24th Mediterranean Conference on Control and Automation (MED)
June 21-24, 2016, Athens, Greece

A linear-time algorithm for minimum-time velocity planning of autonomous vehicles.

L. Consolini, M. Locatelli, A. Minari, A. Piazzi[1]

Abstract— **Velocity planning on a path to be followed by a wheeled autonomous vehicle may be difficult when high curvatures and velocities are allowed. A fast, straightforward algorithm to address this problem is presented. It has linear-time computational complexity and provides an optimal minimum-time velocity profile. The algorithm is based on a *curvilinear discretization* that makes easy to take into account the constraint on the vehicle's maximal normal acceleration. Formal proofs of the algorithm's properties are included. Two examples illustrate the proposed approach.**

I. Introduction

The design and development of autonomous vehicles is a major technological advancement influencing and shaping our future mobility and quality of life [1]. Intense research is internationally carried out on the various technologies enabling the automated driving of wheeled autonomous vehicles [2]. In this context, focusing on motion planning a significant problem is the *minimum-time* motion of a car-like vehicle from a start configuration to a target one while avoiding collisions (obstacle avoidance) and satisfying kinematic, dynamic and mechanical constraints (on velocities, accelerations, maximal steering angle, etc.). This problem can be then approached as: 1) a *minimum-time trajectory planning* where both the path to be followed by the vehicle and the timing law on this path (i.e., the vehicle's velocity) are simultaneously designed; or 2) a (geometric) *path planning* followed by a *minimum-time velocity planning* on the planned path.

In general, the latter is a sub-optimal approach with respect to the former. On the other hand, the latter, also known as *path-velocity decomposition* [3] is easier to implement in real-time when the vehicle's motion supervisor uses a sensor-based iterative motion planning [4], [5]. To comply with the vehicle's nonholomonic constraints, the planned path must have a certain geometric continuity and satisfy geometric interpolation conditions at its endpoints [5], [6], [7]. In turn, the velocity to be planned must comply with certain kinematic constraints (typically on velocities and accelerations) and interpolating boundary conditions.

An almost time-optimal velocity planning on a given path based on *ternary* polynomials is presented in [8]. A simple solution based on root extraction of a quartic equation is reported in [9] emphasizing arbitrary velocity/acceleration boundary conditions. A more elaborate solution based on *time discretization* is presented in [10]. It uses a sequence of linear programming feasibility tests to obtain a minimum-time velocity profile with constraints on velocities, accelerations and jerks. However, these works do not explicitly consider the constraint on the maximum normal (centripetal) acceleration of the vehicle.

This constraint is central to avoid the vehicle's sideslip especially when high curvatures and velocities may be present. An early work considering this constraint is [11]. An algorithm is then presented using linear profiles for the longitudinal accelerations. Other works addressing this issue are [12], [13], [14], [15]. Solea and Nunes [12] achieve a solution with an algorithm based on the five-splines scheme of [16]. Villagra et al. [13] propose a closed-form speed profiler by using the elementary velocity profiles provided in [17]. The velocity planning of Chen et al. [14] is composed of simple linear or quadratic speed profiles. Lin et al. [15] propose the generation of a velocity profile on the curvilinear frame (i.e., a velocity as a function of the path's arc-length) taking into account maximal normal and longitudinal accelerations and maximal brake decelerations. However, they do not formalize a minimum-time velocity planning problem.

In this paper a minimum-time velocity planning problem on a given path is proposed and solved. All the most relevant constraints are considered, namely: given bounds on maximum velocity; minimum and maximum longitudinal accelerations; and maximum normal acceleration. The interpolating boundary conditions are the vehicle's current velocity at the start of the path and the final velocity to be reached at the path's end.

A key point of the proposed solution is the *curvilinear discretization* on the path for which the sought velocity profile is a function of the arc-length (as in [15]). This discretization leads to a simpler nonlinear optimization problem with respect to the problem that would be obtained with the standard time discretization (cf. Section II). An algorithm that provides a global solution to the problem is then presented (cf. Section III). Remarkably, this algorithm has a computational time that increases linearly with the dimension n of the problem (being n the number of velocity samples on the path). Moreover, the algorithm during its three iterations (cf. (24), (25), (26)) requires very few computations and logical operations so that its implementation in real-time should be greatly facilitated.

Paper organization: Section II reports the motivation of the addressed velocity planning and the formal definition of the posed optimization problem (cf. Problem 1). Solution to

[1]The authors are with the Dipartimento di Ingegneria dell'Informazione, Università di Parma, Parco Area delle Scienze 181/A, 43124 Parma, Italy. E-mails: {luca.consolini, marco.locatelli, aurelio.piazzi}@unipr.it, andrea.minari2@studenti.unipr.it

978-1-4673-8345-5/16/$31.00 ©2016 IEEE

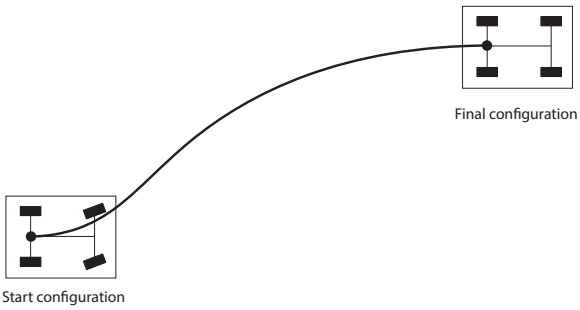

Fig. 1. A path to follow for an autonomous car-like vehicle.

