HAL
open science

Interleaving Planning and Control of Mobiles Robots in Urban Environments Using Road-Map

Guillaume Lozenguez, Lounis Adouane, Aurélie Beynier, Abdel-Illah Mouaddib, Philippe Martinet

To cite this version:

Guillaume Lozenguez, Lounis Adouane, Aurélie Beynier, Abdel-Illah Mouaddib, Philippe Martinet. Interleaving Planning and Control of Mobiles Robots in Urban Environments Using Road-Map. International Conference on Intelligent Autonomous Systems, Jun 2012, Jeju Island, South Korea. pp.683-691, 10.1007/978-3-642-33926-4_65 . hal-00971722

HAL Id: hal-00971722

https://hal.science/hal-00971722

Submitted on 23 Feb 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Interleaving Planning and Control of Mobiles Robots in Urban Environments using Road-Map

Guillaume Lozenguez[12], Lounis Adouane[2], Aurélie Beynier[3], Abdel-Illah Mouaddib[1] and Philippe Martinet[24]

Abstract This paper presents a robot solution that allows to automatically reach a set of goals attributed to a robot. The challenge is to design autonomous robots assigned to perform missions without a predefined plan. We address the stochastic salesman problem where the goal is to visit a set of points of interest. A stochastic Road-Map is defined as a topological representation of an unstructured environment with uncertainty on the path achievement. The Road-Map allows us to split deliberation and reactive control. The proposed decision making uses a computation of Markov Decision Processes (MDPs) in order to plan all the reactive tasks to perform while there are goals which are not yet reached. Finally, from a brief explanation on how the approach could be extended to multi-robot missions, experiments in real conditions permit to evaluate the proposed architecture for multi-robot stochastic salesmen missions.

1 Introduction

Designing fully autonomous robots to achieve complex goals requires the fusion of severals capabilities from perception to control. Such design induces the need of engineers from different domains to work together regarding the type of used robots, their equipments and the targeted environment. For ground robots, the notion of

[1] GREYC, Campus Côte de Nacre, Bd Marchal Juin, BP 5186, 14032 Caen Cedex France
e-mail: [firstname.lastname]@info.unicaen.fr

[2] Institut Pascal - 24 Avenue des Landais - 63177 Aubiere Cedex France
e-mail: [firstname.lastname]@univ-bpclermont.fr

[3] LIP6, University Pierre & Marie Curie, Boîte courrier 169, 4 place Jussieu, 75005 Paris France
e-mail: [firstname.lastname]@lip6.fr

[4] IRCCYN, 1 rue de la Noë, 44312 Nantes France
e-mail: [firstname.lastname]@irccyn.ec-nantes.fr

mobility requires mapping the environment in order to localize the robot regarding its goals' positions. Two kinds of map models are provided in the literature, they permit to connect perception, decision and control: The occupancy grid maps [1] which are discretized metric representations of the robot's environment in a global basis; and the topological maps [2] which are graphs where nodes match particular locations and arcs represent path.

The robot's behavior has to converge efficiently towards goals while reacting safely to unexpected events. The most common deliberative approaches for robot are based on grid representation [3] [4]. The discretization is suitable to compute the policies. So, the occupancy grid map is used as a unified representation of the environment for the perception (localization and mapping), the deliberation and a step by step control. Grid representation strongly depends on the robot configuration. This kind of map is hard to produce off line with different data sources from the robot sensors. Furthermore, they are not efficient for large outdoor environments with difference in altitude and limited range sensors.

Reactive control approaches allow the robot to make safe and smooth movements using a simple and on-line local representation. Reactive control requires supervision to optimize the overall task achievement. Hierarchical architectures [5] [6] [7] allow to separate decision making from control with several levels of abstraction.

The paper focuses on mobile ground robots and presents a topological map (*Road-Map*) in a hierarchical architecture to allows the deliberation to plan high level actions. Considering the reactive control capabilities, we suppose that it is possible to decrease the deliberation input data size in order to focus on abstract and efficient maps for control supervision. The *Road-Map* which represents the environment has to integrate perception and control constraints and has to be suitable for decision making. For somes applications, it is interesting to be able to produce the *Road-Map* from aerial data.

