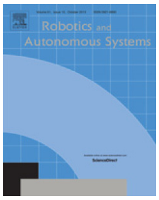
Robotics and Autonomous Systems 79 (2016) 1–11

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

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



Optimisation based path planning for car parking in narrow environments

Patrik Zips *, Martin Böck, Andreas Kugi

Automation and Control Institute, Vienna University of Technology, 1040 Vienna, Austria

HIGHLIGHTS

- Path is calculated by solving several local static optimisation problems iteratively.
- Driving direction changes are locally determined based on simple rules.
- Obtained paths are similar to manoeuvres of a human driver.
- Tree-based extension guides the local planner efficiently.
- High rate of success in narrow environments with low computing costs.

ARTICLE INFO

Article history:
Received 2 March 2015
Received in revised form
17 February 2016
Accepted 26 February 2016
Available online 4 March 2016

Keywords:
Non-holonomic path planning
Car-like robots
Static optimisation
Minkowski sum

ABSTRACT

In the last decades, a large variety of robot motion planners emerged. However, manoeuvring in very narrow environments, e.g., for common parking scenarios, cannot be reliably handled with existing path planners at low computing costs. This is why, this paper presents a fast optimisation based path planner which specialises on narrow environments. In the proposed approach, the kinematic differential equations are discretised. For the resulting discrete path segments, a static optimisation problem is formulated to determine the path independently of the considered scenario. In each iteration step, the path length is also optimised to handle close distances to the obstacles as well as longer driving distances. Due to the local nature heuristic rules for driving direction changes are formulated which intend to imitate the behaviour of a human driver. The drawback of the local nature is the lack of global information to handle scenarios with obstacles blocking the path. To overcome this problem, a tree-based guidance for the local planner is introduced. The landmarks for the tree are implicitly created by the local planner allowing an efficient exploration of the free configuration space. The performance of the algorithm is evaluated utilising Monte-Carlo simulations and compared to state-of-the-art path planning algorithms.

1. Introduction

One prevailing topic of recent research in automotive industry is autonomous driving, which is especially challenging in urban environments. There, narrow corridors, tight turns, and the partly unpredictable behaviour of other traffic participants have to be handled in a systematic way. A special topic within this area of research is automated parking control, which is not only useful for autonomous vehicles, but also in conventional cars as parking assistance system. A necessary requirement for such a system is a path planner which plans a feasible path between the current position and a specified parking position. Hereby, the challenge arises from the non-holonomic constraints of the car and the proximity to the obstacles.

In robot path planning, several algorithms have been developed to tackle this problem. One class of path planners rely on building up a roadmap in the considered environment. The map is set up in a learning phase at start-up and saved as a graph with the landmarks as nodes and the feasible paths as edges. After the learning phase, the path planning can be performed in a very efficient way by using this graph. A famous representative of this class is the so called kinematic roadmap. Based on "Ariadne's Clew" [1,2], it consists of a global planner EXPLORE and a local planner SEARCH. The global planner places landmarks in the reachable region of an already existing landmark and the local planner tries to connect these placed landmarks with the target position. All landmarks together are called kinematic roadmap. A similar approach is the "probabilistic roadmap" [3]. Here, landmarks are randomly placed

---

* Corresponding author. Tel.: +43 158801 376260; fax: +43 158801 37699.
E-mail address: zips@acin.tuwien.ac.at (P. Zips).

http://dx.doi.org/10.1016/j.robot.2016.02.004
0921-8890/

in the free configuration space based on a certain probability distribution. In this approach, sensor uncertainties can be explicitly considered, e.g., by utilising Monte-Carlo simulations to obtain a probability of success for each edge of the graph [4].

Another class of methods builds up a tree of robot motions beginning from the starting position. In this context, the most popular representative are the Rapidly-exploring Random Trees (RRT) [5]. Here, the point from where the tree expansion starts is chosen close to the target position in each iteration. This size of the Voronoi regions together with a probability distribution serves as decision criterion. To account for kinematic constraints of a robot, the tree can be expanded using a constant input for a certain driving distance. For example, randomised inputs [6] and inputs based on the analytic solution [7] are used. Integrating the differential equations of the robot for a specified distance with the chosen input yields a new configuration (also called node) from which the process is repeated. If a node is in collision, it is discarded and an existing node in the tree is picked as a new starting point. In this manner, a path can be iteratively constructed.

