frac{\sqrt{\int_{t_0}^{t_f}\left(\kappa_j^g(\tau)\right)^2 d\tau}}{\max_t|\kappa_j^g(t)|} \ . \tag{27}$$

The overall cost is formulated as

$$J = w_1N_i + w_2\sum_{j=1}^{N_i}\xi_j \ , \tag{28}$$

where w_i , $i = 1, 2$ are positive weights. The selection of the number of the path points, N_i, is related to this cost function, with contradicting terms, since large N_i leads to small ξ_j. It should be noted that geodesic paths are often comprised of line segments near the boundaries of the manifold $\partial\Omega$. For this reason a lower bound on N_i should exist.

IV. Simulation Studies

In this section simulation results which prove the efficacy of the proposed algorithm are presented. The utilised robot has velocity constraints $\upsilon_{\max} = 0.5$ m/sec and $\omega_{\max} = 3$ rad/sec. The area under investigation is the non compact case of Figure 1, which corresponds to four Gaussian distributions, or

$$\mathcal{H}(x,y,z) = z - \sum_{i=1}^{4} A_i e^{-\frac{(x-\overline{x_i})^2}{2\sigma_{x_i}^2}-\frac{(y-\overline{y_i})^2}{2\sigma_{y_i}^2}} \ ,$$

where $A_i = \{2,4,6,3\}$, $\overline{x_i} = \{2,10,10,2\}$, $\overline{y_i} = \{2,2,10,10\}$, $\sigma_{x_i} = \{1,1,2,1\}$, $\sigma_{y_i} = \{1.5,1.5,2,1.5\}$, for $i = 1,\ldots,4$. The 'prohibited' (obstacle) area is marked by three obstacles bounded by their xy-corners located at Obstacle 1: [(5,2);(9,2);(9,4);(5,4)], Obstacle 2: [(5,8);(9,8);(9,9);(5,9)], Obstacle 3: [(11,5);(13,5);(13,8);(11,8)]. This manifold is constrained in a xy-square of 15×15; the target position is $P_T = (10, 12, 3.64)$, while the robot is initially at $X = (9.8, 1, 3.14)$.

In the absence of the obstacles, the geodesic path appears in Figure 3 with a length of $L_g = 14.2m$.

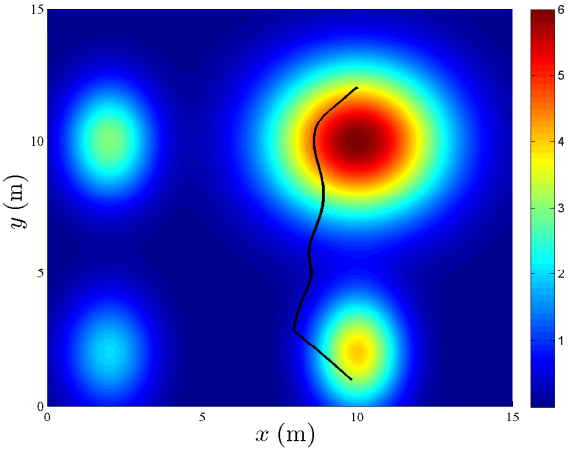

Fig. 3. Obstacle-Free Geodesic Path

In the presence of the obstacles, the generated 'constrained' geodesic path appears in Figure 4 with an $L_g = 14.6m$ length.

It is evident that the proposed gradient algorithm altered the robot's path to avoid collision with the obstacles, at the cost of a longer path. If the elevation of the area was ignored and a Euclidean distance based navigation function was utilised (a straight line between initial position and target) this would result in a path of larger length $L_g = 16.2m$. For the derivation of the trajectory from the 'constrained' geodesic path, in the first case $N_i = 54$ way points were selected. The resulting trajectory appears in Figure 5 and has a total length of $L_g = 15.7m$ and is traversed in $T_P = 40$ sec.

The reference velocity profile for each segment (and its convex hull for visualization purposes) is seen in Figure 6 where it is clear that the robot's velocity constraints are satisfied. The N_i waypoints are shown as red-dots on the