The proposed approach is illustrated by the need of a fleet of cooperative robots to visit a set of point of interest given an initial approximate map (Fig. 1). Such situation can be met for search and rescue applications where an UAV (Unmanned Aerial Vehicle) can evaluate the situation and send an abstract map to the robot. Once the points of interest are allocated among the robots, each robot computes its

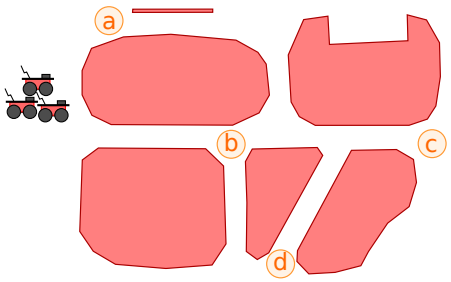

Fig. 1 Exploration area with 4 points of interest.

own plan to visit its attributed points of interest. The paper focuses on a single robot in the fleet and the control supervision from an initial plan. The plan is produced from the abstract *Road-Map* based on the UAV information. We show in this paper that the proposed approach works for urban environments.

The paper is organized as follows: Section 2 presents the robot navigation mission and the proposed solution. Experimental results are discussed in Section 3. Finally, the paper ends by future research directions.

2 Robot Navigation and Decisional Strategy

This section present the proposed architecture for the autonomous mobile robots control. It is organized in two levels: the reactive and the cognitive. The reactive level includes perceptions and local controls of the vehicle. The cognitive level includes a model of the environment, the decision making algorithm to plan movements and a supervision part to connect the built plan to local controls.

2.1 Reactive Level for Navigation

The approach is evaluated with a simple perception module without global sensors as GPS or external global camera. Each robot is equipped to estimate its pose and to detect the local free navigation space. The localization is done by odometry in a global Cartesian basis and an error grows with movements. The local free navigation space is defined in the robot basis by the list of limit points. From the limit points, semantic perception states (*PS*) are defined as: • *"left to obstacle"* or *"right to obstacle"* if one obstacle is detected in the left or the right of the robot • *"corridor"* if obstacles are detected with a unique way between them • *"intersection"* if there are more than one possible ways • *"blocked"* if obstacles prevent any movement (like a dead-end corridor with a robot without a rear perception). • *"undefined"* if no obstacle is detected or the obstacle configuration is unstable

The perception is designed with three refiners: global localization (odometer), local free navigation space detection and perception state estimation. In fact, we consider that the recognition of particular places or landmarks in the sensors input stream allows a moving robot to, punctually, localize itself [2][8][9] and re-initialize its odometry.

In parallel, we define several controllers to: reach a local target, navigate on the voronoi or circle an obstacle. The reaching controller is the default task of our robots. We design it as Multi-controller [10] to allow the robot to smoothly switch between the attraction to a target and obstacle avoidance. Our voronoi controller reaches the middle of a path in corridor perception state. We also define a left or right circle controller to allow the robot to navigate along an obstacle by keeping a

safe distance. This controller is available in all perceptive states where at least one obstacle is detected.

2.2 Road-Map Definition

A *Road-Map* $< W,\ P >$ (Fig 2) is a topological map where: the nodes represent particular way-points; the edges represent the paths connectivity between nodes. The knowledge is defined regarding the capacities of our perception and control modules. We define a way-point $w \in W$ regarding the expected robot pose given by the odometer localization $(x_w,\ y_w,\ \theta)$ and a path $p \in P$ is characterized by the attached active controller regarding the expected perceptive state $s_p \in PS$.

Several attributes complete the path p knowledge from w_p to w'_p in order to permit efficient control and decision making. An attribute c_p gives the associated movement cost; it depends on the distance and the quality of the path. In fact, the movement cost between two positions varies according to the type of ground, the slope or the obstruction of the path. According to the possible error during a movement, a normalized function d_p (deviation) gives the probability to reach other way-points. The d_p function is initialized proportionally to the distances from w'_p to all other connected way-points to w_p. The *Road-Map* $< W, P >$ is defined as:

$$W = \{ \quad (x_w, y_w, \theta_w) \in \mathbb{R}^3 \quad \},\ P = \{ \quad (w_p, w'_p, s_p, c_p, d_p)$$
$$w_p,\ w'_p \in W,\ s_p \in PS,\ c_p \in \mathbb{R},\ d_p : T \rightarrow [0,1] \quad \}$$

Actually, the designed *Road-Map* could be defined as a mix of a voronoi graph and a visibility graph when the voronoi could not be locally detected. Structuring knowledge in a stochastic collision-free connectivity permits to plan a safe and efficient path between two positions [11]. Given a global topology as the *Road-Map*, the deliberation module has to cooperatively plan a policy to reach several goals.