this problem is reported in Section III. It presents a linear-time algorithm composed of three phases: a *forward iteration* (24), a *backward iteration* (25), and a final *join iteration* (26). Formal properties on the algorithm are expressed with Theorems 1, 2, and 3. Two examples are illustrated and discussed in Section IV. Concluding remarks are made in Section V. Proofs of Propositions 1, 2 and 3 leading to the theorems of Section III are reported in the final Section VI.

Notation: The set of real continuous functions with piecewise-continuous first-order derivative is denoted by $C^1_{\rm p}$. A Cartesian path has first-order geometric continuity or is a G^1-path if it is continuous along the curvilinear abscissa with continuous unit tangent vector. A G^2-path, or a path with second-order geometric continuity, is a G^1-path with continuous curvature along the curvilinear abscissa. For more information on geometric continuity, see [18], [19].

II. Motivation and problem formulation

In an automated driving scenario, a wheeled autonomous vehicle must follow a given Cartesian path starting from its current configuration to a future desired one (see Figure 1). The arc length measured along the path is s and $k(s)$ is the the path curvature as a function of s. Let $s_{\rm f}$ denote the total length of the path: this distance is requested to be travelled in minimum-time while satisfying constraints on the velocity and on the longitudinal and normal acceleration of the vehicle.

By denoting $t_{\rm f}$ the time to travel the path, the problem could be then approached by searching a velocity profile $v(t) \in C^1_{\rm p}[0,t_{\rm f}]$ according to:

$$\min_{v \in C^1_{\rm p}[0,t_{\rm f}]} t_{\rm f} \tag{1}$$

such that

$$s(t_{\rm f}) = s_{\rm f} \quad \text{with} \quad s(t) := \int_0^t v(\xi)d\xi, \tag{2}$$

$$v(0) = v_{\rm s}, \quad v(t_{\rm f}) = v_{\rm f}, \tag{3}$$

$$0 \le v(t) \le v_{\max},\ t \in [0,t_{\rm f}], \tag{4}$$

$$a^L_{\min} \le \dot{v}(t) \le a^L_{\max},\ t \in [0,t_{\rm f}], \tag{5}$$

$$v^2(t)|k(s(t))| \le a^N_{\max},\ t \in [0,t_{\rm f}]. \tag{6}$$

Equalities (3) are the interpolation conditions where $v_{\rm s}$ is the vehicle's start velocity and $v_{\rm f}$ is the assigned final velocity at the path end. Inequalities (4) impose that the velocity cannot be negative (i.e., the vehicle cannot go backward along the planned path) and does not exceed the maximum value $v_{\max}$. Constraints (5) limit the maximum and minimum longitudinal accelerations by $a^L_{\min}$ and $a^L_{\max}$ whereas inequality (6) imposes the upper bound $a^N_{\max}$ on the absolute value of the normal acceleration.

A difficulty in addressing the optimization problem (1)-(6) lies in the normal acceleration constraint (6). In this form, it is a strongly nonlinear constraint that makes difficult to find a global solution of (1)-(6). With the aim to overcome this difficulty we propose a problem reformulation based on curvilinear discretization.

The path is uniformly divided into $n-1$ elementary path parts, each one of length $h := s_{\rm f}/(n-1)$. The endpoints of these elementary parts are denoted by $\mathbf{p}_i$ with $i = 1,2,\ldots,n$ ($\mathbf{p}_1$ and $\mathbf{p}_n$ are the starting and final points respectively). Let v_i be the vehicle's velocity at point $\mathbf{p}_i$. Then the average speed on the i-th path part (between $\mathbf{p}_i$ and $\mathbf{p}_{i+1}$) is $(v_i + v_{i+1})/2$ and the corresponding time to travel this part is

$$t_i := \frac{2h}{v_i + v_{i+1}}. \tag{7}$$

The total time to travel the complete path is then

$$t_{\rm f} = \sum_{i=1}^{n-1} t_i = 2h \sum_{i=1}^{n-1} \frac{1}{v_i + v_{i+1}}. \tag{8}$$

