Smooth path and speed planning for an automated public transport vehicle

Jorge Villagra *, Vicente Milanes, Joshué Pérez, Jorge Godoy

AUTOPIA Program, Centre for Automation and Robotics (CAR, UPM-CSIC), Carretera de Campo Real, km 0.200, 28500 La Poveda, Arganda del Rey (Madrid), Spain

A R T I C L E   I N F O

Article history:
Received 17 May 2011
Received in revised form 26 October 2011
Accepted 4 November 2011
Available online 20 November 2011

Keywords:
Path planning
Speed planning
Automated vehicles

A B S T R A C T

This paper presents a path and speed planner for automated public transport vehicles in unstructured environments. Since efficiency and comfort are two of the key issues in promoting this kind of transportation system, they are dealt with explicitly in the proposed planning algorithm. To that end, a global path planner has been designed with bounded continuous curvature and bounded curvature derivative to ensure smooth driving. This will allow the public transport system to know a priori which is the shortest path within a selected area that guarantees lateral accelerations and steering wheelspeeds below given pre-set thresholds. A closed-form speed profiler uses semantic information provided by the path planner to set a continuous velocity reference that takes into account not only bounds on lateral and longitudinal accelerations consistent with comfort, but also a bound on longitudinal jerk. The suitability of the above two features was compared to manual driving in a real instrumented public transport vehicle on a test track.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

The growth of private car use involves increased carbon dioxide emissions and frequency and severity of traffic jams, especially in urban environments. Indeed, only 21% of EU27 citizens use public transport as their main mode of transport, with the irregularity of the schedules being the main reason for its not being a more popular option in towns (cf. [2]).

Given this context, the EU decided in 2009 to adopt an action plan for urban mobility in which public transport, and specifically the use of electric vehicles, is the main key to improving urban mobility [3] because of their efficiency, lack of CO₂ emission, safety, and comfort. In this connection, fully automated passenger vehicles are expected to soon be extensively used in urban areas, airport terminals, hospital complexes, university campuses, and pedestrian zones. While these electric public transport vehicles will be given traffic priority to encourage citizens to use them, this will not be sufficient to ensure the success of these systems. It will also be necessary to appropriately select the route for the vehicle to cover to improve efficiency and reduce users’ waiting and travel times. To this end, both an accurate and robust control system and a time-optimal path selection algorithm are essential for full advantage to be taken of these transportation systems.

However, although motion planning for autonomous vehicles has been studied in some detail over the last few decades, the requirement of producing trajectories with bounded lateral and longitudinal accelerations and jerk has received little attention in the literature. The approach to solving this problem taken here is twofold: we describe on the one hand a suboptimal path planner that is capable of providing the most efficient route consistent with given comfort and safety requirements, and on the other, an automatic speed profile generator that takes into account the previously planned path and some additional drivability constraints.

1.1. State of the art

Path planning for autonomous robots or vehicles has been extensively studied in recent years to meet a variety of kinematic, dynamic, and environmental constraints. General techniques to obtain optimal trajectories can be grouped into two categories: indirect and direct. Indirect techniques discretize the state/control variables, and convert the trajectory problem into one of parameter optimization which is solved via nonlinear programming [4,5] or by stochastic techniques [6,7]. The latter use Pontryagin’s maximum principle (PMP) and re-express the optimality conditions as a boundary value problem [8].

The complex and incomplete nature of most indirect techniques has motivated the development of optimal control techniques using PMP solutions. These have been based on Dubins’ pioneering work [9] which presented the first set of paths (with straight line segments and arcs of circles) constrained to go from point to point with given initial and final orientations in a minimal
time. In this context of path planning in a free environment, there have been several extensions of Dubins’ result. Reeds and Shepp [10] generalized it to a forward/backward moving car, and other approaches have been proposed to solve time-optimal path-planning problems with linear/angular velocity bounds using line segments and circles (cf. e.g., [11–15]).

A first analysis of these PMP results suggests that the resulting maneuvers are “bang–bang” trajectories, and therefore that, when implemented on real wheeled robots, there would be at least one discontinuity in the path’s curvature profile at circle–segment transitions. To avoid this problem, several studies have been aimed at obtaining “nearly time-optimal paths” which correspond to smoother trajectories than those provided by Dubins’ curves.

Kanayama and Hartman [16] showed that using cost functions such as the integral over the square of either the curvature itself or the curvature’s derivative leads to concatenations of clothoids or of cubic spirals, respectively.

In this connection, Fraichard and Scheuer [17] uses the set of optimal curves described in Sussman’s paper (straight line segments, arcs of circles, and clothoids) to implement an algorithm based on families of Dubins’ curves, modifying simple turns into continuous-curvature turns (with the aid of the clothoids). With this strategy, the curvature profile along the path is continuous and trapezoidal in shape. However, the path presents discontinuities in the steering wheel angular velocity in each transition between circular arc and straight line segment generated by clothoids. Various workers interested in path planning for very high speed vehicles have therefore proposed alternative fundamental curves, namely:

- Curves whose coordinates have a closed expression: $B$-splines [18], quintic polynomials [19], polar splines [20], cardioids [21], $C^2$-splines [22,23], $\eta^3$-splines [24], and Bézier curves [25].

- Curves whose curvature is a function of their arc length: clothoids [26], Kanayama & Hartmann curves, and intrinsic splines [27].

Even though some of these approaches provide greater smoothness than the solution given by Fraichard and Scheuer [17], most make use of an optimization algorithm whose completeness and topological admissibility is not always guaranteed. Moreover, since their main concern is smoothness, the resulting paths are sometimes far from time optimal.