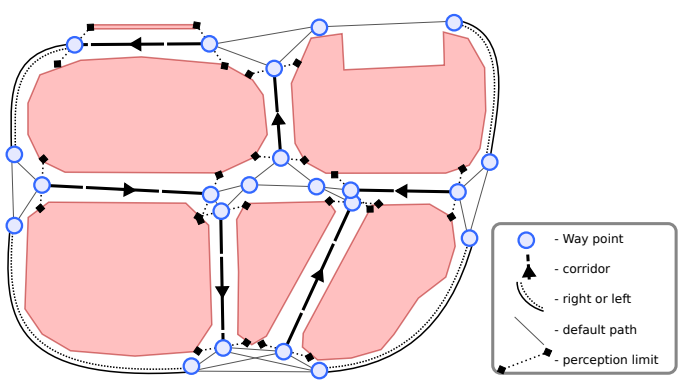

Fig. 2 Corresponding *Road-Map* to the environment described in Fig 1.

2.3 Policies Computation

The aim of decision making is to associate a reactive task to perform (action) to each robot state in order to allow the robot to reach its mission by minimizing the needed resources. Due to the uncertainties related to the *Road-Map* and the reactive control, we chose to use MDPs solved by each robot. An MDP is defined as a tuple $< S, A, t, r >$ with S and A are respectively, the state and the action sets that define the system and its control possibilities. t is the transition function defined as $t : S \times A \times S \rightarrow [0,1]$ that gives the probability $t(s,a,s')$ to reach the state s' from s by doing action $a \in A$. The reward function r is defined as $r : S \times A \rightarrow \mathbb{R}$, $r(s,a)$ gives the reward obtained by executing a from s.

Regarding the treated problem, an MDP state s needs to include the last recognized position $w_s \in W$ of the robot and the set of already visited points of interest I_s. An action a is added for each path $p_a \in P$ for reaching a way-point w'_{p_a}. Furthermore, the action of visiting a point of interest w_s is modeled by a loop path p_s.

$$S = \{ \ (w_s, I_s) \mid w_s \in W, \quad I_s \subset I \ \}$$
$$A = \{ \ (p_a) \mid p_a \in P \ \} \cup \{ \ (p_s) = (w_s, \ w_s) \mid w_s \in I \ \}$$

As the states and actions are directly computed from the Road-Map, the transition and reward functions are build from the attribute attached to each corresponding path p_a of an action a. When executing the available action $a = p_a$ from the way point w_s, the transition function t returns the probabilities to reach neighbor positions according to the deviation function d_{p_a}. Deterministic transition are added for action corresponding to visit a new points of interest (p_s).

The reward function returns a negative value regarding the movement cost (linked to the path), and a positive constant gain g if a new point of interest is reached. Optimally solving the MDP consists in searching an optimal policy π^* that maximizes the expected gain. A policy $\pi : S \rightarrow A$ is a function which associate an action to each state. The value iteration algorithm [12] permits to compute the optimal policy π^* and the associated value function V^{π^*}. Previous study [13] allows the robots to compute, on-line, sub-optimal policies involving a large number of point to visit. It is done by splitting the *Road-Map*, into regions and applying a hierarchical solving.

2.4 Supervision and Localization

The individual policy gives the succession of paths to reach in the *Road-Map* in order to visit the attributed points of interest. Each path involves the next targeted way-point, the expected perception state along the path and the controller to activate. The control supervision monitors the odometrer position and the perceptive state to verify which way-point is reached by the reactive control. While a way-point is crossed, the reactive task is upgraded regarding the policy. This allows the robot to adapt its actions in function of the deviation due to the reactive control. A way-

point is considered crossed when the robot is in a given safe distance around this way-point.

In parallel, the robot has to re-initialize the odometer. In fact, while reaching a way-point, the distance to it decreases. In a close radius around the reached way-point, the instant where the distance between the robot and the way-point start growing, the odometer is automatically re-initialized by considering the robot is just-on it. The re-localization is particularly used with voronoi or circle control where the way-point is not directly reached but the trajectory of the robot is defined by its perception. For example (Fig 3), if the robot reaches a target position before the end of a corridor, regardless the odometry, the robot has to cross the position by the voronoi controler.

In practice, the *Road-Map* has to be predefined with way-points involving coherent positions and orientations. That means the way-points are in the voronoi of corridors and at a safe distance of circled obstacles. Furthermore, the way-points have to be defined in local stable orientation area to correct the robot orientation with the best possible accuracy while the robot is not exactly on the targeted way-point.

