Robust motion planning for rough terrain navigation

Alain Haït, Thierry Siméon, Michel Taïx

Published in IEEE Int. Conf. on Intelligent Robots and Systems
Kyongju, Korea, October 1999.

Robust motion planning for rough terrain navigation

A. Haït T. Siméon and M. Taïx

LAAS-CNRS
7, avenue du Colonel-Roche
31077 Toulouse Cedex - France
{hait, nic, taix}@laas.fr

Abstract: *This paper deals with motion planning for a mobile robot on rough terrain. We proposed in [9] a geometrical planner for articulated robots which can take into account uncertainty on the terrain and on the position of the robot. This paper aims at improving the robustness of the trajectory using landmark based approach.*

We consider regions of the terrain where natural landmarks are visible. We propose a two-step planning approach taking advantage of these regions to reduce position uncertainty. First a path is determined between the initial and goal configurations, based on simplified models of the constraints. Then a trajectory is planned along this path, verifying the validity and visibility constraints.

1 Introduction

Recent success of Sojourner rover mission on Mars gave a new fervor to planetary exploration with robots. Future missions may include complex tasks : pick up samples, long range displacements,... A higher level of autonomy may be required to perform such missions, Autonomous navigation on natural terrain remains however a complex and challenging problem.

Several systems for outdoor navigation have been developed. The *Rocky* robots [15] from Jet Propulsion Laboratory can reach specified positions, using a behavior control approach [12, 7]. *Nomad* [1] from Carnegie Mellon University operated more than 200 kilometers using various navigation modes (principally teleoperation). Another system from CMU, *Navlab* also performed autonomous navigation on natural terrain [17].

The adaptative navigation approach developed at LAAS within the framework of the EDEN experiment [3] demonstrated autonomous short-range navigation in a natural environment gradually discovered by the robot. The approach combines various navigation modes (*reflex, 2D and 3D*) in order to adapt the robot behavior to the complexity of the environment.

This paper addresses the problem of motion planning for the *3D navigation mode* [14], selected when very rugged terrain has to be crossed by the robot. On uneven or highly cluttered areas, the obstacle notion is closely linked with the constraints on the robot attitude. Planning a trajectory on such areas requires a detailed modeling of the terrain and also of the robot's locomotion system. Several contributions recently addressed motion planning [16, 13, 10, 9, 6, 5].
In the navigation experiments [14] performed using the planning algorithms of [16] , motion control was limited to executing the paths returned by this planner by relying on odometry and inertial data. The unpredictable and cumulative errors generated by these sensors, especially significant on uneven and slippery areas, often caused important deviations leading to a lack of robustness at execution. To overcome this problem, the robot may be equipped with environment sensors (*e.g.* cameras) that can provide additional information by identifying appropriate features of the terrain. Localization on these features allow to overcome the problem of cumulative errors. It however introduces new constraints; in particular the visibility of the terrain features has to be checked along the trajectory. Some contributions addressed this problem (*e.g.* [8, 11, 4]).

In this paper we extend the motion planner proposed in [16, 9] in order to consider a set of given landmarks (*e.g.* terrain peaks). The landmarks partition the terrain into regions where they remain visible from the robot's sensor. The approach allows to produce trajectories that remain, if necessary, inside these regions where the robot can navigate with respect to the given landmarks using closed-loop primitives relying onto the sensor's data.

2 The planning approach

2.1 Problem statement

We consider a geometric model of an articulated robot shown on Figure 1, right. The robot is composed of three axles linked by passive joints allowing to adapt its shape to the terrain relief, like Marsokhod robot LAMA (see Fig. 1, left). The terrain is described by surface patches defined from an elevation map in z associated with a regular grid in (x, y). A configuration is a vector $q = (x, y, \theta)$ specifying the horizontal position/heading of the robot in the global frame. Additional parameters $r(q)$ describing the angles of the articulations are determined from configuration q.