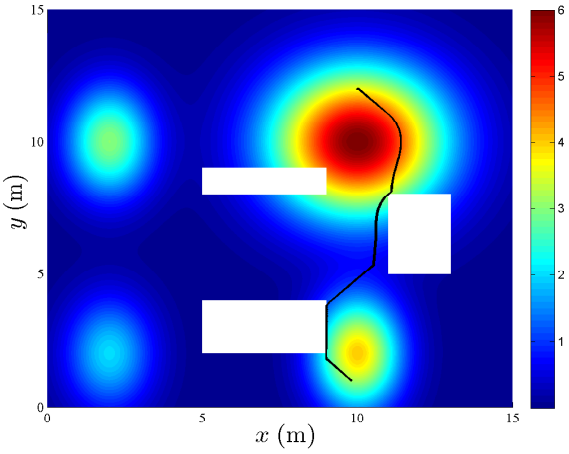

Fig. 4. Constrained Geodesic Path

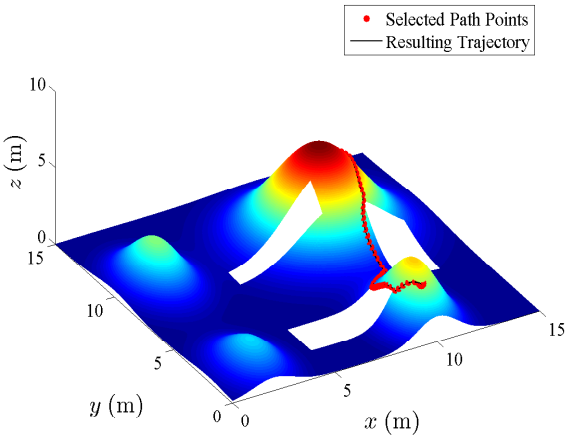

Fig. 5. Kinodynamicaly Constrained Trajectory for $N_i = 54$

constrained geodesic path. The cost from equation (28) with the weights chosen as $w_1 = 0.5$ and $w_1 = 1$ is $J = 36.26$.

If $N_i = 14$ are selected, then the resulting trajectory appears in Figure 7, will be traversed in $T_P = 37.2\,\text{sec}$ and has a length of $L_g = 15.3m$. This selection of N_i results to a geodesically longer but smoother trajectory, with higher translational and lower angular reference velocities. The cost in this case is $J = 25.6$ and the reference velocities are shown in Figure 9.

It should be noted that the selection of a small $N_i = 14$ in this case results in a trajectory that has a part which briefly enters in the 'prohibited' zone (area of an obstacle, in this case the bottom obstacle, as seen in Figure 8) caused by the convex hull of a certain Bezier curve. This can be checked in an a posteriori manner and further emphasises the fact that a lower bound on selection of N_i should be imposed.

V. Conclusions

In this paper a motion planning algorithm for 3D-terrains was presented. Firstly a navigation function based on the geodesic distance for path creation was analysed. The path is transformed into a continuous trajectory via a Bèzier curve

