

Robot navigation in multi-terrain outdoor environments

Guilherme A. S. Pereira¹, Luciano C. A. Pimenta², Luiz Chaimowicz³, Alexandre R. Fonseca², Daniel S. C. de Almeida³, Leonardo de Q. Corrêa², Renato C. Mesquita¹, and Mario F. M. Campos³

¹ Departamento de Engenharia Elétrica, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil.
gpereira@ufmg.br, renato@ufmg.br

² Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil.
lucpim@cpdee.ufmg.br, afonseca@cpdee.ufmg.br, leoqc@ufmg.br

³ Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil.
chaimo@dcc.ufmg.br, chaltein@dcc.ufmg.br, mario@dcc.ufmg.br

Abstract. This paper presents a methodology for motion planning in outdoor environments that takes into account specific characteristics of the terrain. Instead of decomposing the robot configuration space into “free” and “occupied”, we consider the existence of several regions with different navigation costs. Costs are determined experimentally by navigating the robot through the regions and measuring the influence of the terrain on its motion. In this paper we measure the robot vertical acceleration, which reflects the terrain roughness. A path planning algorithm is used to determine a sequence of triangles that minimize the navigation cost. Robot control is accomplished by a piecewise continuous vector field that drives the robot through those regions. This vector field allows the robot velocity to change according to the characteristics of the terrain. Experimental results with a differential driven, all terrain mobile robot illustrate the proposed approach.

1 Introduction

The motion planning problem for mobile robots can be loosely stated as the problem of finding and executing an obstacle free path from an initial position to a pre-specified goal. When dealing with indoor environments, examples of obstacles are walls, stairs, furniture, and other objects that can compromise robot safety and the overall task execution. Thus, obstacles are considered as forbidden regions in the robot’s configuration space. On the other hand, outdoor environments, which are sparsely occupied and composed by different regions, pose new challenges in robot navigation. In general, robots will have motion constraints related to their interaction with the environment. Robots will face different surfaces (grass, concrete, gravel), uneven terrains and other constraints, which are not necessarily obstacles to be avoided, but must be considered while planning the robot motion.

In this paper, we propose a methodology for motion planning in outdoor environments that takes into account these characteristics. Instead of decomposing the configuration space into “free” and “occupied”, we consider the existence of several regions with different navigation costs. Costs are determined experimentally by navigating the robot through these regions and measuring its performance. Robot control is accomplished by a piecewise continuous vector field that drives the robot through different discrete regions. An important characteristic of this vector field is the possibility of specifying a velocity profile for the robot. We use this characteristic to bound some effects of the terrain in the robot motion.

There are some works on the outdoor robot navigation problem, for example [9,6,7,13,10]. Among these, our work is closely related to [7], [13], and [10], but with important differences. Firstly, differently from [7], which computes cost values based on terrain slope and roughness, we propose a new cost metric obtained experimentally. Secondly, although we initially work with continuous maps as in [10], our path searching approach is based on discrete maps. However, instead of regular [7] or quadtree [13] decomposition, we decompose the environment using the Constrained Delaunay Triangulation (CDT) [12]. An important

advantage of CDT over quadtree representations is the smaller number of cells when representing complex structures, common in outdoors environments. Finally, differently from previous works on outdoor navigation, our controller is based on a piecewise continuous, terrain cost dependent vector field that leads the robot through different triangular regions. The construction of this vector field is based on the indoor approach presented in [2].

This paper is organized as follows: next section presents the theoretical details of our approach. Extensions to actual robots and environments are in section 3. Experimental results with a differential driven all terrain robot in a multi-terrain environment are in section 4. Finally, conclusions and future work are in section 5.

2 Methodology

2.1 Problem Statement

Consider a robot R operating in an outdoor environment. The environment is represented by a thematic map defined as $\mathcal{W} = \{(x, y, g(x, y)) | x_{min} \leq x \leq x_{max}; y_{min} \leq y \leq y_{max}\}$, where $0 \leq g(x, y) \leq +\infty$ is a function that represents a specific characteristic of the environment at robot position (x, y) , such as, obstacles, slope, terrain roughness, and others. Actually, $g(x, y)$ gives the cost of traversing a region per unit distance. At first, no map discretization is assumed and $(x, y) \in \mathbb{R}^2$. Higher values of $g(x, y)$ represent challenging regions for the robot or for the completion of its task. In the limit, $g(x, y) = +\infty$ is assigned to regions where the robot cannot move. Thus, we define the forbidden regions of the map, \mathcal{O} , as the regions where $g(x, y) = +\infty$ and the free regions of the map as $\mathcal{F} = \mathcal{W} \setminus \mathcal{O}$.