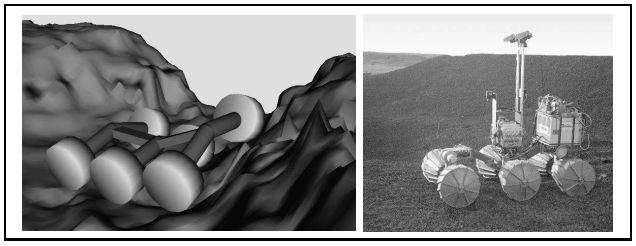

Figure 1: Example of placement - The mobile robot LAMA

The terrain model also contains a set of point landmarks corresponding to major terrain features that the robot should be able to track at execution. Whenever possible, landmarks are used to localize the robot; otherwise odometry is used, involving a growing position uncertainty along the trajectory.
Consequently, the motion planner must take into account:

- **validity constraints** related to the safety of the motion (*e.g.* stability of the vehicle, collision avoidance with the terrain and mechanical limits);
- **visibility constraints** which represent the ability of the robot to detect one or a set of landmarks from a given configuration;
- **position uncertainty** along the trajectory.

The planned trajectory is a sequence of valid configurations between initial and goal configurations q_{init} and q_{goal}. It also indicates the landmarks the robot should track during execution.

2.2 A two-step approach

In order to limit the complexity of the problem, the constraints are not considered simultaneously. We propose a motion planner based on the following two-step approach:

- the first step is based on simplified models of the constraints in order to obtain a 2D trajectory called **path**. This path is composed of portions where landmarks are visible and others where no landmark can be used during execution;
- in the second step, a 3D **trajectory** is planned along the path. This trajectory must verify the validity constraints, and along portions for which landmarks are visible, the visibility constraints for these landmarks must also be verified.

The final trajectory is thus a sequence of trajectories alternately verifying or not the visibility of landmarks. The main advantage of this approach is that the selection of the landmarks is done in the first step, with simplified models. The visibility constraints are then tested with these landmarks when it is necessary in the second step. Another advantage is that trajectory planning is fast because search is only done along the path and not all over the terrain.
The following sections describe the two steps, called **path planning** and **trajectory planning**.

3 Path planning

This step consists in determining a path, *i.e.* a sequence of points (x_i, y_j) on a grid with the same discretization as the terrain model. Initial point P_{init} of the path is the nearest point on the grid from initial configuration q_{init} (the same for P_{goal}).
A numerical propagation on the grid, initiated at P_{init}, is used to determine the path. It is based on simplified 2D models of the constraints.

3.1 Constraint models

In order to check the constraints without computing the complete robot placement $(q, r(q))$, 2D models are presented, relying on position $P(x_P, y_P)$ of the robot. To guarantee a correct link with the trajectory planning step, it is important that these 2D constraints reflect well the 3D ones.

3.1.1 Validity

A **cost bitmap** is computed by evaluating for each point of the terrain model, the slope and the roughness of a circular domain CD centered at this position, with a radius related to the size of the robot. It represents the difficulty for the robot to cross the domain (see Fig. 2).

3.1.2 Landmark visibility

The sensor position is estimated from the circular domain CD representing the robot (see Fig.3) : S is

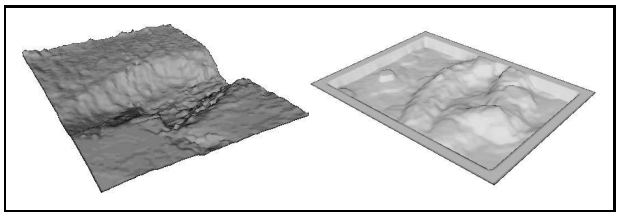

Figure 2: A terrain model and the associated cost bitmap

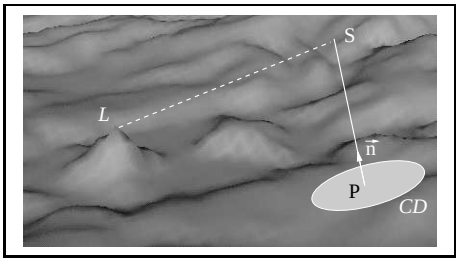

Figure 3: Landmark visibility from position P

placed on normal $\vec{n}$ to domain CD at P, and its position only depends on the position of the robot. In this case, two constraints must be verified to detect the landmark :

- *field of view* constraint : the distance between sensor S and landmark L is within some limit values;
- *non-occultation* constraint : L is not hidden by the terrain *i.e.* segment $[SL]$ does not intersect the terrain.