One of the interesting features of [17] is that not only is the curvature bounded but also the curvature derivative is constrained as pre-set by the user. This allows one to compensate the derivative continuity limitation for vehicles running at a moderate speed such as in urban public transport systems. In this connection, this path planner may explicitly adapt the vehicle’s technological (maximum steering wheel speed) and comfort (maximum lateral acceleration) constraints to its design parameters.

Furthermore, there has as yet been little attention paid to combining smooth path planning with obstacle avoidance. Work which has addressed this issue includes “elastic band” [28] and “rapidly-exploring random tree” [29] techniques in successfully generating paths for mobile robots or vehicles with obstacle avoidance and dynamic planning. However, these techniques share two main drawbacks: the shortest path (or a solution close to it) cannot always be guaranteed, and the computational burden is not re-usable to search for time-optimal paths within the selected area.\(^3\)

Finally, since one is not only looking for an adapted path planner but also a speed profile generator, a link between the two tasks would be desirable. In this connection, a semantic interpretation of the path can be useful both to help the driver or co-pilot\(^4\) understand the path to be followed and to reduce the computational cost of calculating the best speed for each situation.

As noted above, smoothness requirements impose constraints on both the path and the speed profile. The speed reference is commonly assumed to be continuously differentiable, and is often designed by optimizing an appropriate performance index (minimum time is the commonest criterion, but minimum jerk has also been used [32]).

Even though several interesting solutions have been presented in the literature (e.g. [33–35]), they all suffer from the same problem. Since a topological semantic map is not used in any case, iterative or optimization processes are required to satisfy a certain number of driving comfort constraints – maximum speed, longitudinal and lateral acceleration, and jerk.

In this connection, smooth trajectory generation has been one of the main concerns for accurate robot control. Thus, the studies of [36–38] present closed-form solutions – via different families of polynomials – to plan a speed profile guaranteeing upper limits on accelerations and jerks.

The present work describes a global solution both to the smooth path planning problem for autonomous vehicles in unstructured environments,\(^5\) and to the generation of an optimal speed profile for the resulting path. The solution to the former problem is based on the combination of the optimal continuous curvature path planner proposed by Fraichard and Scheuer [17] to deal with obstacle-free environments, and the probabilistic roadmap for path planning introduced by Kavraki et al. [30] that is responsible for efficiently finding the minimum number of landmarks necessary to cover an obstacle-free space. The speed profile is determined from the information provided by the path planner and an adaptation of the polynomial-based smooth trajectory generation of [37].

1.2. Contribution

With the above premises, the main contributions can be summarized as follows:

- A global path planner with continuous bounded curvature and curvature derivative aimed at providing smooth driving. This planner will allow the public transport system to know a priori which is the shortest path within a selected area that is consistent with a maximum lateral acceleration and steering wheel speed. Moreover, once the global planning learning phase has been completed, any pair of start and end points of the selected circuit will be quasi-instantaneously connectable by the path planner.

- A closed-form speed profiler, intelligently combined with the path planner, will provide a continuous velocity reference that takes into account comfort constraints not only on lateral and longitudinal accelerations, but also on longitudinal jerk.

- The suitability of the above two features was evaluated with a real instrumented public transport vehicle on a test track. To that end, several manual driving tests were compared with the planning algorithm in terms of time and comfort.

---

\(^3\) A clothoid is a curve whose curvature is a linear function of its arc length $s$ such that $\kappa = a/s$, where $a$ is a real constant called the sharpness of the clothoid.

\(^4\) The term co-pilot has been introduced into the AUTOPIA Architecture for Automatic Driving [31] to refer to the intelligent decision system that chooses from among the set of available maneuvers.

\(^5\) A structured environment in this sense is an area for which a geometrical road network description is available, while for unstructured environments such as parking lots, loading zones, and off-road areas, no such semantic description is available.
The remainder of the paper is structured as follows. Section 2 presents the formalism of the problem that is to be tackled, and gives an overview of the solution adopted which is, as mentioned above, based on a global planner that makes use of a continuous and bounded curvature metric. These characteristics of the local path planner will be highlighted in Section 3, and the global planning algorithm will be described in 4. Some of the results will be compared with real driving tests and discussed in Section 6. Finally, Section 7 presents some concluding remarks.

2. Statement of the problem

Given a start and end point configuration, the specific path and speed planning problem we face is to find a path $\Gamma$ and a speed profile $\Sigma$ such that:

1. $\Gamma$ satisfies the extreme point conditions of $\Gamma_0$ and $\Gamma_e$.
2. $\Gamma$ respects local path planner constraints, i.e., kinematic and dynamic vehicle constraints, as will be detailed in the local path planner section.
3. $\Gamma$ is obstacle free, i.e., it is entirely included in the free configuration space which will be assumed static.
4. $\Sigma$ respects dynamic comfort constraints (maximum speed, acceleration, and jerk).
5. $\Sigma$ and $\Gamma$ provide the time-optimal trip solution — if one exists — connecting the extreme points, while respecting the above constraints.

With these premises, a test-bed circuit was selected on the facilities of the Centre for Automation and Robotics to evaluate the proposed algorithm. There are two types of test track in these facilities (see Fig. 1): a typical urban area with straight stretches, U-turns, and a roundabout that has a very precise and semantically interpretable geometry; and a more generic and not easily classifiable (i.e., an unstructured) circuit. The present work focuses on path and speed planning for the latter, which is represented by circuit $C_1$, in blue in Fig. 1. In order to generalize the results for such a problem, a more complete circuit $C_2$ (red path in Fig. 1) has also been evaluated to reproduce a real urban scenario faced by public transportation systems. As can be observed, circuit $C_2$ combines the unstructured zone with the roundabout and several U-turns.

2.1. Proposed solution

In the path planning literature, it is usual to solve the problem at hand by decomposing it into two hierarchical stages: a local algorithm computes a path between two configurations in the absence of obstacles. From that result, a global motion planning scheme (e.g., [30] or [39]) is used to deal with the obstacles and solve the full problem. In these schemes, the local planner is used together with a collision checker to connect pairs of selected configurations.