Our first goal is to find the path given by a curve $\mathcal{Y} \in \mathcal{F}$, which connects the robot initial position $\mathbf{q}_0 = (x_0, y_0)$ to the goal position $\mathbf{q}_g = (x_g, y_g)$, and minimizes the cost functional given by:

$$I = \int_{\mathcal{Y}} g(x, y) ds, \quad (1)$$

where ds is the differential of the arc length.

Second, we need a robot control law that besides guiding the robot to follow \mathcal{Y} with a bounded error also takes into account some characteristics of a common outdoor environment such as presence of dynamic obstacles, localization errors and wheel slippage.

Our solution for these problems includes: (i) dividing the continuous map into discrete regions, each one with a constant cost function $g(x, y)$; (ii) discretizing each region with a small number of triangles; (iii) finding a sequence of triangles that contains \mathcal{Y} ; (iv) computing a piecewise continuous vector field that drives the robot through the sequence of triangles; and (v) controlling the robot to follow the vector field. Each one of these five steps will be detailed next.

2.2 Path planning

A proper data structure to represent a thematic map is a *planar map* [5]. A planar map subdivides the plane into vertices, edges, and faces. Since a planar map assumes each face is a polygonal region, circular regions and arcs must be approximated by a set of line segments. In the case of large and sparsely occupied workspaces this is not an important constraint, as it will be seen in the Experiments section.

The way we represent the cost to traverse a given region in a specific thematic map is similar to [10]. The face, $f_i \subset \mathbb{R}^2$, represents a specific property in a thematic map and has assigned to it a traversing cost $\alpha_i \in [0, +\infty]$ per unit distance. Therefore, the cost function $g(x, y)$ presented in the previous section can be defined as:

$$g(x, y) = \alpha_i \quad \forall (x, y) \in f_i, i = 1, 2, \dots, m, \quad (2)$$

where m is the number of faces in the map \mathcal{W} .

The total cost $\beta(\mathbf{q}_1, \mathbf{q}_2)$ of a line segment that links two points, \mathbf{q}_1 and \mathbf{q}_2 , placed in face f_i is given by:

$$\beta(\mathbf{q}_1, \mathbf{q}_2) = \alpha_i |\mathbf{q}_1 \mathbf{q}_2|, \quad (3)$$

where $|\mathbf{q}_1 \mathbf{q}_2|$ is the Euclidean distance between them. However, Equation (3) is only valid for any point of f_i if this face is convex. In this paper we guarantee convexity by subdividing each non-convex face into a set of triangles. Each triangle is considered to be a new face of the map. Actually, to avoid an extra test for convexity and to maintain homogeneity, we discretize all faces of the map using the Constrained Delaunay Triangulation (CDT) [12]. CDT maintains conformity with the original boundaries of the faces.

In order to plan a path from the initial position $\mathbf{q}_0 \in f_0$ to the goal position $\mathbf{q}_g \in f_g$ we first create a graph $G(\mathcal{V}, \mathcal{E})$, where the set of graph nodes \mathcal{V} is composed by the midpoints of each map edge and points \mathbf{q}_0 and \mathbf{q}_g , and the segments linking the nodes at the same face constitute the edge set \mathcal{E} . The costs associated with the graph edges are computed using Equation (3).

Our continuous problem is now transformed into a discrete one stated as follows: find the path of minimum cost from \mathbf{q}_0 to \mathbf{q}_g in the graph $G(\mathcal{V}, \mathcal{E})$, given the edge costs β computed as described before. This path searching can be performed by well known algorithms such as A^* or Dijkstra. The later one guarantees the optimal path for the given graph. Notice that we do not find the optimal path for the continuous problem, but for the discrete one. Nonetheless, better approximations are obtained as the faces get smaller.

Although the previous methodology is able to compute a path from \mathbf{q}_0 to \mathbf{q}_g [4], our control approach, described in the next section, only relies on a sequence of adjacent triangles between f_0 and f_g . The chosen sequence contains the optimal graph path and is directly obtained from the graph search algorithm.

2.3 Vector Field for Robot Control

For robot control, a piecewise continuous vector field is constructed only inside the ‘‘corridor’’ formed by the sequence of triangular cells between the initial position and the goal. Basically, a set of vectors at the triangles vertices are interpolated to generate the vector field. This idea was introduced in [2].