The average longitudinal acceleration on the i-th part of the path is

$$a^L_i := \frac{v_{i+1} - v_i}{t_i} = \frac{v^2_{i+1} - v^2_i}{2h}. \tag{9}$$

At point $\mathbf{p}_i$, let s_i be the arc length and the normal acceleration is then $v^2_i k(s_i)$. With this reformulation, the addressed planning is to find a velocity sequence $\mathbf{v} := (v_1,\ldots,v_n) \in \mathbb{R}^n$ to solve the following optimization problem:

$$\min_{\mathbf{v} \in \mathbb{R}^n} 2h \sum_{i=1}^{n-1} \frac{1}{v_i + v_{i+1}} \tag{10}$$

such that

$$v_1 = v_{\rm s},\ v_n = v_{\rm f} \tag{11}$$

$$0 \le v_i \le v_{\max},\ i = 1,\ldots,n \tag{12}$$

$$a^L_{\min} \le a^L_i \le a^L_{\max},\ i = 1,\ldots,n-1, \tag{13}$$

$$v^2_i |k(s_i)| \le a^N_{\max},\ i = 1,\ldots,n. \tag{14}$$

Define $k_i := k(s_i)$, $l_B := 2ha^L_{\min}$, $u_B := 2ha^L_{\max}$, and

$$\bar{v}_i := \min\left\{v_{\max}, \sqrt{a^N_{\max}/|k_i|}\right\},\ i = 1,\ldots,n. \tag{15}$$

Hence, Problem (10)-(14) is equivalent to the following one:

Problem 1 (minimum-time velocity planning problem):

$$\min_{\mathbf{v} \in \mathbb{R}^n} \sum_{i=1}^{n-1} \frac{1}{v_i + v_{i+1}} \tag{16}$$

such that

$$v_1 = v_s,\ v_n = v_f \tag{17}$$
$$0 \le v_i \le \bar{v}_i,\ i = 1,\dots,n \tag{18}$$
$$l_B \le v_{i+1}^2 - v_i^2 \le u_B,\ i = 1,\dots,n-1\,. \tag{19}$$

III. The linear-time algorithm

In this section, we present an algorithm that solves Problem 1 in linear time with respect to the number of variables n. First, note that constraints (18), (19) can be rewritten as

$$v_i^2 - v_{i-1}^2 \le u_B,\ i = 2,\dots,n \tag{20}$$
$$v_i^2 - v_{i+1}^2 \le -l_B,\ i = 1,\dots,n-1 \tag{21}$$
$$v_i \le \bar{v}_i,\ i = 1,\dots,n \tag{22}$$
$$0 \le v_i,\ i = 1,\dots,n\,. \tag{23}$$

The algorithm is composed of a sequence of three iterations on the problem variables: the *forward iteration*, the *backward iteration* and the *join iteration*, detailed in the following.

The *forward iteration* computes a *forward solution* vector $\mathbf{f} := (f_1,\dots,f_n) \in \mathbb{R}^n$ according to

$$\begin{aligned} f_1 &= v_s \\ f_i &= \min\left\{\sqrt{f_{i-1}^2 + u_B}, \bar{v}_i\right\}, i = 2,\dots,n\,. \end{aligned} \tag{24}$$

The initial condition f_1 is the velocity initial value v_s and f_i is set equal to the maximum speed such that the acceleration constraint (20) and the maximum velocity constraint (22) are satisfied with $v_i = f_i$, $v_{i-1} = f_{i-1}$. In fact, the maximum velocity that satisfies (20) is given by $f_i = \sqrt{f_{i-1}^2 + u_B}$ and iteration (24) define f_i as the minimum between this value and maximum velocity $\bar{v}_i$.

The *backward iteration* is analogous but in reverse direction. It computes a velocity vector $\mathbf{b} := (b_1,\dots,b_n) \in \mathbb{R}^n$ as the solution of

$$\begin{aligned} b_n &= v_f \\ b_i &= \min\left\{\sqrt{b_{i+1}^2 - l_B}, \bar{v}_i\right\},\ i = 1,\dots,n-1\,. \end{aligned} \tag{25}$$

The final condition b_n is set as the velocity final value v_f and the backward iteration (25) sets the value of b_i equal to the maximum speed such that the deceleration constaint (21) and the maximum velocity constraint (22) are satisfied with $v_i = b_i$, $v_{i+1} = b_{i+1}$. Namely, the maximum velocity that satisfies (21) is given by $\sqrt{b_{i+1}^2 - l_B}$ and iteration (25) sets b_i equal to the minimum between this value and maximum velocity $\bar{v}_i$.

In the final step (*join iteration*), the optimal velocity profile is simply computed by

$$v_i^* = \min\{f_i, b_i\}, i = 0,\dots,n\,. \tag{26}$$

The overall procedure is reported in Algorithm 1. In particular, lines 1-3 implement the forward iteration, lines 4-6 the backward iteration and line 7-8 the join iteration.

We claim that Algorithm 1 is able to check the feasibility of problem (16) and that, in case the problem is feasible, vector $\mathbf{v}^*$ represents its solution. Moreover, Algorithm 1 is an algorithm of optimal time-complexity. More specifically, the following results hold.

Algorithm 1: The minimum-time velocity planning algorithm

input : $v_s,\ v_f,\ u_B,\ l_B,\ \bar{v}_i, i = 1,\dots,n$
output: $\mathbf{v}^*$

1 $f_1 \leftarrow v_s$
2 **for** $i \leftarrow 2$ **to** n **do**
3 $\quad f_i = \min\left\{\sqrt{f_{i-1}^2 + u_B}, \bar{v}_i\right\}$
4 $b_n \leftarrow v_f$
5 **for** $i \leftarrow n-1$ **to** 1 **do**
6 $\quad b_i = \min\left\{\sqrt{b_{i+1}^2 - l_B}, \bar{v}_i\right\}$
7 **for** $i \leftarrow 1$ **to** n **do**
8 $\quad v_i^* = \min\{f_i, b_i\}$

Theorem 1: Problem 1 is feasible if and only if vector $\mathbf{v}^*$, obtained by Algorithm 1, satisfies conditions

$$v_1^* = v_s, v_n^* = v_f\,.$$

Theorem 2: If Problem 1 is feasible, then vector $\mathbf{v}^*$, obtained by Algorithm 1, is a solution.