Fig. 2 depicts how the local and global algorithms co-operate to find a solution to the aforementioned problem. The family of paths used in the present work was based on the results of [17] which employed a straight line segment, circular arc, clothoid metric (upper block of Fig. 2). If the resulting path is not obstacle-free and the initial and final configurations are still unconnected — see configurations 1–3 of the bottom block in Fig. 2 — then a global planner inspired by the Probabilistic Path Planner (PPP) [30] is used to intelligently select a new intermediate configuration. One of the main features of the proposed planner is that this phase of learning the obstacle-free configuration space will be available for any future query within the same area.

Finally, the semantic map provided by the local planner will be exploited to simplify the generation of the optimal speed profile. To this end, we adapted work by Liu [37] to our specific trajectory planning — see the block on the right of Fig. 2 — imposing the longitudinal maxima of speed, acceleration, and jerk on the vehicle’s motion for stretches in which there are no turning lateral dynamics.

3. Local planner

The car-like mobile robot considered has a body frame attached to a reference point with coordinates $\xi = (x, y)^T$ in a global
coordinate frame, and orientation $\theta$ with respect to the global frame. The generalized coordinates of this wheeled mobile robot are therefore $(x, y, \theta)$. The vehicle is equipped with a front-centered steerable wheel (whose angle is represented by $\phi$) and fixed parallel rear wheels that move with velocity $v$. With the assumption of small velocities, the dynamics of a car having a wheelbase of length $L$ can be described by

$$
\begin{bmatrix}
\dot{x} \\
\dot{y} \\
\dot{\theta} \\
\dot{\phi}
\end{bmatrix} =
\begin{bmatrix}
v \cos \theta & v \sin \theta & 0 & 0 \\
0 & v & 0 & 0 \\
-\tan \phi & 0 & 1 & 0 \\
L & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
\theta \\
\phi
\end{bmatrix} +
\begin{bmatrix}
0 \\
0 \\
0 \\
0
\end{bmatrix}
\gamma_x
$$

(1)

where $\omega$ is the angular velocity, and $\gamma_x$ is the longitudinal acceleration. Note that the steering angle, steering speed, longitudinal velocity, and acceleration are subject to the constraints $|\phi| \leq \phi_{\text{max}}$, $|\omega| \leq \omega_{\text{max}}$, $v \leq V_{\text{max}}$, $|\gamma_x| \leq \gamma_{x\text{max}}$, respectively. As shown in [40], the kinematic model of the wheeled mobile robot given by the three first equations of (1) is flat with flat outputs $\{x, y\}$. As a consequence of this, any state or input variable can be expressed in terms of the $\{x, y\}$ and their derivatives:

$$
\begin{align*}
\theta(t) &= \arctan\left(\frac{\dot{y}(t)}{\dot{x}(t)}\right) \\
v(t) &= \sqrt{\dot{x}^2(t) + \dot{y}^2(t)} \\
\phi(t) &= \arctan\left(\frac{L(\dot{x}(t)\dot{y}(t) - \ddot{x}(t)\dot{y}(t))}{\dddot{x}(t) + \dddot{y}(t)}\right).
\end{align*}
$$

(2)

We shall use that the arc length $s_t$ of a planar curve $\{x, y\}$ can be expressed as

$$
f : [u_0, u_1] \rightarrow [0, f(u)],
$$

$$
u \mapsto s_v = \int_{u_0}^{u} \sqrt{\dot{x}(\xi)^2 + \dot{y}(\xi)^2} \, d\xi
$$

and that the scalar curvature is, according to the Frenet formulas, a function of $\{x, y\}$ and their derivatives

$$
k(t) = \frac{\dddot{x}(t)\dot{y}(t) - \dddot{y}(t)\dot{x}(t)}{(\dot{x}(t)^2 + \dot{y}(t)^2)^{3/2}}.
$$

(3)

Consider $\kappa_{\text{max}}$ the maximum curvature and $\sigma_{\text{max}} = \frac{dk}{ds}_{\text{max}}$ the maximum curvature derivative with respect to the arc length. From (2) and (3), it is straightforward to obtain

$$
\begin{align*}
\tan \phi(t) &= L \kappa(t) \Rightarrow \kappa_{\text{max}} = \frac{\tan \phi_{\text{max}}}{L} \quad \text{(4)}
\frac{dk}{dt} = \frac{dk}{ds} \frac{ds}{dt} \Rightarrow \sigma_{\text{max}} = \frac{\dot{\phi}}{s_L \cos^2 \phi}_{\text{max}}. \quad \text{(5)}
\end{align*}
$$

**Proposition 1.** A path $\Gamma$ in the $(x, y)$ plane is generated by model (1) with input $\omega(t) \in \mathbb{E}^1$, $\omega \in [-\omega_{\text{max}}, \omega_{\text{max}}]$, and $V(t) \in \mathbb{E}^1$, $V \in [0, V_{\text{max}}]$. $\forall t \geq 0$, if and only if its curvature $\kappa \in \mathbb{E}^1$, $\kappa \in [0, \kappa_{\text{max}}]$ and its curvature derivative $\sigma \in [-\sigma_{\text{max}}, \sigma_{\text{max}}]$.

A sketch of the proof follows. Given any $C^2$-curve $p(u)$ with $u \in [u_0, u_1]$, the inverse arc length function $s^{-1}_u$ is defined and is, by definition, a continuous function. Moreover, the scalar curvature $\kappa(u)$ is also continuous over $[u_0, u_1]$ because $p(u)$ is a $C^2$-curve. Consider the curvature bound to be such that the steering angle is $|\phi| \leq \phi_{\text{max}}$. At the initial time $t_0$, consider the state of model (1) given by $[x(u_0), y(u_0), \theta(u_0)]$. Then, applying the continuous input $\omega(t) = (v(t)\kappa(s^{-1}_u(v(t - t_0))))$.

