Robotics and Autonomous Systems 60 (2012) 609–619

Contents lists available at SciVerse ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Trajectory planning for a car-like robot by environment abstraction

Narcis Ghita, Marius Kloetzer *

Department of Automatic Control and Applied Informatics, Technical University "Gheorghe Asachi" of Iasi, Romania

ARTICLE INFO

Article history:
Received 2 July 2011
Received in revised form
28 November 2011
Accepted 9 December 2011
Available online 17 December 2011

Keywords:
Path planning
Car-like robot
Nonholonomic robot
Path following

ABSTRACT

This work proposes a fully automatic planning and control strategy for solving a navigation problem for a car-like robot with non-negligible size and constraint control inputs. The approach uses cell decompositions for abstracting the robot behavior to a final state description on which the planning problem is solved. As part of the solution, we obtain a ranking of different cell decomposition types that are suitable for planning the motion of a car-like robot. The originality of our method mainly comes from the iterative procedure for finding a feasible path based on cell decompositions. Although the approach is not complete, it benefits from a fully-automatic planning and control strategy and from a reduced computational complexity. The solution is implemented as a user-friendly freely-downloadable MATLAB package. This may come as a handy tool for employing the strategy for automatic planning and control of a car-like robot in a real scenario.

1. Introduction

Path planning and controlling of mobile robots draws increasing attention during the last two decades [1–3]. The classical navigation problem consists in finding a control strategy in order to move a mobile robot from a start position to a desired one, without colliding with static obstacles. In the majority of such problems, an optimal solution is rather unlikely to be found, or the approach would be too expensive from the computational point of view.

Most of the works referring to indoor path planning focus on robots with a simple mathematical model (usually holonomic robots) mainly because of the shrink maneuvering space, the turning capability, and the resulting small computational effort necessary for finding a feasible solution. In these scenarios, some authors focus on adding more expressivity to the classical problem of path planning, by incorporating high-level, human like specifications for the motion task [4–10]. For outdoor applications, the main focus is on nonholonomic robots mainly because of the large number of the existing car-like platforms. In this case, one of the priorities is on making the method attractive from the computational point of view [11] and on ensuring continuity in command inputs [12]. Another worth noticing aspect is that one of the main tasks for outdoor path planning applications is the exploration of partially known or unknown environments. In this case, available approaches involve framed quad-tree based algorithms [13,14] or rapidly-exploring random trees [15,16]. Although attractive from the computational point of view, some of these algorithms do not guarantee collision free movements on designed trajectories.

Navigation problems for mobile robots are usually solved in two steps, namely a path planning part, whose output is a reference trajectory for the robot, and a trajectory following part, whose output is a continuous controller that keeps the robot as close as possible to the reference trajectory.

Path planning methods can be divided into two main categories, based on the employed mathematical model of the environment. Thus, some approaches directly work on the initial (continuous) environment, while others abstract the environment to a finite (discrete) representation. In the first case, the solutions are usually based on potential and navigation functions [17,18], where the planning and the control part are interconnected. The main difficulty consists in designing such functions without local minima and in handling non-spherical obstacles. In the second case, the approaches can be further divided into ones that use roadmaps (e.g., visibility graphs or Voronoi diagrams) [2,3], and others based on decomposition of the free space into cells (geometric shapes) [2,3,19,20]. It is worth mentioning that for holonomic robots an optimal solution from the point of view of the length of the resulted path can be obtained based on visibility graphs [2,3]. A drawback of visibility-graph-based methods is the closeness of robot's trajectory to obstacles. For some nonholonomic robots, an optimal solution can be found in a free infinite environment, e.g. such approaches for car-like robots are described in [21–24].

The task of designing control laws that allow tracking given reference trajectories reduces to a control problem with constraints issued by the robot dynamics. However, it is possible that for nontrivial dynamics the trajectory tracking cannot be enforced. Therefore, for handling specific robots, path planning and controller

* Corresponding author. Tel.: +40 232 230751; fax: +40 232 230751.
E-mail addresses: narcis_ac@yahoo.com (N. Ghita), kmarius@ac.tuiasi.ro (M. Kloetzer).

0921-8890/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2011.12.004

design are interconnected, either by restricting the path planning based on the control capabilities of the robot, or by iterating both parts until a feasible solution is obtained.

In this work, we approach the problem of solving a navigation problem for a car-like robot of non-negligible size, in a bounded environment cluttered with static obstacles. The map of the environment is a priori known, and we focus on car-like robots (wheeled nonholonomic robots) described by their kinematics.

In solving path planning problem for this kind of robots the dynamics are frequently ignored, while from the point of view of the control a discussion on when to use kinematic models and when it is necessary to use dynamic models can be found in [25]. For controlling car-like robots, the majority of available solutions propose trajectory tracking methods [26–29]. An interesting control policy for nonholonomic systems is designed in [12,30], where the environment is abstracted into a finite set of curved-surface cells, and the continuity in command angle is ensured by increasing the space dimension of the model. Besides complexity issues, a drawback of the proposed method is the lack of an automated strategy for abstracting the environment. The planning part of our approach is based on cell decomposition methods, and it ensures an output trajectory that can be followed by the car-like robot. The trajectory following part consists of an adapted version of the virtual vehicle approach proposed in [31,32].