Let $\mathcal{S} = \{f_0, f_1, f_2, \dots, f_n\}$ be an ordered sequence of n consecutive triangles, where $f_n = f_g$ (see Fig. 1(a)). The edges at the boundary of this sequence constitute the set \mathcal{B} . The common edges between two triangles form the set \mathcal{X} . We want to build a continuous vector field $\mathbf{u}(\mathbf{q})$ that drives a holonomic robot with kinematics $\dot{\mathbf{q}} = \mathbf{u}(\mathbf{q})$ from $\mathbf{q}_0 \in f_0$ to $\mathbf{q}_g \in f_n$, that fulfills the following requirements:

- (R1) for any time t , $\mathbf{q}(t) \in f_i$, and $f_i \in \mathcal{S}$,
- (R2) i is monotonically crescent.

Suppose that each vertex of the triangles in $\mathcal{S} \setminus f_n$ has an associated base vector, which satisfies the following constraints:

- (C1) its projection on the outward normal vector of an incident edge is negative if the edge is in \mathcal{B} , and
- (C2) its projection on the outward normal vector of an incident edge is positive if the edge is in \mathcal{X} .

Call \mathbf{v}_1 , \mathbf{v}_2 and \mathbf{v}_3 the base vectors at vertices (x_1, y_1) , (x_2, y_2) and (x_3, y_3) of f_i , ordered in a counter-clockwise way. A vector field, $\mathbf{u}(\mathbf{q})$, that simultaneously fulfills requirements (R1) and (R2) can be computed by the convex combination of these vectors given by:

$$\mathbf{u}(\mathbf{q}) = \psi_1 \mathbf{v}_1 + \psi_2 \mathbf{v}_2 + \psi_3 \mathbf{v}_3, \quad (4)$$

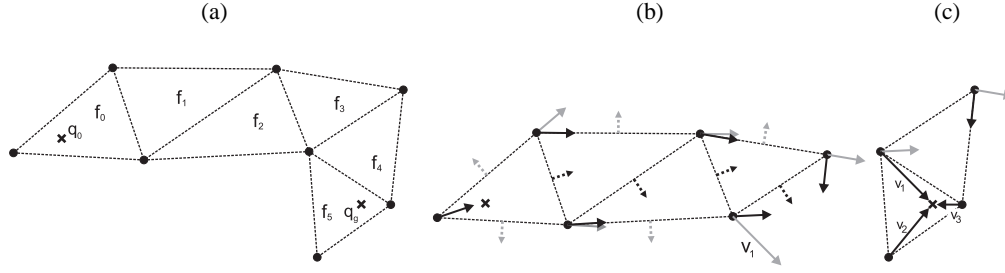


Fig. 1. Example of a typical sequence of triangles and the choice of the vector field base vectors. (a) Entire sequence composed by f_0, \dots, f_5 . (b) Dashed gray normal vectors are constraints associated to the boundary edges (edges in \mathcal{B}). These constraints are used to guarantee that the robot never leaves the “corridor” of triangles. Dashed black vectors are constraints associated to the output edges of each triangle (edges in \mathcal{X}). These constraints are used to make sure the robot will never move backwards. Since \mathbf{v}_1 and the other vector associated to the same vertex do not satisfy the constraints for all incident triangles (f_2, f_3, f_4 , and f_5), a discontinuity appears at this point of the sequence. Therefore we have two continuous sequences: (b) and (c). In both cases black and gray vectors indicate the vectors that respect and violate the constraints, respectively. In (c), the vectors in the last triangle, $\mathbf{v}_1, \mathbf{v}_2$, and \mathbf{v}_3 are used to stop the robot at the goal position.

where parameter ψ_l is maximum at vertex l and zero for all points on the opposite edge. Also $\psi_1 + \psi_2 + \psi_3 = 1$ and, as in [8]:

$$\psi_l(\mathbf{q}) = \begin{cases} \rho_l + \lambda_l x + \gamma_l y & \text{if } \mathbf{q} \in f_i \\ 0 & \text{if } \mathbf{q} \notin f_i \end{cases}, \quad (5)$$

where $l = 1, 2, 3$, $\mathbf{q} = (x, y)$, and ρ_l, λ_l and γ_l are suitable constants. For ψ_1 , these constants are:

$$\rho_1 = \frac{1}{2A}(x_2 y_3 - x_3 y_2), \quad \lambda_1 = \frac{1}{2A}(y_2 - y_3), \quad \gamma_1 = \frac{1}{2A}(x_3 - x_2), \quad (6)$$

where A is the area of the triangle. For ψ_2 and ψ_3 , the constants can be obtained by the cyclic permutation of the indices $(1, 2, 3)$. Once constraints (C1) and (C2) are satisfied, it is easy to prove that $\mathbf{u}(\mathbf{q})$ never drives the robot outside the corridor (requirement (R1)) and always moves the robot to the next triangle in the sequence (requirement (R2)). Observe that, since two adjacent triangles share two vertices (with the same associated vectors) at the common edge and the interpolation is linear along this edge, the interpolated vector field is continuous.