(6)

the vehicle’s motion from $t_0$ to $t_0 + u s^{-1}_u$ exactly matches the path of the given curve if $|\omega| \leq \omega_{\text{max}}$, $v \leq V_{\text{max}}$, $|\gamma_x| \leq \gamma_{x\text{max}}$. Moreover, if $k(u) \in \mathbb{E}^1$ and $v(u) \in \mathbb{E}^1$, then $\omega(u) \in \mathbb{E}^1$.

**Remark 1.** Although model (1) satisfactorily represents a wheeled mobile robot’s behavior, public transport vehicles driving in urban areas exhibit nonlinear dynamics and are affected by many disturbances, such as turning and static friction, or variations in the vehicle’s weight. Since these poorly known effects are difficult to model and include in the path planner, a robust, non-model-based, control system (e.g. [41] or [42]) should be used to precisely track the resulting path.

**Remark 2.** Note that the lateral acceleration $\gamma_x$ of a planar non-slipping vehicle is equal to $v^2 \dot{\theta}$ (cf. [43]), which combined with (2) and (3), and considering small turning radius yields $\gamma_x = v^2 \kappa$.

As mentioned above, Friaichard and Scheuer [17] propose an efficient solution that generates suboptimal paths with an upper bounded continuous curvature and an upper bounded curvature derivative. The resulting path will consist of straight line segments, circular arcs, and clothoids, which will be automatically combined to connect configurations with null curvature. Even if the algorithm...
Fig. 3. (a) Path between starting configuration A and final configuration B with $k_{\text{max}} = 0.8$ and $\sigma_{\text{max}} = 0.3$; (b) curvature versus arc length.

The exploration of the environment is done by successively adding to the graph a random configuration $p_i$ — a data set consisting of $x_i, y_i, \theta_i$ — and trying to directionally connect this configuration to a maximum number of nodes of the graph with the local path planner.

As the aim of the planner is to find the shortest possible path, the graph weight $f$ will be the Fourier transform of the path curvature — in other words, a measure of the frequency of the steering angle motion. This weighting variable was chosen instead of the classical path length criterion because the configuration space is too narrow to allow any considerable reduction in path length so it seems more reasonable to maximize the passengers’ comfort.

We modified the original path planner [30] slightly in order to achieve better graph connectivity over complicated access zones. To that end, two main refinements were introduced into the initial algorithm:

- Increase the probability of generating intermediate points in sensitive areas by assigning priorities related to the search zones geometric characteristics, i.e., $P_z \in [0, 1]$, with 1 being a critical zone and 0 a very easily connectable zone. To automatically set the value of $P_z$, a fuzzy decision system, based on the curvature, the surrounding curvature and configuration space width of the node, has been implemented. Fig. 4 shows how from a curvature $\kappa$, curvature integral $\Delta \kappa$ and width $W$ mappings of the road, a fuzzy function $P_z = f(\kappa, \Delta \kappa, W) \in [0, 1]$ is inferred for the whole selected circuit.

- Decrease the probability of obtaining intermediate points in a given area — characterized by a ball of center $p_i$ and radius $r B_{r_i}(p_i)$ — when many configurations $P_i = [x_i, y_i, \theta_i]$ have already been tried in this area, and the connectivity rate is sufficiently high. To that end, the following function $P_i \in [0, 1]$ is used

$$P_i = \max \left(1 - \sum_{j=1}^{n} \frac{k_d}{\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}}, 0 \right)$$

$$p_1 \ldots p_n \in B_{r_i}(p_i).$$

Thus, for a new node $p_i$, the local planner tries to connect it with each node belonging to the graph whose weight $f$ is less than a pre-set value $f_{\text{max}}$. However, if the resulting associated value $P_i = P_1 + P_2$ is lower than the most adapted heuristic value $P_i = 1$, the new node $c$ is not included in the graph, and a new node is generated until it satisfies the previous condition. This process continues until a graph connectivity condition has been fulfilled (for instance, to obtain at least one solution to a specific query) and a fixed number of nodes has been reached. This selective process leads to a significant reduction in the number of intermediate points needed to attain a given degree of connectivity (Fig. 5 and Table 1). Note in the figure that,
while with purely random configurations there are two large zones that remained unconnected, the prioritized algorithm provides much better connectivity with the same number of nodes. The results for this specific example can be quantified and generalized, as shown in Table 1, with the mean results obtained after 10 different learning phases for each one of the two circuits - $C_1$ and $C_2$ - presented in Section 2. To quantify the enhancements provided by the proposed refinements, two different metrics have been introduced: the Graph Density ($GD$) and the Unconnected Nodes Ratio (UNR).\(^{10}\) Both indexes are separately evaluated for nodes whose $P_z$ values is lower than 0.5 — easy access zones — and higher than 0.5 — complicated access zones. Note that the refined algorithm not only provides better results in terms of density, but it also significantly increases the UNR values over complicated zones, which allows the algorithm to improve connectivity with relaxed curvature constraints.

As can be appreciated in Algorithm 2, if both of the above conditions are fulfilled, the learning phase ends. Otherwise, a connectivity refinement process is initiated. To this end, forward and backward unconnected nodes are automatically detected, and connections of the latter with the closest nodes in configuration space are sought by iteratively using different curvature and curvature derivative bounds.

As mentioned in the introduction, one of the interesting features of this solution is that this learning phase can be done offline, because it only depends on the environment, not on the query configurations.