For each point of the terrain respecting the first constraint, the second one is tested, using the efficient collision checker developped for the trajectory planner [9]. A **visibility region** is thus associated to each landmark, defined by the set of points respecting both constraints (see Fig.4).

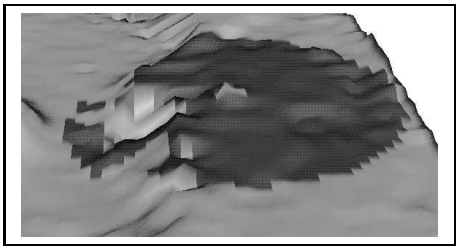

Figure 4: Landmark visibility region

3.1.3 Position uncertainty

For a position P, the uncertainty is represented by a circular domain centered in P, and with a radius $\mathcal{I}$.

The value of $\mathcal{I}$ depends on the path followed and the localizations on landmarks (see Fig. 5).
Cost value Γ_P at P with uncertainty $\mathcal{I}$ is the maximal value of the cost bitmap on the domain centered at P and with a radius $\mathcal{I}$.

Out of visibility regions uncertainty grows proportionally to distance and terrain difficulty :

$$\mathcal{I}_B = \mathcal{I}_A + f(d_{AB}, \Gamma_B)$$

In visibility regions uncertainty is computed in two steps (see Fig. 5). First uncertainty without localization is evaluated as presented above (*e.g.* on Fig. 5 $\mathcal{I}'_E = \mathcal{I}_D + f(d_{DE}, \Gamma_E)$). Then a localization model is used :

$$\mathcal{I}_E = \mathcal{F}(\{L_i, d_i\})\mathcal{I}'_E$$

$$with\ \mathcal{F}(\{Li, d_i\}) = \prod_{i=1}^{n}(1 - \sqrt{1 - \frac{d_i}{D_{max}}})$$

where $\{L_i\}_{i=1..n}$ is the set of visible landmarks from the current position, d_i is the distance between the sensor and landmark L_i, and D_{max} is the higher distance allowing to detect a landmark.

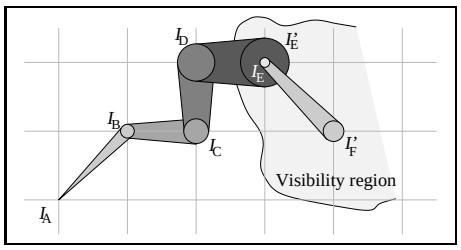

Figure 5: Uncertainty along a path on the grid

3.2 The propagation algorithm

Path search is based on a numerical propagation technique, similar to the one used in [2].
The propagation consists in a wavefront expansion of a numerical potential, obtained by integrating the cost (see section 3.1.1) and the uncertainty (see section 3.1.3) across the grid, starting from P_{init}. Each node B of the propagation contains the following informations :

- position on the grid (x_B, y_B) ;
- cost Γ_B ;
- potential $\Pi_B = \Pi_A + d_{AB}.\Gamma_B$ (A : previous node) ;
- uncertainty $\mathcal{I}_B$ as presented in §3.1.3 ;
- link to the previous node A.

The propagation starts form the first node with $\Pi_{init} = 0$ and stops when the wavefront reaches the goal P_{goal}. The solution path is then obtained by chaining the nodes back to the first one.

3.2.1 Propagation out of visibility regions

In this case, no localization on landmark is possible. Consequently potential Π_B and uncertainty $\mathcal{I}_B$ increase the same way, depending on the difficulty of the terrain.
For each node, cost, potential and uncertainty are evaluated. The node is then placed in a list sorted by cost and potential values. The wavefront expansion thus simultaneously considers the local terrain difficulty and the global cost of the followed path.
At each iteration, the first node of the list is expanded, using a 8-neighbor propagation. A new node is created if there is no other node corresponding to the same point (x_i, y_j) of the grid. Node creation is also forbidden if uncertainty exceeds a bound $\mathcal{I}^{max}$, in order to limit path search to realist values of uncertainty.
The propagation stops when the list is empty (failure), or when the wavefront reaches the node corresponding to P_{goal}. If uncertainty $\mathcal{I}_{goal}$ is lower than the expected limit $\mathcal{I}^{max}_{goal} (\leqslant \mathcal{I}^{max})$, a solution is found. Otherwise, the goal cannot be reached by the robot within the given uncertainty limits $\mathcal{I}^{max}_{goal}$ and $\mathcal{I}^{max}$.