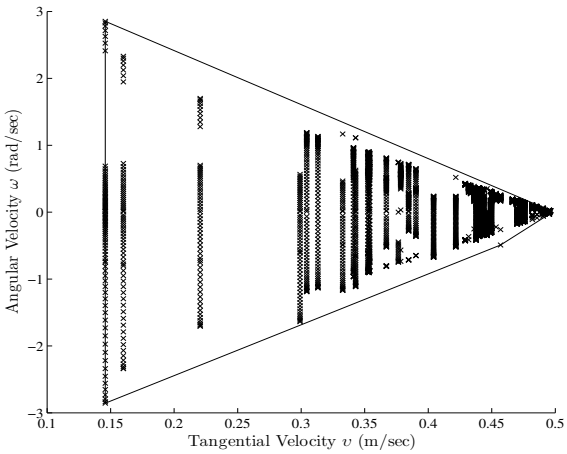

Fig. 6. Reference $(v \times \omega)$ profiles for $N_i = 54$ segments

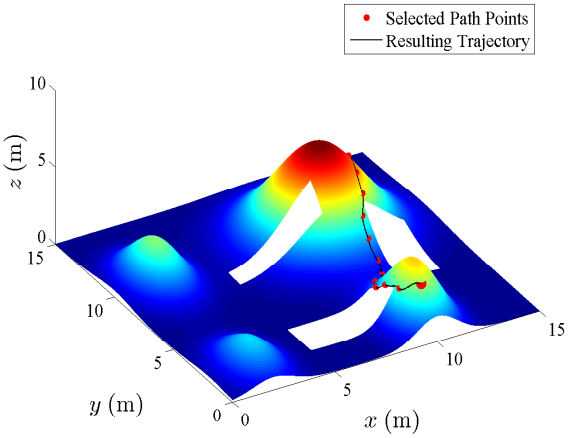

Fig. 7. Kinodynamicaly Constrained Trajectory for $N_i = 14$

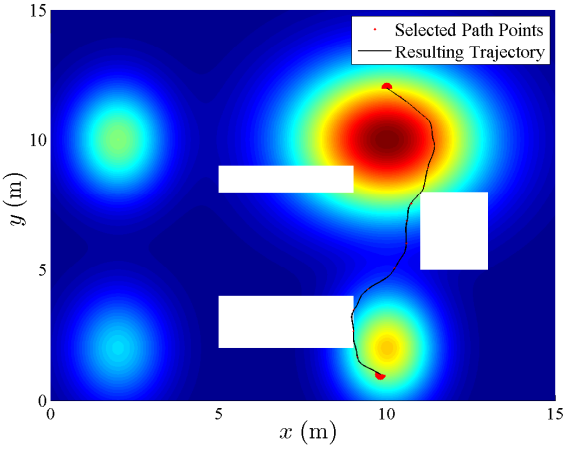

Fig. 8. Footprint of the Kinodynamicaly Constrained Trajectory for $N_i = 14$

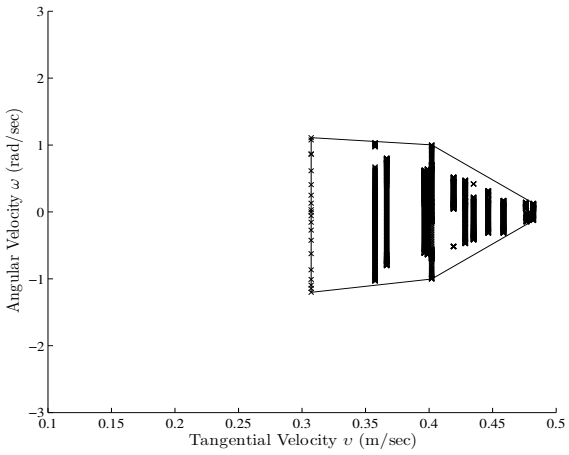

Fig. 9. Reference $(v \times \omega)$ profiles for $N_i = 14$ segments

based optimization algorithm, that satisfies the kinodynamic constraints of a differential drive robot. Simulation results prove the efficacy of the proposed algorithm.

REFERENCES

[1] E. Coste-Maniere and R. Simmons, "Architecture, the backbone of robotic systems," in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol. 1, 2000, pp. 67 –72 vol.1.

[2] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006.

[3] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, vol. 2, Mar 1985, pp. 500 – 505.

[4] E. Rimon and D. Koditschek, "Exact robot navigation using artificial potential functions," *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 5, pp. 501 –518, October 1992.

[5] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *Robotics and Automation, IEEE Transactions on*, vol. 7, no. 3, pp. 278 –288, June 1991.

[6] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 4, pp. 566 –580, Aug 1996.

[7] J. Kuffner, J.J. and S. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol. 2, 2000, pp. 995 –1001 vol.2.

[8] B. Donald, P. Xavier, J. Canny, and J. Reif, "Kinodynamic motion planning," *J. ACM*, vol. 40, no. 5, pp. 1048–1066, Nov. 1993.

[9] D. Page, A. Koschan, M. Abidi, and J. Overholt, "Ridge-valley path planning for 3d terrains," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, 2006, pp. 119–124.

[10] K. Iagnemma, S. Shimoda, and Z. Shiller, "Near-optimal navigation of high speed mobile robots on uneven terrain," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, September 2008, pp. 4098 –4103.

[11] V. Eathakota, G. Aditya, and M. Krishna, "Quasi-static motion planning on uneven terrain for a wheeled mobile robot," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, September 2011, pp. 4314 –4320.

[12] I. Waheed and R. Fotouhi, "Trajectory and temporal planning of a wheeled mobile robot on an uneven surface," *Robotica*, vol. 27, pp. 481–498, June 2009. [Online]. Available: http://journals.cambridge.org/article_S0263574708004876

[13] A. Singh, K. Krishna, and S. Saripalli, "Planning trajectories on uneven terrain using optimization and non-linear time scaling techniques," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 3538–3545.

[14] A. Tahirovic and G. Magnani, "General framework for mobile robot navigation using passivity-based mpc," *Automatic Control, IEEE Transactions on*, vol. 56, no. 1, pp. 184–190, 2011.

[15] J. Miro, G. Dumonteil, C. Beck, and G. Dissanayake, "A kyno-dynamic metric to plan stable paths over uneven terrain," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010, pp. 294–299.

[16] P. Marantos, Y. Koveos, J. Stergiopoulos, A. Panousopoulou, and A. Tzes, "Mobile robot odometry relying on data fusion from rf and ultrasound measurements in a wireless sensor framework," in *Control and Automation, 2008 16th Mediterranean Conference on*, 2008, pp. 523–528.

[17] T. Fraichard and A. Scheuer, "From reeds and shepp's to continuous-curvature paths," *Robotics, IEEE Transactions on*, vol. 20, no. 6, pp. 1025 – 1035, Dec. 2004.

[18] K. Yang and S. Sukkarieh, "An analytical continuous-curvature path-smoothing algorithm," *Robotics, IEEE Transactions on*, vol. 26, no. 3, pp. 561 –568, June 2010.

[19] G. Farin, *Curves and Surfaces for CAGD, Fifth Edition: A Practical Guide*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001.

[20] I. Arvanitakis and A. Tzes, "Trajectory optimization satisfying the robot's kinodynamic constraints for obstacle avoidance," in *Control Automation (MED), 2012 20th Mediterranean Conference on*, 2012, pp. 128–133.