Differently from [2], where a linear programming technique is used to compute $\mathbf{v}_1, \mathbf{v}_2$ and \mathbf{v}_3 , we choose these vectors based on the geometry of the map and the cost function of the terrain. Although the algorithm in [2] yields a potentially better vector field (in the sense of shorter robot paths), our method is more intuitive and less expensive. We also have explicit control on the magnitude of the vectors, which we will show later to be very useful for navigation in multi-terrain environments. Besides these aspects, both methods are apparently interchangeable, since their results, drawbacks, and advantages seem to be similar. We will now proceed to explain our technique.

Except for f_0 and f_n , each vector \mathbf{v}_l of f_i is chosen to be parallel to one of the incident edges of vertex l that are in \mathcal{B} , pointing towards the direction determined by the sequence of triangles. Notice that we need to choose between two vectors (see Fig. 1(b)). The chosen one is the vector that simultaneously satisfies constraints (C1) and (C2). Except for collinear incident edges, just one vector may satisfy both constraints. When none of the vectors satisfies constraints (C1) and (C2) it can be proved that no other fixed

base vector can satisfy these constraints simultaneously [2]. This is inherent to the geometry of the problem and constitutes the main drawback of the methodology. However, a simple solution is to split the sequence of triangles into two subsequences, resulting in a discontinuity in the vector field. Therefore, the final vector field for the whole sequence of triangles is piecewise continuous.

For triangle f_0 , one of the vertices is not shared by other triangles (see Fig. 1(a)). The vector at this vertex can be any linear combination of the direction vectors of the incident edges. A good choice would be a vector that is orthogonal to the opposite edge. For f_n , since the robot must stop at \mathbf{q}_g , the vectors at each vertex l are computed as $\mathbf{q}_g - (x_l, y_l)$ (see Fig. 1(c)).

The procedure described so far is only able to chose the directions of \mathbf{v}_1 , \mathbf{v}_2 and \mathbf{v}_3 , but nothing has been said about their magnitudes. Since we are working in multi-terrain environments, the magnitudes of the vectors can be used to determine the robot velocity in different terrains. Thus, each triangle f_i (which by definition represents a single terrain) is associated with a maximum robot velocity. The magnitude of the vector \mathbf{v}_l is then defined as the minimum velocity associated to each of the incident triangles to vertex l . In this way, by Equation (4), we guarantee that the maximum robot velocity inside each triangle is smaller than or equal to the maximum velocity associated to each triangle. It is important to mention that we must maintain the proportion among the magnitudes of \mathbf{v}_1 , \mathbf{v}_2 and \mathbf{v}_3 of f_n , in order to guarantee that the robot will reach the goal.

3 Extension to Real Robots

The methodology, as explained thus far assumes a pointwise, holonomic robot represented by its exact configuration \mathbf{q} and working in a static environment. These assumptions do not always hold in real world. Therefore, it is natural to ask whether the methodology could be applied to control generic shaped, non-holonomic robots with estimated configuration $\hat{\mathbf{q}}$ in a dynamic environment.

Traditionally, the assumption of a pointwise robot is easily relaxed if obstacles are dilated by the size of the robot. We can extend this idea by dilating the higher cost faces of the map over the lower cost regions.

To take into account non-holonomic constraints, a feedback-linearization based controller can be used to make the robot follow the vector field [11]. Without loss of generality, in this section we consider a differential driven robot equipped with odometry, gyroscope and GPS (Global Positioning System). By defining the robot control point to be at distance d from its center of mass, we have:

$$\begin{bmatrix} V \\ \omega \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\frac{\sin \theta}{d} & \frac{\cos \theta}{d} \end{bmatrix} \mathbf{u}(\mathbf{q}), \quad (7)$$

where V and ω are the linear velocity of the robot's center of mass and its angular velocity respectively, θ is the robot's orientation, and $\mathbf{u}(\mathbf{q})$ is the control law for the holonomic robot given by Equation (4).

The estimation of \mathbf{q} is one of the most difficult problems for real robots. Several good solutions have already been proposed in literature for indoor localization, but outdoor localization is still a difficult problem. A proposed solution is to use recursive estimators, such as the Kalman Filter and its variations, to combine information from several different sensors. In this paper we propose the use of an Extended Kalman Filter to combine odometry and inertial information with GPS data.