3.2.2 Localization on landmarks

The position uncertainty decreases when a localization is performed in a landmark visibility region. Path planning takes into account localization with the uncertainty model presented in §3.1.3. Consequently, for a given uncertainty limit $\mathcal{I}_{max}$, the wavefront propagation covers a wider area than previously.
Modifications to the propagation algorithm concern the node expansion. Node creation now depends on the following conditions:

- uncertainty is lower than $\mathcal{I}_{max}$;
- there is no node corresponding to the same position, **or :**
- after a localization, the uncertainty of the new node is lower than the old node's one.

Consequently, for each visibility region reached by the wavefront, a new wavefront propagation is performed, overlapping the previous one. Various wavefronts can successively reach the goal, with different uncertainty values $\mathcal{I}_{goal}$. The propagation goes on while $\mathcal{I}_{goal} > \mathcal{I}^{max}_{goal}$. The case where no wavefront reach the goal with the expected uncertainty is considered as a failure.
The solution path is composed of portions in and out of the visibility regions. It is a sequence of positions between P_{init} and P_{goal} for which an evaluation of the uncertainty and the visible landmark are available.

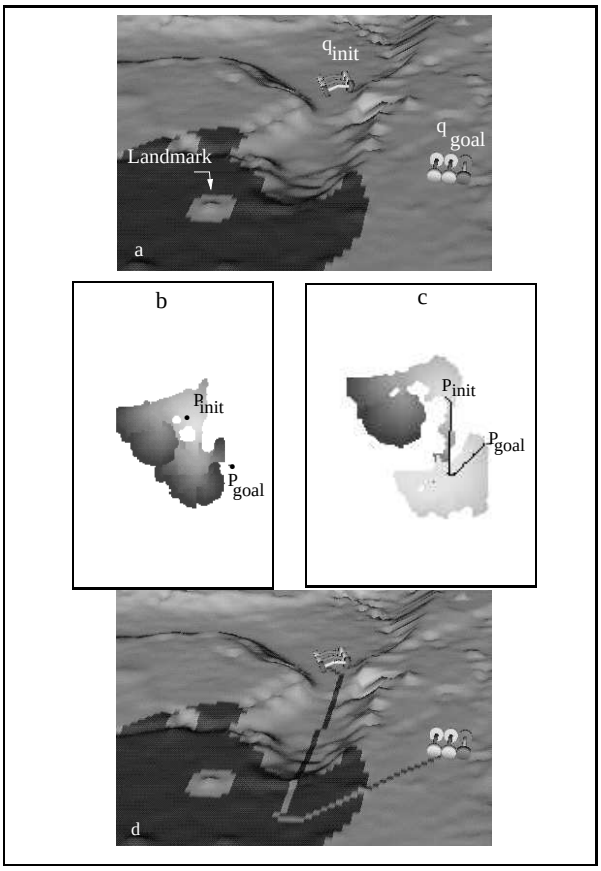

Figure 6: Path planning

3.3 Example

Figure 6 shows an example of path planning with one landmark visibility region. Fig. 6-**b** and Fig. 6-**c** present the wavefront propagations. Grey levels correspond to uncertainty value (from light to dark). When no localization is possible (**b**), uncertainty value is to high and the propagation does not reach P_{goal}. After a localization onto the landmark (**c**), the propagation reaches P_{goal}. Fig. 6-**d** presents the solution path onto the terrain. One may note that thanks to the cost bitmap, the path avoids uneven terrain, so ensuring consistency with the 3D validity constraints in the trajectory planning step.

4 Trajectory planning

The path planner provides a sequence of portions between P_{init} and P_{goal}, obtained with simplified models of the constraints. The trajectory planner then

searches for a trajectory (*i.e.* a sequence of configurations $q(x, y, \theta)$) following the path and respecting the validity constraints (cf. Fig. 7) and also the visibility constraints. This planner is based on the algorithms described in [9].