In the query phase, given start and final configuration are connected to the two closest nodes of the graph using the local planner, and then a graph search is performed between these two nodes. This step is carried out with the aid of the Dijkstra algorithm\(^{44}\) which finds the least-cost (the smoothest in the present case) path from a single source node to all other nodes in a weighted and directed graph.

---

\(^{9}\) The density of a graph is the ratio of the number of edges and the number of possible edges.

\(^{10}\) UNR is defined as the number of unconnected nodes in a particular region with respect to the overall number of unconnected nodes in the graph.
Fig. 6. Global path planner refinements: (a) plot of nodes and links representing the adjacency matrix; (b) feasible paths; (c) curvature profile of the feasible paths; (d) selected curvature profile. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Algorithm 2 Learning algorithm of the global path planner.

1: while $c = 0$ | $i < n_{\text{max}}$ do
2:     while $P_t(p_i) < = P_c$ do
3:         $(p_i, P_t(p_i)) =$ random configuration
4:     end while
5:     for $j = 1; j <= \text{length}(A_{\text{adj}})$ do
6:         $(A_{\text{adj}}, A_{\text{Coord}}) =$ local path planner($p_j, p_i, \kappa_0, \sigma_0$)
7:     end for
8:     $i := i + 1$
9: end while
10: while $c = 0$ & $i = n_{\text{max}}$ do
11:     $(\kappa_k, \sigma_k) =$ new curvature constraints
12:     $p_l =$ find unconnected nodes($A_{\text{adj}}, A_{\text{Coord}})$, $k = 1 \ldots L$
13:     for $l = 1; l <= L$ do
14:         for $m = 1; m <= \text{length}(A_{\text{adj}})$ do
15:             $(A_{\text{adj}}, A_{\text{Coord}}) =$ local path planner($p_l, p_m, \kappa_k, \sigma_k$)
16:         end for
17:     end for
18: end while

Note that, according to its authors [30], this planner is probabilistic complete. This means that any problem which can be solved using an overall path made up of several elementary paths (generated by the local planner) will be solved, provided that the exploration is carried out for a sufficient amount of time. Moreover, [45] proved that under certain geometric assumptions on the free configuration space — which are fulfilled for this work, a link between the expected running time and the size of the graph using PPP can be made. Specifically, the expected size of a probabilistic roadmap required for solving this problem grows only logarithmically in the complexity of the problem. Moreover, the number of nodes required to be generated, in order for the planner to succeed with probability at least $1 - \alpha$ is logarithmic in $\frac{1}{\alpha}$ and polynomial in the inverse of the clearance — the minimal distance between the path and the obstacles configuration space.

In the case that a path with bounded curvature and curvature derivative cannot be found, the algorithm was designed to enhance the graph’s connectivity by looking for intermediate paths with more permissive constraints.

Fig. 6 shows the effects of these refinements in a case when the curvature constraints are too demanding for the obstacle-free configuration space. In the upper left plot, the graph with its nodes and edges is represented in the $x$-$y$ configuration space. Note that, with the initial curvature constraints, there is an inaccessible part at the bottom of the graph — remark that if the red nodes were removed, the remaining blue nodes, which represent the
paths with $\kappa_{\text{max}} = 0.08$, would not permit us to complete the circuit. However, the second part of the Algorithm 2 (lines 11–19) provides much better connectivity by relaxing the maximum constraint bound for the unconnected nodes — red nodes. This is also observed in the upper right and lower plots of the figure in which the feasible trajectories and the associated curvature profiles are shown in blue or red depending on whether the maximum curvature is the initial value — $\kappa_{\text{max}} = 0.08$ — or has been relaxed — the curvature bound is a fixed value between 0.08 and the inverse of minimum curvature radius of the vehicle, which is around 0.2.

5. Speed profile generation

As mentioned above, the concatenation of continuous curvature paths provides a semantic interpretation of the overall trajectory. In other words, any path in an unstructured environment can be decomposed, with the help of the proposed path planning algorithm, into a succession of turns — composed of clothoids and arcs of circles — and straight lines. This result is extremely useful in finding closed-form optimal speed profiles because both straight line segments and circle arcs can be associated with constant speeds. More precisely, when a turn is initiated the maximum velocity will be constrained by the comfort lateral acceleration threshold, and when a straight segment is being tracked, the maximum longitudinal speed, acceleration, and jerk will be the limits imposed on the reference speed.

With these premises, the overall speed profile will consist of three families of curves:

1. Constant speed curves at a minimum value $V_{\text{min}}$ when the curvature profile is a circular arc or its preceding clothoid. The minimal speed is fixed by the curvature bound for the curve $\kappa_{\text{max}}$, i.e., by combining (1) and (3) to yield $V_{\text{min}} = \sqrt{\gamma_{\text{f}} / \kappa_{\text{max}}}$, where $\gamma_{\text{f}}$ is the maximum lateral acceleration.
2. A smooth transition from the minimum value $V_{\text{min}}$ to a maximum allowed speed $V_{\text{max}}$ and back again to $V_{\text{min}}$ that fulfills the acceleration and jerk constraints.
3. A set of one or two smooth transition curves (of type 2 above) that go from zero to the maximum speed, and vice versa. This passage will be performed with a single curve if no turns are necessary in the corresponding path. Otherwise, a straight line segment will be introduced between two transition speed curves whose length will be determined by the turn deflection.

Fig. 7 shows the concatenation of curvature profiles between nodes whose positions are represented by red dots, and the associated speeds, accelerations, and jerks.

In order to obtain closed-form expressions for the second type of curve, various smooth trajectory generation techniques were evaluated. Of those in the literature on robot manipulators, that of [37] not only provides a deterministic optimal speed fulfilling the length and comfort constraints — bounds on speed, acceleration, and jerk — but also allows the speed profile to be regenerated on-line in case a dynamic obstacle is detected so that the path planner can take it into account.