First, the gyroscope angular velocity ω_g is combined with angular velocity ω_o given by odometry using a static Kalman Filter:

$$\hat{\omega} = \frac{\sigma_g^2}{\sigma_g^2 + \sigma_o^2} \omega_o + \frac{\sigma_o^2}{\sigma_g^2 + \sigma_o^2} \omega_g, \quad \sigma_\omega^2 = \frac{\sigma_g^2 \sigma_o^2}{\sigma_g^2 + \sigma_o^2}, \quad (8)$$

where $\hat{\omega}$ and σ_{ω}^2 are the estimated angular velocity and its covariance, respectively, σ_g^2 is the gyroscope's covariance and σ_o^2 is the odometry based angular velocity covariance.

On one hand, the estimated angular velocity $\hat{\omega}$ and the linear velocity \hat{V} based on odometry data are used as inputs to the robot prediction model. On the other hand, the absolute GPS data vector $\mathbf{X} = (x_{GPS}, y_{GPS}, \theta_{GPS})$ is used in the filter correction step. Practically, x_{GPS} and y_{GPS} are given by transforming (using a transformation matrix) measurements in a UTM (Universal Transverse Mercator) coordinate system to the map coordinate system. θ_{GPS} is given by the GPS course information. This information is only available if the GPS sensor displaces at a minimum linear velocity.

One advantage of most vector field approaches is that they are robust to small localization errors. Therefore, even if the Kalman Filter estimates have small drifts and errors, and as long as these errors do not indicate that the robot is outside the sequence of triangles, the robot will move in the correct direction. However, the approach proposed in the previous sections is very sensitive to the geometry of the map. One can observe that small triangles may lead to situations where the robot can find itself outside the corridor of triangles, due to localization errors, or even leave the corridor, due to actuator errors and dynamics. In order to avoid small triangles and to obtain good practical results, small obstacles and terrain details should not be included in the map. By terrain details we mean small variations in a specific kind of terrain that could be easily handled by an all-terrain robot.

Regarding dynamic and unmodeled obstacles in the environment, the vector field can be locally modified to allow obstacle avoidance. One approach is to find a suitable vector to be followed that has a positive projection on both the vector field and the vector normal to the obstacle, as proposed in [3]. Further attention should be paid when the robot is close to the boundaries of the sequence of triangles. In this case the resulting vector must also have negative projection on the outward normal vectors of the boundaries.

4 Experiments

Our experimental platform is a Pioneer 3 All Terrain mobile robot (P3AT) (Fig. 2). It is a four wheel, differential driven robot equipped with GPS, encoders, gyroscope, laser range scanner, among other sensors. A laptop was used on board and most of the programming was done using ARIA, the software framework provided with the robot. The robot maximum linear velocity in a flat surface is about 0.8 m/s.

The outdoor environment where experiments were conducted has about $2,500m^2$ of area with five distinct types of terrains: concrete, grass, gravel and two types of cobblestone. This environment is represented by the polygonal map shown in Fig. 2.

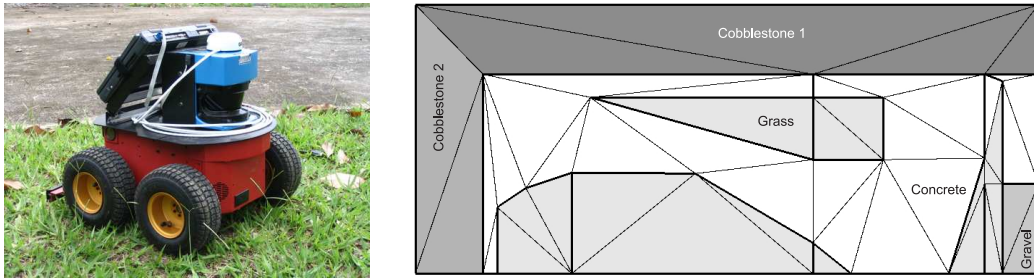


Fig. 2. Robot *Pioneer3 AT* used in the experiments and the map representing the workspace.