Path planning for cars and car-like robots received special attention in robot motion planning. By focusing on these systems, improved convergence and path quality can be achieved. The first works in this field trace back to Dubins, who analysed the paths for a particle with constrained curvature [8]. It is shown that the shortest path for this particle in the (x, y)-plane is composed of lines and arcs with minimal curvature. These ideas can be directly applied to a car, as the non-holonomic and the physical constraints result in a constrained curvature. Reeds and Shepp generalised these results for a car which drives forwards and backwards [9]. They defined nine curvature classes, which can describe every optimal path between any points in the (x, y)-plane considering the orientation. However, in [8,9] only obstacle free environments are considered. Therefore, many algorithms combine these ideas with a suitable obstacle avoidance scheme. For example, a path into a parking spot can be obtained by stringing together lines and minimal curvature arcs, see, e.g., [10–12]. In [13] these lines and arcs are defined as translation (TM) and rotation movements (RM). Starting from the parking position, N-cycles of every possible TM–RM–TM combination are linked together, where N has to be chosen according to the available computing time and the problem complexity. A higher number of iterations N consequently leads to a larger computing time. Instead of lines and arcs other curves can be used to obtain smoother paths, e.g., β-spline [14,15], Beziér [16] or polynomial [17] curves.

Another method based on Reeds and Shepp's curves combined with the well-known A*-algorithm is proposed in [18]. Here, the system equations are integrated for a certain distance with a constant input. The continuous end configuration of the integration is assigned to a discrete cell. A heuristics relying on Reeds and Shepp's curves and a holonomic path finder guides the path planner towards the target position. The input of the system also includes the driving direction which allows to calculate driving direction switching points. A cost penalty for changing the driving direction aims at minimising the overall number of driving direction switching points.

To handle the non-holonomic characteristics of a car, [19] proposes an algorithm which computes a holonomic path for a given environment and tries to move along this path under consideration of the non-holonomic constraints. Due to the fact that a car is small-time-controllable [20] this is always possible, but often results in highly manoeuvring paths particularly in narrow environments. Thus, different methods to follow a holonomic path are proposed. In [21] the differential equations of the car are transformed into a chained-form system and sinusoidal inputs are applied. In [22] Dubins curves are used to obtain the non-holonomic path and in [23] a local continuous curvature planner using clothoids, as presented in [24], in combination with a shortest feasible path metric [25] is used.

Other approaches rely on solving an optimal control problem. Here, the challenge arises from the mathematical description of the obstacles. In [26] the obstacles are discretised to obtain boundaries composed of a finite number of points. Every point has a potential field value which increases if the vehicle gets closer. The values of all points are summed up in one inequality constraint which has to be lower than or equal to zero. This inequality can only hold if no point collides with the vehicle. The drawback of this method is the inaccurate description of the obstacles. By adding more points, the model gets more precise but the computing time increases.

The described path planners, among others, are able to calculate feasible paths for most environments. Nevertheless, no algorithm yields satisfactory results for typical car parking which consists of the parking at a parking spot as well as parking scenarios at a parking deck or in a narrow parking lot. The methods of classical robot path planning, like the roadmap and the tree-based methods, rely on discretising the available space or assigning a certain distance for integration. Thus, in narrow environments the discretisation has to be decreased which largely increases the computing costs for longer driving distances. Methods specialised on car parking are able to obtain paths in narrow environments in an efficient way for one certain scenario, like, e.g., parallel parking. However, they cannot calculate feasible paths in more general environments, like, for instance, a parking deck.

This is why, an optimisation based path planner specialised on narrow environments with low computational costs for real-time applications is proposed. The step length is variable and included in the optimisation problem. Thus, the planner is able to handle longer driving distances in narrow environments without increasing the computing costs. The cost function is designed to guide the path planner towards the target position and does not rely on a specific scenario. The paper is organised as follows: Section 2 introduces the considered car model and environment. Section 3 presents the path planning concept for common parking spots by solving a sequence of static optimisation problems. Section 4 is devoted to an extension of the parking algorithm to handle longer driving distances, as they typically appear on a parking deck, by calculating a tree of landmarks. Simulation studies for different parking scenarios are carried out in Section 5 showing the practical feasibility of the proposed algorithm. Section 6 contains some conclusions of the presented path planner. This paper extends our previous work [27] giving insight into the choice of the parameters and improving the convergence of the path planner. Additionally, a tree-based extension is proposed to guide the local path planner for longer driving distances. Furthermore, quantitative results of the convergence are provided by extensive Monte-Carlo simulation studies and compared to different state-of-the-art path planners.

2. Problem statement

Three common parking scenarios, namely parallel, garage, and angle parking, constitute the starting point of this paper. These scenarios and the corresponding notations are shown in Fig. 1, where the shaded polygons represent obstacles like, e.g., other cars. The lines to the left and right side of each scenario are boundaries which shall not be violated, e.g., the side walk or the lane separator of the street. In a further step, car parking in more general environments as, for instance, a parking deck and a parking lot, will be considered.

2.1. System dynamics

In the following, the car model with Ackerman steering, as depicted in Fig. 2, serves as a basis for the mathematical description of the car. Thereby, the tyre slip angle is neglected, which is