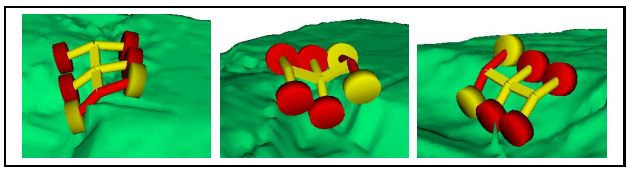

Figure 7: Examples of placement's constraints (collision, mechanical limits, stability)

4.1 Planning areas

The trajectory is searched on a domain called *planning area* around the portion of path. The area size depends locally on the evaluated uncertainty along the path. Trajectory planning is performed in two steps on this area, as in [9] : a preprocessing step to compute robot placements on the area, and then the planning step itself.

4.2 Graph search

The planning step incrementally builds a graph of discrete configurations that can be reached from the initial configuration by applying sequences of discrete controls during a short time interval. The arcs of the graph correspond to feasible portions of trajectory, computed for a given control. Only the arcs verifying the validity constraints are considered during the search. A numerical potential development is performed to guide the search along the path (*heuristic* function).
When the corresponding portion of path belongs to a landmark visibility region, the visibility constraint of this landmark is also tested along the arc. When the landmark is not visible at some points, a penalty is given to the associated arc for the search (if the landmark is not visible all along the arc, this one is not considered).

4.3 Complete trajectory

For the first portion of trajectory, the search starts at configuration q_{init}. It stops when the current configuration is in a domain $(dx, dy, d\theta)$ around the final point of the portion of path. This domain prevents from unexpected maneuvers near the goal. If the next portion belongs to a visibility region, the last configuration must also verify the visibility of the landmark. The next portion of trajectory begins at the last configuration of the previous portion, and so on until the last portion, where the goal configuration q_{goal} must be considered. The complete trajectory is then the sequence of portions between q_{init} and q_{goal}. It contains information about landmark visibility, and the relative position between the robot and the landmarks in order to move the sensor in the right direction.

5 Simulation results

The complete approach is illustrated onto a terrain model containing two landmarks. Figure 8 shows the corresponding visibility regions, the path found at the path planning level, and also the uncertainty evolution along the path.

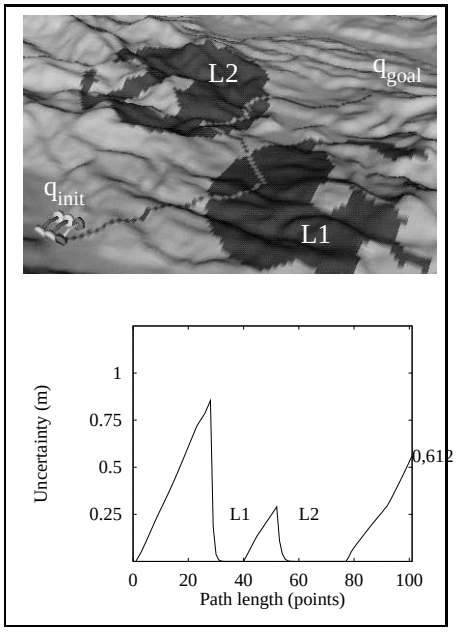

Figure 8: Path planning

The results of the trajectory planner are shown on Figure 9. First the heuristic function is computed on the planning area. Fig.9-**a** shows a planning area all along the path, and the associated potential values, leading to the goal. The planner then searches for a

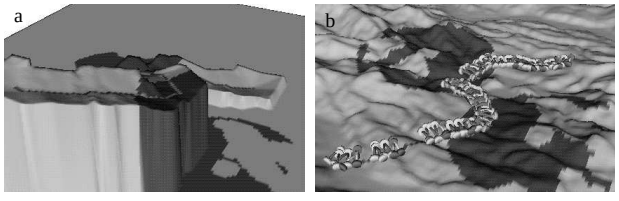

Figure 9: Trajectory planning

feasible trajectory, also respecting the visibility constraints. Fig.9-**b** presents the final trajectory. The complete planning approach for this example required less than one minute.