Theorem 3: Algorithm 1 solves Problem 1 with linear time complexity with respect to the number of variables n. Moreover, such complexity is optimal.

Proofs of the results. Theorems 1 and 2 are consequences of Propositions 2 and 3, proved in section VI. The proof of Theorem 3 is straightforward. Indeed, the time complexity of Algorithm 1 is linear with respect to n, since it is composed of three iterations on n. This is optimal in the sense that the time complexity of any algorithm able to solve Problem 1 is bounded from below by the $O(n)$ time required to load the problem data $\bar{v}_i$, $i = 1,\dots,n$.

IV. Examples

A. Example 1

As a first example to explain the steps of Algorithm 1, a simple G^1-path is considered. It is composed of a line segment, a circle arc, and a final line segment (cf. Figure 2). The minimum-time velocity planning on this path, whose total length is $s_f = 90$ m (cf. Problem 1), is addressed with the following data. The initial and final velocities are zero (i.e., $v_s = v_f = 0$), the maximal velocity is $v_{\max} = 30$ ms^{-1}, the longitudinal acceleration limits are $a_{\min}^L = -1.5$ ms^{-2} and $a_{\max}^L = 1.5$ ms^{-2}, and the maximal normal acceleration is $a_{\max}^N = 1$ ms^{-2}.

The number of samples is chosen as $n = 500$. Figures 3 and 4 show the corresponding functions $\mathbf{f}$ and $\mathbf{b}$ computed as the solution of the forward iteration (24) and, respectively, backward iteration (25). Figure 5 shows the final optimal solution $\mathbf{v}^*$ computed as the minimum of $\mathbf{f}$ and $\mathbf{b}$ as in (26). This solution corresponds to the minimum-time $t_f^* = 20.39$ s. Note that the vehicle starts from zero velocity and it

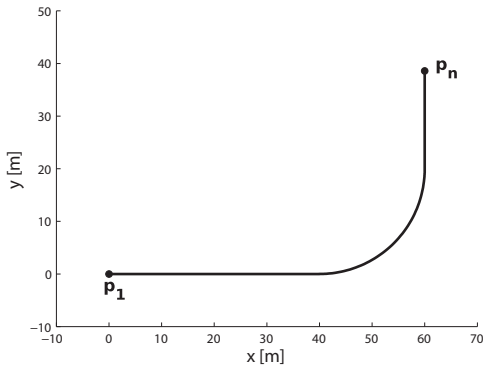

Fig. 2. Example 1: a simple G^1-path.

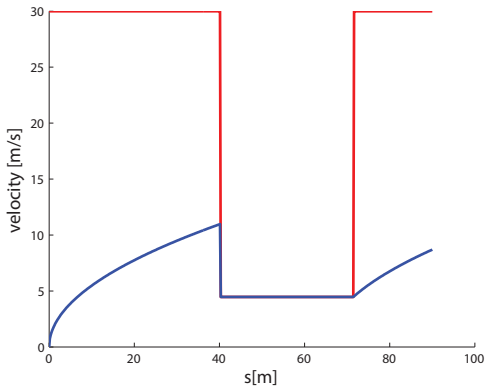

Fig. 3. Example 1 (*Forward iteration*): The red line represents the maximum velocity allowed along the path while the blue line represents the velocity sequence **f** computed after *Forward iteration*.

accelerates to a maximum velocity. Then, it slows down before the beginning of the curve, in order to respect the maximum velocity constraint due to the lateral acceleration on the curve. At the end of the curve, the vehicle accelerates and reaches a second maximum velocity after which it decelerates in order to satisfy the final zero velocity condition.

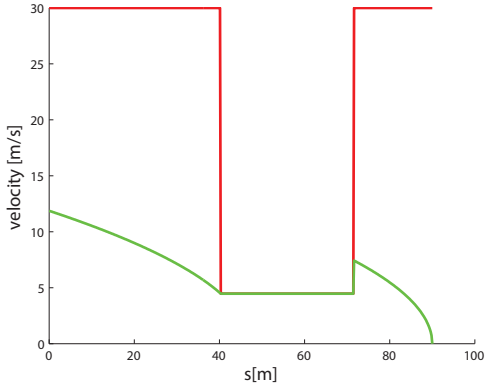

Fig. 4. Example 1 (*Backward iteration*): The red line represents the maximum velocity allowed along the path while the green line represents the velocity sequence **b** computed after *Backward iteration*.