The speed trajectory is divided into seven intervals $[t_{i-1}, t_i]$, $i = 1, \ldots, 7$ (see Fig. 8). This piecewise function can be expressed in terms of the arc length $s_i$ as follows:

$$s_i(t) = \begin{cases} 0 & t \in [t_0, t_1] \vee t \in [t_5, t_6] \\ \frac{\bar{s}_i}{\kappa_{\text{max}}} & t \in [t_1, t_5] \vee t \in [t_6, t_7] \\ \frac{\bar{s}_i}{\kappa_{\text{max}}} & t \in [t_1, t_5] \vee t \in [t_6, t_7] \end{cases}$$

$$\dot{s}_i(t) = \bar{s}_i(t_{i-1}) + \bar{s}_i(t)(t - t_{i-1})$$

$$\ddot{s}_i(t) = \bar{s}_i(t_{i-1}) + \bar{s}_i(t_{i-1})(t - t_{i-1}) + \frac{1}{2} \bar{s}_i(t)(t - t_{i-1})^2$$

$$\dddot{s}_i(t) = s_i(t_{i-1}) + \ddot{s}_i(t_{i-1})(t - t_{i-1}) + \frac{1}{2} \dddot{s}_i(t)(t - t_{i-1})^2$$

The speeds, accelerations, and jerks in each stretch depend upon the initial conditions $s_i(t_0) = s_i(t_0)$, $\dot{s}_i(t_0) = \dot{s}_i(t_0)$, $\ddot{s}_i(t_0) = \ddot{s}_i(t_0)$, the final conditions $s_i(t_f) = s_i(t_f)$, $\dot{s}_i(t_f) = \dot{s}_i(t_f)$, $\ddot{s}_i(t_f) = \ddot{s}_i(t_f)$, and the comfort constraints $|\dddot{s}_i(t)| \leq \dot{s}_{\text{max}}, \bar{s}_i(t) \leq \kappa_{\text{max}}, |\dddot{s}_i(t)| \leq \dot{s}_{\text{max}}$.

The analysis given in [37] establishes four different cases depending on the boundary conditions, but the present work will consider only the most general case: when all seven intervals plotted in Fig. 8 exist, i.e., the maximum speed, acceleration, and jerk are attained.

In particular, the arc length will go from the initial point of the closing clothoid in a turn ($s_{\text{f}}$) to the final point of a straight line segment ($s_{\text{i}}$), the initial and final speeds ($\dot{s}_{\text{i}}$, $\dot{s}_{\text{f}}$) will be set...
by a minimum speed \( V_{\text{min}} \), and the initial and final accelerations \((\dot{s}_0, \ddot{s}_0)\) and \((\dot{s}_f, \ddot{s}_f)\) will both be equal to zero. Concerning the comfort constraints, the maximum speed \( s_{\text{max}} \) will be \( V^*_{\text{max}} \) and the maximum acceleration will be determined by the design parameters \( \gamma_{\text{max}} \) and \( f_{\text{max}} \).

Note that the value of \( V^*_{\text{max}} \) corresponds to the \( V_{\text{max}} \) previously defined if there is enough distance to reach the target. If the available arc length is less than some critical value, the maximum speed will be set equal to the initial speed \( V_0 \) resulting in the generation of a constant speed profile. Otherwise, a maximum speed between \( V_0 \) and \( V^*_{\text{max}} \) will be computed. The closed form polynomial expression of Eq. (7) permits the maximum speed to be computed as follows:

\[
V^*_{\text{max}} = \begin{cases} 
V_{\text{max}}, & \text{if } \Delta_i > (V_{\text{max}} + V_0) \sqrt{\frac{V_{\text{max}} - V_0}{f_{\text{max}}}} \\
+ (V_{\text{max}} + V_0) \sqrt{\frac{V_{\text{max}} - V_0}{f_{\text{max}}}}, & \text{if } V_0, \text{ if } \Delta_i < V_0/2 \ast (V_0/\gamma_{\text{max}} + \gamma_{\text{max}}/f_{\text{max}}) \\
-\frac{1}{2f_{\text{max}}} (\gamma_{\text{max}} \sqrt{V_{\text{max}} - V_0} \Delta_i + 4\gamma_{\text{max}}^2 V_0^2 - 4\gamma_{\text{max}}^2 f_{\text{max}} V_0), & \text{else} 
\end{cases}
\]  

where \( \Delta_i = s_i - s_0 \). The interested reader can obtain further details in [37] about the generation of (8) for the specific case when \( s_0 = 0 \), that has been generalized for any final speed \( s_f \).

Since this is a rather conservative approach, an alternative algorithm was implemented to reduce the overall time needed to cover the planned path by slightly compromising the passengers’ comfort. Instead of reducing the speed to \( V_{\text{max}} \) at each turn, only the non-degenerate turns are taken into account for this purpose. Thus, if the curvature profile presents a concatenation of symmetric clothoids, the speed planner will not interpret this as a turn. Therefore, the speed planner will provide greater speeds in zones in which some degenerate turn is present in the overall path than would have been given by the algorithm presented above. However, the bounds on the lateral acceleration will be briefly exceeded in these degenerate turns. To quantify these effects, both speed planning approaches have been tested in circuits \( C_1 \) and \( C_2 \) and a discussion about the obtained quantitative comfort results is presented in Section 6.

6. Experimental results

6.1. Instrumented public transport vehicle

Molinero is a model EGK6152K electric minibus (Fig. 9) with capacity for 14 passengers. It is manufactured by Con-Auto. This vehicle has four arrays of four 12 V batteries each, and a 5 kW electric motor that allows it to reach 10 m s\(^{-1}\) with a maximum longitudinal acceleration in first gear of around 2 m s\(^{-2}\). The steering wheel has a maximum angle of 37° and a maximum speed of around 70° s\(^{-1}\).