6 Conclusion

In this paper, we considered the problem of robust motion planning for a robot on a rough terrain. In order to improve the trajectory robustness, planning takes into account the presence of natural terrain features that may allow to localize precisely the robot. We proposed a two-step approach based on various models of the constraints. The first step relies on a numerical propagation to determine a 2D path. The second step searches for a trajectory along this path, respecting the validity and landmark visibility constraints. Tests on various terrain models demonstrated the ability of the planner to solve efficiently difficult problems.
p

References

[1] D. Bapna, E. Rollins, J. Murphy, M. Maimone, W. Whittaker, and D. Wettergreen. The atacama desert trek: outcomes. In *Proc. IEEE International Conference on Robotics and Automation, Leuven (Belgium)*, pages 597–604, 1998.

[2] B. Bouilly. *Planification de stratégies de déplacements robustes pour robot mobile*. PhD thesis, Institut National Polytechnique, Toulouse, France, 1997.

[3] R. Chatila, S. Lacroix, T. Siméon, and M. Herrb. Planetary exploration by a mobile robot: mission teleprogramming and autonomous navigation. *Autonomous Robots*, 2(4):333–344, 1995.

[4] X. Deng, E. Milios, and A. Mirzaian. Landmark selection for path execution. In *IEEE International Conference On Intelligent Robots and Systems, Yokohama (Japan)*, pages 1339–1346, 1993.

[5] S. Farritor, H. Hacot, and S. Dubowski. Physics-based planning for planetary exploration. In *Proc. IEEE International Conference on Robotics and Automation, Leuven (Belgium)*, pages 278–283, 1998.

[6] R. Fayek and A. Wong. Using hypergraphs knowledge representation for natural terrain robot navigation and path planning. In *Proc. IEEE International Conference on Robotics and Automation, Minneapolis (USA)*, pages 3625–3630, 1996.

[7] E. Gat, R. Desai, R. Ivlev, J. Loch, and D.P. Miller. Behavior control for robotic exploration of planetary surfaces. *IEEE Journal of Robotics and Automation*, 4(10):490–503, 1994.

[8] E. Gat, M.G. Slack, D.P. Miller, and R.J. Firby. Path planning and execution monitoring for a planetary rover. In *Proc. IEEE International Conference on Robotics and Automation, Cincinnati (USA)*, pages 20–25, 1990.

[9] A. Haït and T. Siméon. Motion planning on rough terrain for an articulated vehicle in presence of uncertainties. In *IEEE International Conference On Intelligent Robots and Systems, Osaka (Japan)*, pages 1670–1677, 1996.

[10] T. Kubota, I. Nakatani, and T Yoshimitsu. Path planning for planetary rover based on traversability probability. In *Proc. International Conference on Advanced Robotics*, pages 739–744, 1995.

[11] A. Lazanas and J.C. Latombe. Landmark-based robot navigation. *Tech. report STAN-CS-92-1428, Stanford University (USA)*, 1992.

[12] L. Matthies, E. Gatt, R. Harrison, B. Wilcox, R. Volpe, and T. Litwin. Mars microrover navigation : performance evaluation and enhancement. *Autonomous robots*, 2(4):291–311, 1995.

[13] Ch. Milési-Bellier, Ch. Laugier, and B. Faverjon. A kinematic simulator for motion planning of a mobile robot on a terrain. In *IEEE International Conference On Intelligent Robots and Systems, Yokohama (Japan)*, pages 339–344, 1993.

[14] F. Nashashibi, P. Fillatreau, B. Dacre-Wright, and T. Siméon. 3d autonomous navigation in a natural environment. In *Proc. IEEE International Conference on Robotics and Automation, San Diego (USA)*, 1994.

[15] D. Shirley and J. Matijevic. Mars pathfinder microrover. *Autonomous Robots*, 2(4):283–289, 1995.

[16] T. Siméon and B Dacre-Wright. A practical motion planner for all-terrain mobile robots. In *IEEE International Conference On Intelligent Robots and Systems, Yokohama (Japan)*, pages 1357–1363, 1993.

[17] A. Stentz and M. Hebert. A complete navigation system for goal acquisition in unknown environments. *Autonomous Robots*, 2(2), 1995.