The current paper has two main contributions: first, we develop a fully automated and computationally feasible control strategy for solving the classical navigation problem for car-like robots evolving in planar bounded environments cluttered with static obstacles. Second, we give a comparative analysis of the algorithm efficiency with respect to various environment abstraction methods (decompositions).

The proposed solution for the navigation problem consists of five main steps: in the first step, the environment is abstracted into a finite set via cell decomposition methods. In the second step an angular path that links the start position with the end one is constructed. Third, based on the angular path and on the steering capabilities of the car-like robot, a smooth path that can be followed by the robot is obtained. In the fourth step we test the feasibility of moving on the obtained smooth path, from the point of view of collisions with obstacles. The second, third and fourth steps are iterated until either a solution to the path planning problem is found, or the problem is declared unfeasible. If a solution is obtained, the last step of the approach consists in designing a trajectory-following control strategy that keeps the robot as close as possible to the designed smooth path.

The remainder of this paper is organized as follows. Section 2 presents some preliminary notions and formulates the problem, while Section 3 describes the algorithmic solution we provide for solving the path planning problem. In Section 4 we include a comparative analysis of various cell decompositions, we provide general recommendations on choosing a decomposition type that is advantageous for the path planning part of a given problem, and finally we discuss some limitations of our planning approach. Section 5 presents a modified virtual vehicle strategy used for controlling the motion along the generated path. Examples supporting the proposed method are included in Section 6, while concluding remarks and possible future research directions are formulated in Section 7.

2. Problem statement

Section 2.1 introduces the car-like model and the assumptions we consider, and Section 2.2 formulates the main objective of this work and outlines our approach. The presentation also comments on the connections with previous developments reported in the literature.

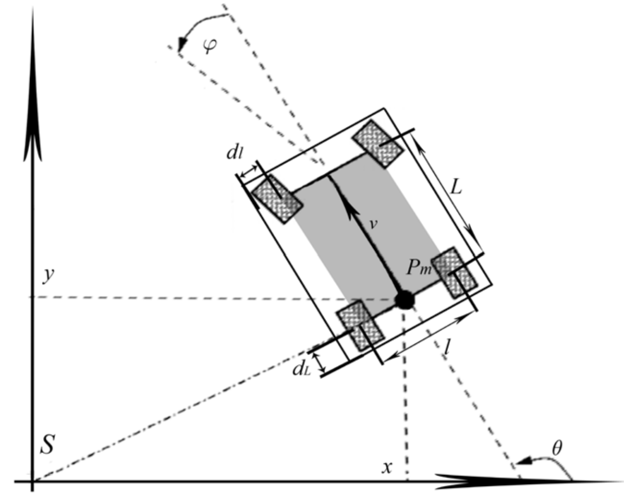

Fig. 1. Representation of a car-like robot in a static frame *S*.

2.1. The kinematic model

Throughout this work we consider a car-like robot described by the kinematic model given in (1), and for detailed information on this model we refer to [25,33]. As recommended in [25], a dynamic car-like model should be used when great velocities or masses are considered. In contrast, for path planning of a low velocity and mass car-like robot, the kinematical model usually suffices and it is generally used for obtaining computationally feasible approaches.

$$\begin{cases} \dot{x} = v \cdot \cos(\theta) \\ \dot{y} = v \cdot \sin(\theta) \\ \dot{\theta} = v \cdot \tan(\varphi)/L, \end{cases} \tag{1}$$

where (x, y) are the Cartesian coordinates in a fixed frame (S) of the reference point P_m, located at mid-distance of the actuated wheels. Angle θ characterizes the robot's chassis orientation with respect to frame S, and L is the distance between the rear and front axle. The control inputs are v, which is the vehicle's velocity, ensured by the rear wheels, and φ, which is the vehicle's steering wheel angle, imposed by the front wheels and measured with respect to the current chassis orientation, as depicted in Figure. 1. We assume a constant velocity v and a steering angle restricted by physical limits of the car-like robot, $\varphi \in [-\varphi_{\max}, \varphi_{\max}]$. Fig. 1 depicts the car-like robot and the involved notations. The distance between wheels from the same axle is denoted by l, and the size of the robot is a rectangle with sides $L + 2d_L$ and $l + 2d_l$ respectively. The size of the robot will be considered in Section 3.4 for testing possible intersections with obstacles.

2.2. Brief description of the navigation strategy

In this subsection we briefly discuss the key elements of the proposed navigation strategy, namely Objective, Path planning (trajectory generation), and Control (path following). A detailed presentation of the algorithmic/mathematical foundations for Path planning and Control is separately organized below, namely in Sections 3 and 5, respectively.

Objective

Given a car-like robot with model (1), which can evolve in a bounded environment cluttered with n convex polygonal and static obstacles, a start position (x_0, y_0) and a final position (x_f, y_f), find a strategy that drives the robot to the goal (final) point while avoiding any collision with obstacles.

We assume that the initial orientation of the robot's chassis can be chosen, while the final orientation is not of interest. We