We propose a navigation process as simple as possible in order to illustrate the capability of the architecture to connect the reactive level with the cognitive level. By using more sophisticated refiner, it is possible to develop more accurate landmark localizations or to work in more complex environments (dynamic, with map error).

3 Experiments

The proposed solution is tested by computer simulation and on real mobile ground robots in an experimental outdoor urban area (Fig 4). For this experiments, the *Road-Map* is defined by an human operator. The free navigable space is delimited by sidewalks and 2 pioneer robots which have to visit a set of 12 points of interest and return to their initial positions. The robots are equipped with a laser at 45 degrees to detect sidewalks that are around one meter in front of the robot and establish the perception states. The area size is 32×26 meters and the robot are parametrized by 0.8 and 4.0 meters for the safe and close distances (*cf.* video[1]).

The multi-robot coordination is done by dividing the set of points of interest in order to allocate part of mission to each robot of the team. Then each robot could perform its own sub-mission alone. Previous study [13] allows a group of robots to compute the allocation in a centralized way where a leader robot search to maximize the sum of individual expected gains.

The communication between the robots and the operator PC is done by Wifi. The operator initializes the robot positions and orientations, builds the map and selects the points of interest (Fig 2) before sending data to all the robots. A way-point in the *road-map* is defined regarding the radius of robot's perception and an aerial

[1] www.greyc.fr/node/1629

snapshot where the sidewalk is easily detectable. It is important to respect the robot's perception and control modules to draw the expected way-point position closer as possible to real robot position.

Experimental results (Fig 5 and 6) show that the *road-map* allows the robots to maintain their localization despite the few numbers of information and their inaccuracy. In this paper, we are more interested in the connexion between cognitive and reactive levels. We observe in simulation (Fig 5) and in real condition (*cf.* the attached video) that the control supervision using the deliberative policy allows the robot to smoothly move between two distant positions involving several way-points.

We notice that the third robot with the 3 points of interest in the south of the map (Fig 5) starts returning to its initial position by circling the local obstacle from the right. At this moment, the policy gave a task of reach a way-point behind the robot. By turning around, the reactive control circles the obstacle and reaches a bad way-point. We observe (as expected) that the supervision adapts correctly the planned path to the initial position according to the policy.

In fact, in this experiments, the re-localisation process has a blind trust in the operator *Road-Map*. The *Road-Map* parameters (way-points positions and orientations, safe and close distances) needed adjustments to assume the robot re-localisation at any position. This is notably true in the real experiments where the robot has to go and come back from the point of interest 2 to the corner north-east only with the odometer (Fig 6). This round trip measures 2×18 meters and the robot reaches the target (point of interest 4) with an error around 3 meters and come back with an error around 3.8 meters. This is translated by a gap of the curve trajectory

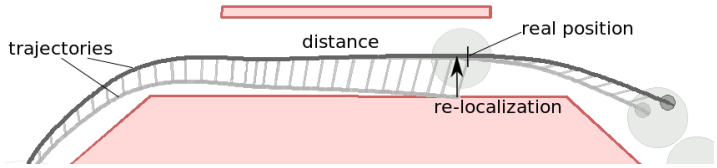

Fig. 3 Robot re-localization after a voronoi control.

Fig. 4 One pioneer robot, the experimental area and the aerial view.

resulting of the avoided sidewalk near the point of interest 2 (Fig 6). At the end, the robot corrects its gap to finish with an error of 0.8 from its start position.

The odometer re-initialization is correct but not perfect (Fig 5 and 6). The process shares an error in the orientation of the obstacle limit (in relation of the Degrees of freedom of the used control) and the error grows while the robot moves in the same direction (Fig 3).

In conclusion, without considering an uncertainty about the *Road-Map*, its definition by the operator has a great importance in the success of the robot movements. Regarding our reactive level, a way-point has to be positioned in retreat of intersection areas to assume stability on the orientation and anticipate a switch between controllers. The error on the map coupled to the odometric error has to be bounded to assume that the robot is able to reach the farthest next way-point where the re-localization involve a different orientation.