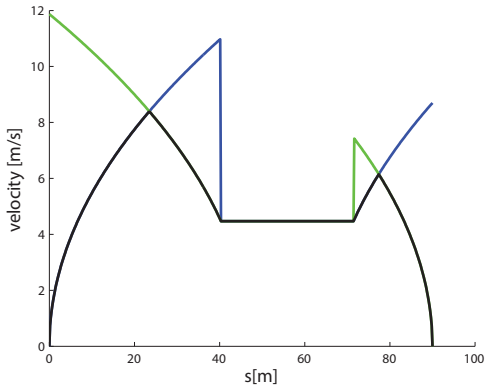

Fig. 5. Example 1 (*Join iteration*): In black the output of Algorithm 1 (the optimal velocity $\mathbf{v}^*$) is shown starting from the velocity sequences **f** and **b** previously computed.

B. Example 2

As a second example, consider a velocity planning on a G^2-path composed with η^2-splines [5] (cf. Figure 6). A single η^2-spline is a quintic polynomial curve that can interpolate given Cartesian points with associated unit tangent vectors and curvatures. The path in Figure 6 is composed by three η^2-splines whose interpolating data is reported in Table I: the θ's are the angles between the x-axis and the unit tangent vectors and the k's are the curvatures. The η parameters of these splines, which are free parameters to shape the spline path without affecting the interpolating conditions, are chosen as $\eta_1 = \eta_2 = 50$ and $\eta_3 = \eta_4 = 0$.

This planned path, which has total length $s_f = 153.05$ m, is composed by a lane-change path, an approximate clothoid, and an approximate circle arc (first, second, and third spline respectively). The velocity planning is addressed with $v_s = v_f = 0$, a maximal velocity $v_{max} = 36.1$ ms^{-1}, longitudinal acceleration limits $a^L_{min} = -10.5$ ms^{-2}, $a^L_{max} = 4$ ms^{-2}, and maximal normal acceleration $a^N_{max} = 7$ ms^{-2}. Algorithm 1 is applied with $n = 100$ achieving the minimum-time $t^*_f = 11.35$ s. The resulting optimal velocity profile is plotted in Figure 7.

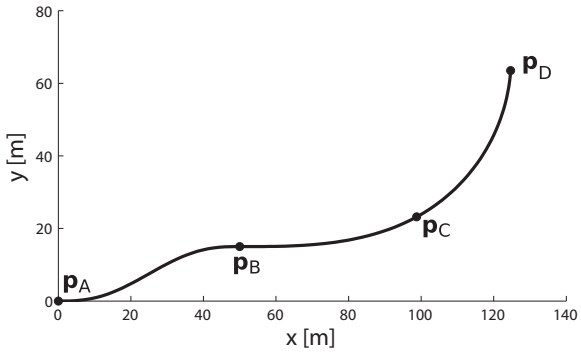

Fig. 6. Example 2: A G^2-path composed with η^2-splines [5].

V. Conclusions

Velocity planning is an important issue of the automated driving of autonomous vehicles. A fast, straightforward al-

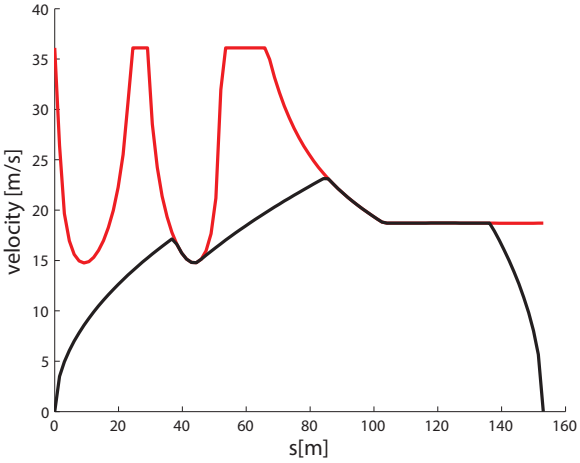

Fig. 7. Example 2: The red line represents the maximum velocity allowed along the path and the black line plots the optimal velocity profile.

TABLE I
INTERPOLATING DATA OF THE G^2-PATH

	x m	y m	θ rad	k m^{-1}
p_A	0	0	0	0
p_B	50.00	15.00	0	0
p_C	98.76	23.19	0.50	1/50
p_D	124.67	63.53	1.50	1/50

gorithm for a velocity planning that is time-optimal has been presented. This planning takes into account the usual kinematic constraints plus a bound on the vehicle's maximum normal acceleration that helps to avoid possible dangerous sideslips.

The algorithm has $O(n)$ computational complexity and it appears easy to implement in real-time. Moreover, the algorithm's output is a velocity profile that is a function of the arc-length along the path to be followed by the automated vehicle. This feature should make the velocity profile a good reference that facilitates the tracking by the feedforward/feedback motion controller of the autonomous vehicle. With the aim to obtain smoother velocity profiles, future research will try to extend the proposed approach by taking into account constraints on maximum longitudinal jerk.