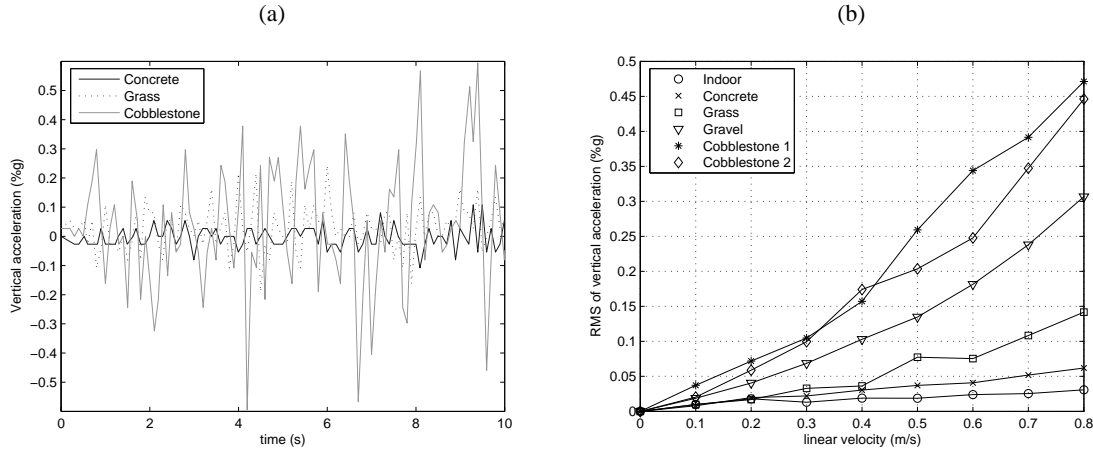


Fig. 3. (a) Vertical acceleration in different terrains; (b) Relationship between vertical acceleration and speed for five types of terrain.

4.1 Terrain Cost Estimation

There are several ways to determine navigation costs. Our approach consists in estimating the terrain roughness by measuring the robot vibration using an accelerometer mounted vertically. A good estimate of terrain roughness is obtained with the acceleration root mean square (RMS) value. This metric is used by the automobilistic industry to evaluate the effectiveness of car suspensions [1].

In our first experiment, the robot was programmed to traverse the five different types of terrain with several distinct velocities. Each trial resulted in a set of vertical accelerations such as the ones shown in Fig. 3(a), for a robot linear velocity of 0.5m/s and three types of terrain.

The RMS of the vertical accelerations for all terrains and velocities tested are depicted in Fig. 3(b). All values in this graph are given in percentage of $1g$ (about $9.8m/s^2$). It can be observed that the RMS value is much larger in rough terrains (e.g. gravel or cobblestone). Also, the vertical acceleration grows when velocity is increased. This graphic suggests, for example, that if it is important to limit the maximum vertical acceleration (e.g. for equipment safety), the robot velocities must be limited to a maximum value for each type of terrain. In other words, the robot may go faster in smooth terrains and slower in the rough ones. This approach is intuitive and is used by most automobile drivers. To determine costs for our graph, we choose the allowed vertical acceleration in each terrain, which consequently will determine the maximum robot velocity. This can be achieved by multiplying the inverse of the robot velocity in each face (α_i) by the graph edge length ($|q_1 q_2|$) yielding in the edge cost (β), which is a metric of time that should be minimized.

4.2 Path Planning and Robot Control

In this section we show how our methodology can be used to make the robot traverse different terrain regions towards a pre-specified final position. Based on the graphic of Fig. 3(b) we chose the maximum values for robot velocities. These values were used to determine the costs for each graph edge and also to limit the magnitude of the vector field base vectors.

Figure 4(a) shows two corridors composed by different terrains. In the first corridor we have only grass and concrete (see the map in Fig. 2). For this sequence of triangles, we aimed at keeping the RMS values

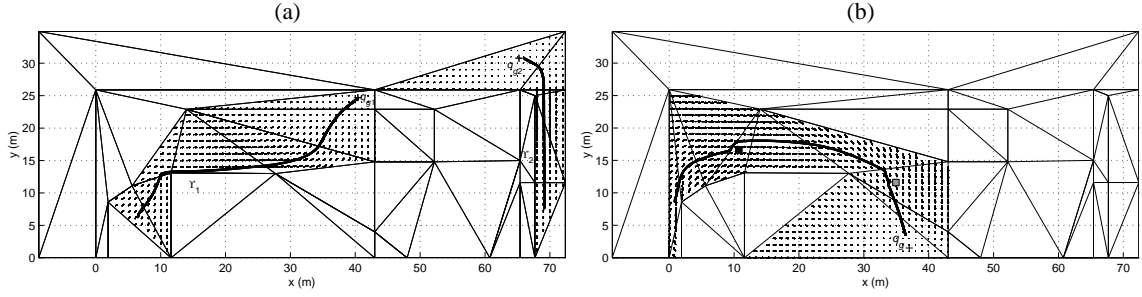


Fig. 4. (a) Two sequences of triangles and their respective vector fields. Two actual robot paths \mathcal{T}_1 and \mathcal{T}_2 are also overlaid on the map. (b) The small boxes represent dynamic (black box) and unmodelled (gray box) obstacles. The robot path shows that these obstacles were avoided in execution time by means of a laser based obstacle avoidance controller.

of vertical acceleration around 0.06% of g . The maximum velocity was limited to be $0.8m/s$ for concrete and $0.3m/s$ for grass. A typical robot path (\mathcal{T}_1) and the vector field for this sequence is also shown in Fig. 4(a). Robot velocities and vertical accelerations profiles for this specific path are presented in Fig. 5(a). The computed maximum acceleration RMS value for the path was 0.052% of g on concrete. Also, it is possible to see the effects of a discontinuity in the field at time 26s. The robot was subjected to a high angular velocity. Furthermore, observe that the robot linear velocity increases in the concrete region, but also gradually decreases before it reaches the next region (grass). This is due to the fact that both concrete and grass triangle share the same vectors, which have the magnitude determined by the maximum velocity on the grass.