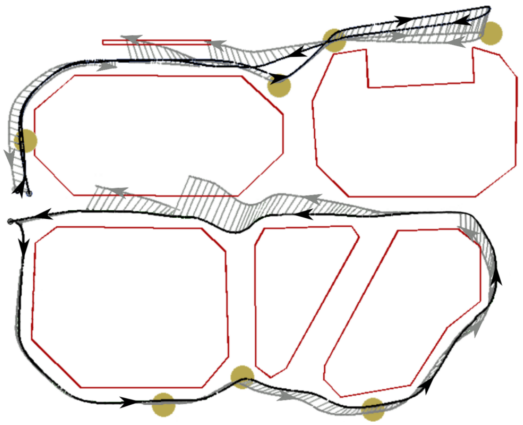

Fig. 5 Real (dark) and supposed (gray) trajectories of the 2 robots (Computer simulated result).

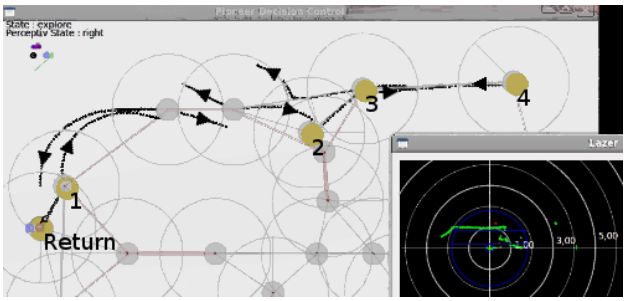

Fig. 6 One robot belief state at the end of a real experiment (*Road-Map*, attributed points of interest, supposed trajectory and laser snapshot).

4 Conclusion

A control architecture based on a *Road-Map* to connect deliberation and reactive control is presented in this paper. Splitting robot control into 2 levels provide flexible solution in regard to the needed robot functionality. The *Road-Map* allow us to merge decision making and reactive control modules developed separately. The use of this solution is illustrated by computing and executing collaborative policies in order to visit a set of points of interest with a fleet of robots. The proposed distributed solution allows the fleet to build a valid task achievement. Experiments in real conditions validate the approach in an urban environment using a *Road-Map* based on UAV information.

In future work, we want to study the consequences of map actualizations on the decision making and introduce uncertainty on the map knowledge. The perceptive module will also be increased to detect robots and humans, the goal is to permit the robots to maintain their localization and their policies in dynamic environment.

References

1. A. Elfe, "Sonar-based real-world mapping and navigation," *Robotics and Automation*, vol. 3, pp. 249–265, 1987.
2. B. Kuipers and Y.-T. Byun, "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations," *Journal of Robotics and Autonomous Systems*, vol. 8, pp. 47–63, 1991.
3. C. V. Goldman and S. Zilberstein, "Decentralized control of cooperative systems: Categorization and complexity analysis," *Journal of Artificial Intelligence Research*, vol. 22, 2004.
4. W. Burgard, M. Moors, C. Stachniss, and F. Schneider, "Coordinated multi-robot exploration," *IEEE Transactions on Robotics*, vol. 21, 2005.
5. R. Alami, R. Chatila, S. Fleury, M. Ghallab, and F. Ingrand, "An architecture for autonomy," *Journal of Robotics Research*, vol. 17, no. 4, pp. 315–337, 1998.
6. R. Volpe, I. Nesnas, T. Estlin, D. Mutz, R. Petras, and H. Das, "The claraty architecture for robotic autonomy," in *Aerospace Conference*, vol. 1, 2001, pp. 121–132.
7. G. Mourioux, C. Novales, and G. Poisson, "Control robot by a generic control architecture," in *IEEE International Conference on Intelligent Robots and Systems*, 2007, pp. 3050 – 3055.
8. S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation," *Artificial Intelligence*, vol. 99, no. 1, pp. 21–71, 1998.
9. I. Ulrich and I. Nourbakhsh, "Appearance-based place recognition for topological localization," in *International Conference on Robotics and Automation*, vol. 2, 2000, pp. 1023–1029.
10. L. Adouane, "Hybrid and safe control architecture for mobile robot navigation," in *Autonomous Robot Systems and Competitions*, 2009.
11. R. Alterovitz, T. Siméon, and K. Goldberg, "The stochastic motion roadmap: A sampling framework for planning with markov motion uncertainty," in *Robotics: Science and Systems*, 2007.
12. M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 1994.
13. G. Lozenguez, L. Adouane, A. Beynier, P. Martinet, and A.-I. Mouaddib, "Map partitioning to approximate an exploration strategy in mobile robotics," in *Advances on Practical Applications of Agents and Multiagent Systems*, vol. 88, 2011, pp. 63–72.