VI. APPENDIX

In order to prove the main results, we need some additional notation. For $a,b \in \mathbb{R}$, define $a \wedge b = \min\{a,b\}$, $a \vee b = \max\{a,b\}$, for $\mathbf{u},\mathbf{v} \in \mathbb{R}^n$, define $\mathbf{u} \wedge \mathbf{v}$, $\mathbf{u} \vee \mathbf{v}$ as the analogous elementwise maximum and minimum operations. Define the partial order $\leq$ on $\mathbb{R}^n$ as

$$\mathbf{u} \leq \mathbf{w} \text{ if } u_i \leq w_i,\ i = 1,\ldots,n.$$

Define functions $F,B,M : \mathbb{R}^n \to \mathbb{R}^n$, such that, for $\mathbf{u} \in \mathbb{R}^n$, $F(\mathbf{u})$, $B(\mathbf{u})$ are given by the solutions of the difference equations

$$\begin{aligned} f_1 &= u_1 \\ f_i &= \sqrt{f_{i-1}^2 + u_B} \wedge u_i,\ i = 2,\ldots,n. \end{aligned} \tag{27}$$

$$\begin{aligned} b_n &= u_n \\ b_i &= \sqrt{b_{i+1}^2 - l_B} \wedge u_i,\ \ i = 1,\ldots,n-1. \end{aligned} \tag{28}$$

and $M(\mathbf{u}) = F(\mathbf{u}) \wedge B(\mathbf{u})$. Setting $\bar{\mathbf{v}} = (\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_n)$, it is apparent from the definition of M that the output $\mathbf{u}^*$ of Algorithm 1 satisfies $\mathbf{u}^* = M(\bar{\mathbf{v}})$. Further, define set

$$U \subset \mathscr{P}(\mathbb{R}^n) = \{\mathbf{u} \in \mathbb{R}^n : M(\mathbf{u}) = \mathbf{u}, 0 \leq \mathbf{u} \leq \bar{\mathbf{v}}, u_1 = v_s, u_n = v_f\}.$$

The following proposition characterizes the properties of F, B, M and U.

Proposition 1: Let $\mathbf{u},\mathbf{v} \in \mathbb{R}^n_+$, then the following properties hold:

a) $\mathbf{u} = M(\mathbf{u})$ if and only if $\mathbf{u} = F(\mathbf{u}) = B(\mathbf{u})$.
b) $\mathbf{u}$ is feasible for Problem 1 if and only if $\mathbf{u} \in U$.
c) $F(\mathbf{u} \vee \mathbf{v}) = F(\mathbf{u}) \vee F(\mathbf{v})$, $B(\mathbf{u} \vee \mathbf{v}) = B(\mathbf{u}) \vee B(\mathbf{v})$.
d) If $\mathbf{u},\mathbf{v} \in U$, then $M(\mathbf{u} \vee \mathbf{v}) = M(\mathbf{u}) \vee M(\mathbf{v})$.
e) If $\mathbf{u},\mathbf{v} \in U$, then $\mathbf{u} \vee \mathbf{v} \in U$.
f) If $\mathbf{u} \leq \mathbf{v}$, then $F(\mathbf{u}) \leq F(\mathbf{v})$, $B(\mathbf{u}) \leq B(\mathbf{v})$, $M(\mathbf{u}) \leq M(\mathbf{v})$.
g) $M(M(\mathbf{u})) = M(\mathbf{u})$.

Proof:

a, $\Rightarrow$): assume $M(\mathbf{u}) = \mathbf{u}$ and set $\mathbf{f} = F(\mathbf{u})$, $\mathbf{b} = B(\mathbf{u})$. Note that $f_1 = u_1$. Moreover, for $i = 2,\ldots,n$, $f_i \geq f_i \wedge b_i = u_i$, further, $f_i = \sqrt{f_{i-1}^2 + u_B} \wedge u_i \leq u_i$, hence $f_i = u_i$. The proof that $B(\mathbf{u}) = \mathbf{u}$ is analogous.

a, $\Leftarrow$): $\mathbf{u} = F(\mathbf{u}) = B(\mathbf{u})$ implies that $M(\mathbf{u}) = F(\mathbf{u}) \vee B(\mathbf{u}) = \mathbf{u}$.

b, $\Rightarrow$): assume that $\mathbf{u}$ is feasible for Problem 1, we first prove that $F(\mathbf{u}) = \mathbf{u}$. Let $\mathbf{f} = F(\mathbf{u})$, note that $f_1 = u_1 = v_s$, by induction, assume that $f_k = u_k$, $k = 1,\ldots,i-1$, then, since $u_i^2 - u_{i-1}^2 \leq u_B$, $f_i = \sqrt{f_{i-1}^2 + u_B} \wedge u_i = \sqrt{u_{i-1}^2 + u_B} \wedge u_i = u_i$. Analogously, we prove that $B(\mathbf{u}) = \mathbf{u}$. Then, by a), it follows that $M(\mathbf{u}) = \mathbf{u}$.

b, $\Leftarrow$) If $\mathbf{u} \in U$, by a) $F(\mathbf{u}) = \mathbf{u}$, then $u_{i+1} = \sqrt{u_i^2 + u_B} \wedge u_{i+1}$, which proves that (20) is satisfied. Analogously, $B(\mathbf{u}) = \mathbf{u}$ implies that (21) holds.