For the second corridor in Fig. 4(a) we limited the robot velocity to $0.65m/s$ for concrete, $0.2m/s$ for gravel and $0.1m/s$ for cobblestone, in order to bound the acceleration RMS value to approximately 0.05% of g . Notice that the field is continuous for this sequence. Robot velocities and vertical accelerations profiles for a specific path (\mathcal{T}_2) in this corridor are presented in Fig. 5(b). Maximum RMS accelerations for this path were 0.060% of g on gravel and 0.088% of g on concrete. This apparently large value of vertical acceleration is due the presence of a small amount of gravel spread over the concrete region.

In Fig. 4(b) we show a third sequence of triangles and its respective vector field. In the experimental trial presented in this figure the robot was exposed to dynamic and unmodelled obstacles. The dynamic obstacle was a walking person that intercepted the robot path at the point marked with a black box in Fig. 4(b). The unmodelled obstacle was a small tree. This obstacle is represented by a gray box. As we mention before, since we chose not to have very small triangles in our sequence, we did not model the environment in all of its details. As we see in Fig. 4(b), the laser based obstacle avoidance controller was able to handle well with such details.

5 Conclusion and Research Perspectives

This paper presented a low cost, terrain based outdoor robot motion planning methodology. Low computational cost is mainly due to the discretization methodology used, that allows a good representation of the environment with a small number of cells. Observe in Fig. 2 that only 46 triangles were used to represent a $2,500m^2$ area. Also, although the resulting path is not necessarily optimal, it is a low cost path in the sense of distance and a terrain metric, which was vertical acceleration in our case.

We control the robot by computing a piecewise continuous vector field over a sequence of discrete regions. The vector field can be also generated very efficiently, since it does not need to be computed for regions far

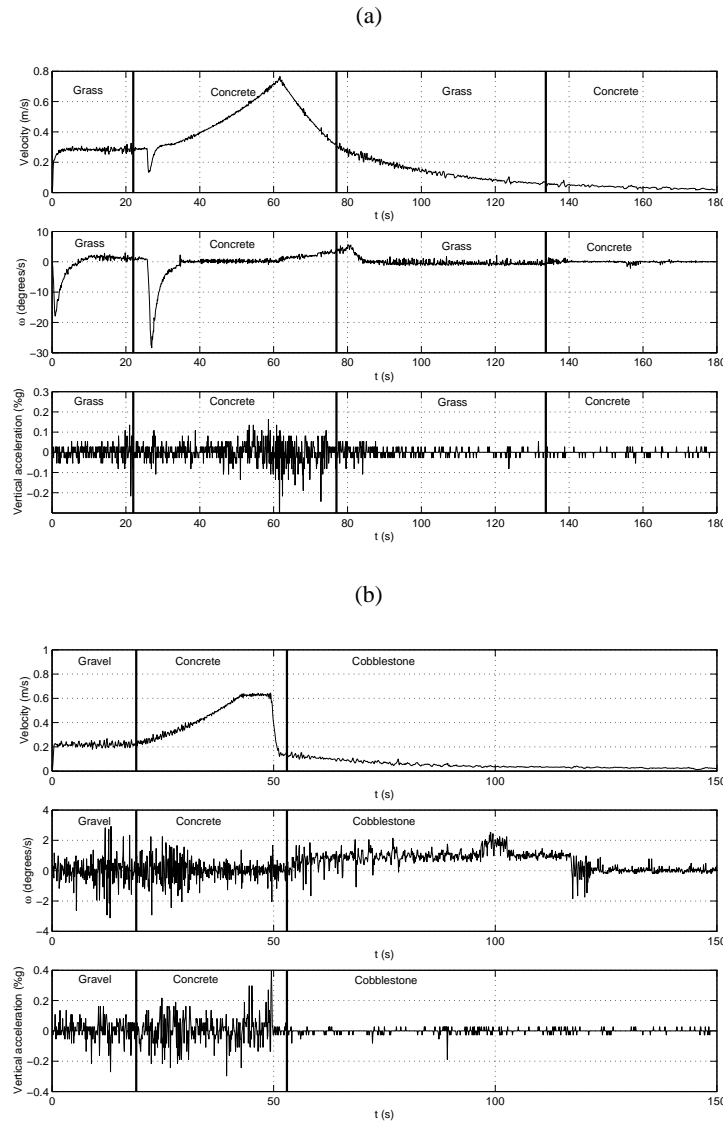


Fig. 5. (a) Velocity and vertical acceleration profiles for path γ_1 in Fig. 4(a). In this path, the robot traversed two types of terrain. (b) Velocity and vertical acceleration profiles for path γ_2 in Fig. 4(a). In this path, the robot traversed three types of terrain.