The vehicle has been instrumented with a Differential Global Positioning System (DGPS) installed at the vehicle’s rear-center, which allows the vehicle to sense its position and velocity at a 10 Hz rate. An inertial measurement unit (IMU) is located as close as possible to the center of gravity of the minibus to provide, on the one hand, back-up positioning in case of GPS receiver failures, and, on the other, data on the longitudinal and lateral acceleration of the vehicle. The architecture’s backbone is an On-board Control Unit which consists of a solid-state industrial computer for automotive applications. The solid state components allow the computer to work normally under driving conditions, avoiding the commonly encountered hard-disk failures due to vibrations.

6.2. Path planning in a real scenario

Note that the obstacle-free configuration space used in the learning phase can be significantly reduced in this specific case. Thus, the search over \( x_i \) and \( y_i \) will follow three basic steps [46]. First, the boundaries of the polygonal obstacles are approximated by subdividing each side of the original polygon into small segments. Second, the Voronoi diagram is computed for the resulting points. And third, the Voronoi edges which have one or both end-points lying within any of the obstacles are eliminated.

Once the Cartesian coordinates of the reduced obstacle-free points have been determined, their orientations \( \theta_i \) are found using a non-slipping hypothesis (\( \theta_i = \arctan\left(\frac{\frac{y_i}{x_i}}{\frac{x_i}{y_i}}\right) \)), where \( x_i \) and \( y_i \) are the coordinates of the closest position in the reduced search space.

In general, a reasonably good approximation can be made for the bounds on the curvature and curvature derivative. Substituting the comfort constraint \( \gamma_{\text{max}} = V^*_{\text{max}} \kappa \leq \gamma_{\text{max}} \) and the maximum curvature expression (5) in (4), the maximum value of the curvature derivative can be estimated as

\[
\sigma_{\text{max}} = \frac{\phi (\tan \phi)^{1/2}}{\gamma_{\text{max}}^{2/3} \cos^2 \phi}.
\]

From the bus turning constraints, Eq. (9), and a maximum lateral acceleration of \( \gamma_{\text{max}} = 1 \text{ m s}^{-2} \), one finds for the maximum curvature and curvature derivative in this specific case \( \kappa_{\text{max}} = 0.189 \) and \( \sigma_{\text{max}} = 0.207 \), respectively.

In order to obtain the parameters \( \kappa_{\text{max}}, \sigma_{\text{max}}, \) and \( n_{\text{max}} \) best adapted to our problem, a set of planning tests with Algorithm 2 was applied to the above configuration space.
Fig. 9. Electric minibus used in the experimental phase.

Fig. 10. (a) Influence of the number of nodes on the cost function; (b) influence of the curvature and curvature derivative on the cost function.

We first looked for a reasonable compromise between optimality and computational cost by running the path planner algorithm with increasing number of nodes \( n_{\text{max}} \in [50, 190] \). Since the global path planner obtains the intermediate points probabilistically, each fixed-size graph was computed five times to obtain meaningful results. One observes in the box plot of Fig. 10(a) that, while for \( n_{\text{max}} = 50 \), 70 there was at least one run which did not obtain a solution, for the interval \( n_{\text{max}} \in [90, 190] \) the cost function box plots varied only slightly. The zoomed-in inset shows the influence of the number of nodes on the total arc length of the planned paths, and is intended to highlight that in this latter interval the arc length is hardly any different (only by some 2%) between the worst and the best cases of 90 and 190 nodes. It thus seems reasonable to consider \( n_{\text{max}} = 90 \) as the most efficient size of the adjacency matrix for our purposes.

In order to evaluate the influence of each parameter on the path planning process, an exhaustive set of tests was performed with \( \kappa_{\text{max}} \in [0.08, 0.2] \), \( \sigma_{\text{max}} \in [0.05, 0.2] \). Note that the upper limits of these intervals are approximated from the previously computed values, while the lower limits are from the empirical results on the configuration space considered. The surface plotted in Fig. 10(b) shows the influence of the curvature and curvature derivative constraints on the cost function — the curvature Fourier transform median.

Remark 3. FFT is an efficient algorithm to compute the Discrete Fourier Transform (DFT) \( \mathcal{F} \)

\[
K_j = \mathcal{F}(\kappa_j) = \sum_{i=0}^{N-1} \kappa_j e^{-\frac{2 \pi ji}{N}}, \quad j = 0 \ldots N-1
\]

where \( \kappa_j \) is the curvature at sample \( j \) and \( N \) is the length of the curvature signal. It is well known that DFT allows us to analyze the frequency spectrum of a sampled signal, and as a consequence, its sharpness. A statistical indicator will be taken from the curvature: the median \( \tilde{\kappa} \) of sequence \( K_j \)

\[
P(K_j < \tilde{\kappa}) \geq \frac{1}{2} \wedge P(K_j > \tilde{\kappa}) \geq \frac{1}{2}
\]

to have a good indicator of the overall curvature evolution while giving reduced importance to outliers.

Fig. 10(b) confirms that, for a given curvature bound, the chosen criterion is clearly related to the maximum curvature derivative. Thus, the maximum curvature derivative will be \( \sigma_{\text{max}} = 0.05 \), and,
Fig. 11. Comparison of human drivers’ behavior with the theoretical dynamics in circuit $C_1$: (a) paths; (b) speeds; (c) total acceleration. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

since there is no solution to the problem for $\kappa_{\text{max}} \leq 0.1$, this value is the constraint chosen for the curvature.