c) Let $\mathbf{u},\mathbf{v} \in U$. Let $\mathbf{a} = F(\mathbf{u})$, $\mathbf{b} = F(\mathbf{v})$, $\mathbf{c} = F(\mathbf{u} \vee \mathbf{v})$. We prove that $\mathbf{c} = \mathbf{a} \vee \mathbf{b}$ by induction. First, note that $c_1 = v_i = a_1 = b_1 = a_1 \vee b_1$, then, assume that $c_k = a_k \vee b_k$, $k = 1,\ldots,i-1$. Then $c_i = \sqrt{(a_{i-1}^2 \vee b_{i-1}^2) + u_B} \wedge u_i = (\sqrt{a_{i-1}^2 + u_B} \wedge u_i) \vee (\sqrt{b_{i-1}^2 + u_B} \wedge u_i) = a_i \vee b_i$. The proof that $B(\mathbf{u} \vee \mathbf{v}) = B(\mathbf{u}) \vee B(\mathbf{v})$ is analogous.

d) Since $\mathbf{u},\mathbf{v} \in U$, by a) $M(\mathbf{u}) = F(\mathbf{u}) = B(\mathbf{u})$, $M(\mathbf{v}) = F(\mathbf{v}) = B(\mathbf{v})$. Then, by c) $M(\mathbf{u} \vee \mathbf{v}) = F(\mathbf{u} \vee \mathbf{v}) \wedge B(\mathbf{u} \vee \mathbf{v}) = (F(\mathbf{u}) \vee F(\mathbf{v})) \wedge (F(\mathbf{u}) \vee F(\mathbf{v})) = F(\mathbf{u}) \vee F(\mathbf{v}) = F(\mathbf{u}) \vee B(\mathbf{v}) = M(\mathbf{u}) \vee M(\mathbf{v})$.

e) By d), $M(\mathbf{u} \vee \mathbf{v}) = M(\mathbf{u}) \vee M(\mathbf{v}) = \mathbf{u} \vee \mathbf{v}$, which implies that $\mathbf{u} \in U$.

f) Let $\mathbf{a} = F(\mathbf{u})$, $\mathbf{b} = F(\mathbf{v})$. Note that $a_1 = b_1$. By induction, assume that $a_k \leq b_k$, $k = 1,\ldots,i-1$. Then $a_i = \sqrt{a_{i-1}^2 + u_B} \wedge$

$u_i \leq \sqrt{b_{i-1}^2 + u_B} \wedge u_i = b_i$. Analogously we prove that $B(\mathbf{u}) \leq B(\mathbf{v})$. Moreover $M(\mathbf{u}) = F(\mathbf{u}) \vee B(\mathbf{u}) \leq F(\mathbf{v}) \vee B(\mathbf{v}) \leq M(\mathbf{v})$.

g) Let $\mathbf{f} = F(\mathbf{u})$, $\mathbf{b} = B(\mathbf{u})$, $\mathbf{m} = M(\mathbf{u})$, $\mathbf{a} = F(M(\mathbf{u}))$, $\mathbf{c} = B(M(\mathbf{u}))$. We prove that $\mathbf{a} = \mathbf{m}$, analogously it can be proved that $\mathbf{c} = \mathbf{m}$, which, by a) implies the thesis. Note that $a_1 = m_1$, by induction assume that $a_k = m_k$, $k = 1, \ldots, i-1$, then $a_i = \sqrt{a_{i-1}^2 + u_B} \wedge m_i = \sqrt{m_{i-1}^2 + u_B} \wedge m_i = \sqrt{f_{i-1}^2 \wedge b_{i-1}^2 + u_B} \wedge f_i \wedge b_i = (\sqrt{f_{i-1}^2 + u_B} \wedge f_i) \wedge (\sqrt{b_{i-1}^2 + u_B} \wedge b_i) = f_i \wedge b_i = m_i$. ■

Note that property h) of Proposition 1 shows that M is order preserving with respect to the partial order $\leq$, property g) shows that M is a projection. The following two propositions directly imply theorems 1 and 2.

Proposition 2: Problem 1 is feasible if and only if $M(\bar{\mathbf{v}}) \in U$.

Proof: $\Rightarrow$) By contradiction, assume that $\mathbf{m} = M(\bar{\mathbf{v}}) \notin U$ and that a feasible $\mathbf{u}$ exists. By g) of Proposition 1, $M(\mathbf{m}) = \mathbf{m}$, since $\mathbf{m} \notin U$, this implies that either $m_1 < v_s$ or $m_n < v_\mathrm{f}$. Since $\mathbf{u}$ is feasible, by b) of Proposition 1, $M(\mathbf{u}) = \mathbf{u}$ and, being $\mathbf{u} \leq \bar{\mathbf{v}}$, by f) of Proposition 1, $\mathbf{u} = M(\mathbf{u}) \leq M(\bar{\mathbf{v}})$, this implies that either $u_1 < v_i$ or $u_n < v_\mathrm{f}$, hence $\mathbf{u}$ is not feasible.

$\Leftarrow$) If $M(\bar{\mathbf{v}}) \in U$ then $M(\bar{\mathbf{v}})$ is feasible for Problem 1 by b) of Proposition 1.