from the robot path. Also, it has the advantages of many potential field approaches, such as robustness to small localization errors and the possibility of avoiding unknown obstacles without replanning. This vector field also guarantees that (a) the robot will never leave the corridor determined by the sequence of triangles and, (b) at the same time, the robot will never move backwards.

The main limitations of the approach are: (a) it depends on a good map of the environment, but not too much detailed; (b) although we find and traverse the sequence of triangles that contains the optimal robot

path, the computed vector field does not guarantee this path will be followed; (c) the vector field inside the sequence of triangles may have discontinuities, which may lead to high angular velocity values; (d) since triangles in the sequence share base vectors, a low maximum velocity in one face limits the maximum velocity in its neighbors.

Based on these limitations, there are important directions for future research. The next steps that we plan to take include investigating other metrics for navigation cost and map building. An interesting metric is related to localization. Notice in Fig. 5 that the angular velocity variance depends on the terrain. It suggests that odometry based localization would result in different pose estimates for different terrains. As to field discontinuity and the influence between neighboring faces, we are currently investigating new interpolation techniques used by the electromagnetic field computation community.

Acknowledgement

The authors would like to thank the financial support of FAPEMIG – Fundação de Amparo à Pesquisa do Estado de Minas Gerais, and CNPq – Conselho Nacional de Desenvolvimento Científico e Tecnológico, Brazil.

References

1. Donald Bastow and Geoffrey P. Howard. *Car Suspension and Handling*. Pentech Press, London, 1993.
2. Calin Belta, Volkan Isler, and George J. Pappas. Discrete abstractions for robot motion planning and control in polygonal environments. *IEEE Transactions on Robotics*, 21(5):864–874, October 2005.
3. J. M. Esposito and V. Kumar. A method for modifying closed-loop motion plans to satisfy unpredictable dynamic constraints at runtime. In *IEEE International Conference on Robotics and Automation*, pages 1691–1696, 2002.
4. A. R. Fonseca, L. C. A. Pimenta, R. C. Mesquita, R. R. Saldanha, and G. A. S. Pereira. Path planning for mobile robots operating in outdoor environments using map overlay and triangular decomposition. In *Proceedings of the International Congress of Mechanical Engineering (COBEM'2005)*, Ouro Preto, Brazil, November 2005.
5. M. Gangnet, J.-C. Hervé, T. Pudet, and J.-M. van Thong. Incremental computation of planar maps. In *Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, pages 345–354, July 1989.
6. Jose Guivant, Eduardo Nebot, Juan Nieto, and Favio Masson. Navigation and mapping in large unstructured environments. *The International Journal of Robotics Research*, 23(4-5):449–472, April-May 2004.
7. Yi Guo, Lynne E. Parker, David Jung, and Zhaoyang Dong. Performance-based rough terrain navigation for nonholonomic mobile robots. In *Proceedings of the IEEE Industrial Electronics Society*, pages 2811–2816, 2003.
8. Nathan Ida and João P. A. Bastos. *Electromagnetics and Calculation of Fields*. Springer-Verlag, 1992.
9. Marin B. Kobilarov and Gaurav S. Sukhatme. Near time-optimal constrained trajectory planning on outdoor terrain. In *Proceeding of the IEEE International Conference on Robotics and Automation*, pages 1833–1840, 2005.
10. J. S. B. Mitchell. The weighted region problem: finding shortest paths through a weighted planar subdivision. *Journal of the Association for Computing Machinery*, 38(1):18–73, 1991.
11. Shankar Sastry. *Nonlinear Systems: Analysis, Stability, and Control*. Springer-Verlag, 1999.
12. Jonathan Richard Shewchuk. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In Ming C. Lin and Dinesh Manocha, editors, *Applied Computational Geometry: Towards Geometric Engineering*, volume 1148 of *Lecture Notes in Computer Science*, pages 203–222. Springer-Verlag, May 1996.
13. Alex Yahja, Sanjiv Singh, and Anthony Stentz. An efficient on-line path planner for outdoor mobile robots. *Robotics and Autonomous Systems*, 32:129–143, 2000.