6.3. Manual driving versus path and speed planning

Several trials were performed with the public transport vehicle described above driven manually. The start and end points ($\Gamma_0 = \Gamma_e$) were the same as in our path planning problem, and the different drivers were requested to choose the route such that turns, accelerations, and braking provided a sensation of comfort to the vehicle’s occupants while minimizing trip time. With the aim of covering a large spectrum of driving attitudes, 15 people have been selected for such a task, who can be classified in three different categories: Slow Drivers (SD), Medium Drivers (MD) and Aggressive Drivers (AD). In the following, these three driver typologies will be compared with the two variants of our Path and Speed Planning (PSP) algorithm.

Remark 4. From Proposition 1, a vehicle that can be modeled by Eq. (1) is able to track the path generated by Algorithm 2 at speeds recommended by the algorithm detailed in Section 5 if an appropriate control algorithm – e.g. [42] – is used. Consequently, the resulting speeds and accelerations can be compared with those of a human driver trying to respect the maximum speed and comfort acceleration constraints. In other words, even if longitudinal and lateral controllers are not still implemented on the vehicle, it seems reasonable to assume that the resulting path and speed profile can be accurately tracked, and therefore can be compared with a human in the loop driving.

Table 2 and Figs. 11–12 present the most significant variables and indicators with which to evaluate the proposed automatic path and speed planning algorithms against manual driving in circuits $C_1$ and $C_2$, respectively.

Following ISO 2631-1 standard [47], comfort can be quantified by computing the whole acceleration on the human body

$$\gamma_t = \sqrt{\gamma_x^2 + \gamma_y^2 + \gamma_z^2}$$

where $\gamma_z^2$ is the vertical acceleration, which is neglected for our
specific purposes. Since \cite{47} sets the limit to a fairly uncomfortable motion at 1 m s\(^{-1}\), the maximum overall acceleration will be set at \(\gamma_t = 1\) m s\(^{-1}\).

To quantify the comfort all through the circuit, two different indicators are taken into consideration:

- the integral squared difference between the total acceleration of the vehicle and the maximum allowed acceleration

\[ I_{\gamma} = \frac{1}{T} \int_0^T (\gamma_t - \gamma_{\text{max}})^2 \]

which measures the mean trip comfort

- the maximum difference between the total acceleration at any instant of the trial and the maximum allowed acceleration

\[ M_{\gamma} = \max |\gamma_t - \gamma_{\text{max}}| \]

which quantifies the most abrupt turn that the vehicle's occupants feel during the trip.

Table 2 summarizes the mean, maximum and minimum values of these two comfort estimators and of the trip time in each one of the two considered circuits \(C_1\) and \(C_2\). Note that the blue figures highlight the maximum values, while the red ones emphasize those with the worst result.
In Fig. 11(a) a representative path of each driver group lie almost on top of each other, while the result of the path planner plotted in red deviates somewhat from them. In particular, one notes that the planned path is in general similar to those taken by human drivers, especially on the left of the circuit, although there appear some differences at the top right and the bottom of the circuit. These divergences reflect the imposition of a maximum curvature of $\kappa_{\text{max}} = 0.1$ on the algorithm, while the driver intuitively looks for the shortest comfortable path. In Fig. 12(a), the planned path for circuit C2 is also pretty similar to those adopted by most of drivers.

We were interested of course in contrasting the planned speeds provided by the two alternatives detailed in Section 5 with the three manual driving groups speed profiles. The former were obtained with a maximum speed of $V_{\text{max}} = 10 \text{ m s}^{-1}$, a maximum longitudinal acceleration of $a_{\text{max}} = 1 \text{ m s}^{-2}$, and a maximum jerk of $J_{\text{max}} = 1 \text{ m s}^{-3}$.

One notes in Fig. 11(c) that all the manual driving tests exceeded the critical value $V_{\text{crit}} = 1 \text{ m s}^{-1}$ in the interval $s_r \in [170, 230]$ (corresponding to the sharp turn at the bottom part of the circuit), while the PSP approaches respect this critical value to a far greater extent — especially PSP1. In Fig. 12(c) this phenomenon is generalized at each significant turn of circuit C2.

One observes from Table 2 that PSP1 clearly provides the best $I_1$ and $M_1$ results while obtaining a trip time approximatively equal or shorter than that of all the human trials except that driven aggressively. The PSP2 strategy gains 10 s in trip time for circuit C1 — around 12 in circuit C2 by allowing some discontinuities in the acceleration profile. These short duration peaks reflect small deflection turns that should not have any great significance in an automated driving context. In any case, its value of $I_1$ is still at the same level as the medium driver trials.

It is evident from Table 2 — in which the minimum and maximum values are given in blue and red, respectively — that PSP provides very interesting path and speed profiles in the sense that they represent better combinations of trip time and comfort than the human drivers were able to achieve in this unstructured environment.

7. Concluding remarks

A new approach to planning smooth path and speed profiles for automated vehicles in unstructured environments has been presented. It comprises three layers: (i) an optimal local continuous curvature planner for obstacle-free situations; (ii) a global planner that finds intermediate points to connect the configuration space to the desired degree, taking obstacles into account; and (iii) a speed planner that uses the set of curves of the previous layer to compute analytically a comfort-constrained profile of velocities and accelerations. The results of the algorithms were satisfactorily contrasted in an automated public transport vehicle with real driving maneuvers performed by human drivers.

The next natural stage of our work is, once the bus has been fully automated, to include the path planner in an overall control scheme in which an adapted robust control algorithm [41,42] will be used to track as closely as possible both the planned path and the planned speed. We are also interested in further investigating the application of the results reported by Yang and Sukkarieh [25] to the present algorithms in order to improve the smoothness of the curvature profile, and hence reduce the trip time and the breadth of the range of operating speeds.

Acknowledgments

This work was supported by the Spanish Ministry of Science and Innovation through Research Grant TRANSITO TRA2008-06602-C03 and by the Spanish Ministry of Development through Research Grant GUIDE P9/08.

References