■

Proposition 3: If Problem 1 is feasible, then $\mathbf{u}^* = M(\bar{\mathbf{v}})$ is its solution.

Proof: By contradiction, assume that there exists $\tilde{\mathbf{u}}$ such that

$$\sum_{i=2}^{n} \frac{1}{\tilde{u}_{i-1} + \tilde{u}_i} < \sum_{i=2}^{n} \frac{1}{u_{i-1}^* + u_i^*},$$

then, there exists i such that $\tilde{u}_i > u_i^*$. By b) and e) of Proposition 1, $\hat{\mathbf{u}} = \tilde{\mathbf{u}} \vee \mathbf{u}^*$ is feasible for Problem 1. Since $\tilde{\mathbf{u}} \leq \bar{\mathbf{v}}$, by f) of Proposition 1, $\tilde{\mathbf{u}} = M(\tilde{\mathbf{u}}) \leq M(\bar{\mathbf{v}}) = \mathbf{u}^*$ which contradicts the fact that $\tilde{u}_i > u_i^*$. ■

REFERENCES

[1] ERTRAC, "ERTRAC Automated Driving roadmap," July 2015, http://www.ertrac.org/index.php?page=ertrac-roadmap.

[2] A. Eskandarian, Ed., *Handbook of Intelligent Vehicles*. Springer Verlag, 2012.

[3] K. Kant and S. Zucker, "Toward efficient trajectory planning: the path-velocity decomposition," *Int. J. of Robotics Research*, vol. 5, no. 3, pp. 72–89, 1986.

[4] T. Fraichard and T. Howard, "Iterative motion planning and safety issue," in *Handbook of Intelligent Vehicles*, A. Eskandarian, Ed. London: Springer-Verlag, 2012, ch. 55, pp. 1433–1458.

[5] A. Piazzi, C. Guarino Lo Bianco, M. Bertozzi, A. Fascioli, and A. Broggi, "Quintic G^2-splines for the iterative steering of vision-based autonomous vehicles," *IEEE Trans. on Intelligent Transportations Systems*, vol. 3, no. 1, pp. 27–36, March 2002.

[6] T. Fraichard and A. Scheuer, "From Reeds and Shepp's to continuous-curvature paths," *IEEE Trans. on Robotics*, vol. 20, no. 6, pp. 1025–1035, Dec. 2004.

[7] A. Piazzi, C. Guarino Lo Bianco, and M. Romano, "η^3-splines for the smooth path generation of wheeled mobile robots," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 1089–1095, October 2007.

[8] F. Gravot, Y. Hirano, and S. Yoshizawa, "Generation of "optimal" speed profile for motion planning," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, Oct 2007, pp. 4071–4076.

[9] G. Lini, A. Piazzi, and L. Consolini, "Algebraic solution to minimum-time velocity planning," *International Journal of Control, Automation, and Systems*, vol. 11, no. 4, pp. 805–814, 2013.

[10] G. Lini, L. Consolini, and A. Piazzi, "Minimum-time constrained velocity planning," in *Control and Automation, 2009. MED '09. 17th Mediterranean Conference on*, June 2009, pp. 748–753.

[11] V. Muñoz, A. Ollero, M. Prado, and A. Simón, "Mobile robot trajectory planning with dynamic and kinematic constraints," in *Proc. of the 1994 IEEE Int. Conf. on Robotics and Automation*, vol. 4, San Diego, CA, May 1994, pp. 2802–2807.

[12] R. Solea and U. Nunes, "Trajectory planning with velocity planner for fully-automated passenger vehicles," in *IEEE Intelligent Transportation Systems Conference, ITSC '06*, September 2006, pp. 474 –480.

[13] J. Villagra, V. Milanés, J. Pérez, and J. Godoy, "Smooth path and speed planning for an automated public transport vehicle," *Robotics and Autonomous Systems*, vol. 60, pp. 252–265, 2012.

[14] C. Chen, Y. He, C. Bu, J. Han, and X. Zhang, "Quartic Bézier curve based trajectory generation for autonomous vehicles with curvature and velocity constraints," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, May 2014, pp. 6108–6113.

[15] X. Li, Z. Sun, A. Kurt, and Q. Zhu, "A sampling-based local trajectory planner for autonomous driving along a reference path," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, June 2014, pp. 376–381.

[16] C. Guarino Lo Bianco, A. Piazzi, and M. Romano, "Velocity planning for autonomous vehicles," in *IEEE Intelligent Vehicles Symposium, IV2004*, Parma, Italy, June 2004, pp. 413–418.

[17] S. Liu, "An on-line reference-trajectory generator for smooth motion of impulse-controlled industrial manipulators," in *Advanced Motion Control, 2002. 7th International Workshop on*, 2002, pp. 365–370.

[18] R. H. Bartels, J. C. Beatty, and B. A. Barsky, *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*. Palo Alto, CA: Morgan Kaufmann, 1995.

[19] F. Ghilardelli, G. Lini, and A. Piazzi, "Path generation using η^4-splines for a truck and trailer vehicle," *Automation Science and Engineering, IEEE Transactions on*, vol. 11, no. 1, pp. 187–203, Jan 2014.