Jerk-limited time-optimal speed planning for arbitrary paths

Antonio Artuñedo, Jorge Villagra, Jorge Godoy

***Abstract*—Both path and speed planning play key roles in automated driving, specially when a certain comfort level is required. This paper proposes a human-like speed planning method for already defined paths while minimizing travel time. The proposed method is able to compute a time-optimal speed profile that meet the given constrains with regard to speed, acceleration and jerk. For this purpose, an initial acceleration-limited approach is introduced. This algorithm serves as a starting point for the subsequent jerk-limited speed planning. Moreover, fallback strategies are included to manage critical driving situations where initial or final conditions cannot be met. The proposed approach has been tested and validated in an experimental platform through extensive trials in real environments. Its performance has been evaluated both in terms of quality of the computed speed profiles and with respect the required computing time.**

***Index Terms*—autonomous vehicles, motion planning, jerk-limited speed planning, comfortable automated driving.**

I. Introduction

In recent decades, research interest in autonomous driving has increased considerably and significant progress has been made in the technologies involved in this area. In addition to ensuring safe driving, it is expected that the driving of autonomous vehicles will be similar to that of human drivers, i.e. smooth and comfortable for the vehicle occupants. Although enormous research efforts have been made on motion planning techniques to enable autonomous vehicles to reach a given destination safely, the smooth behavior of automated vehicles remains a challenge, especially in medium and high speed environments

State-of-the-art decision-making architectures for automated vehicles are typically structured into global route planning, behavioural/manoeuvre planning and local motion planning. The latter functionality is of utmost importance as it has to generate a safe and feasible trajectory for the vehicle control system [19].

Automated driving requires parameterizable planning methods that enable the adaptability of the vehicle speed profile based on the driving scene understanding and predictions of the nearby environment. Although safety is the most important aspect when designing planning algorithms, comfort should also be considered. In this regard, it is important to note that the main factors contributing to uncomfortable driving are high levels of jerk and acceleration [6, 26], having jerk a stronger influence than acceleration [7]. Moreover, jerk-limited trajectories increase the similarity between automated and human-driven driving. Although several algorithms for obtaining jerk-limited profiles can be found in the literature, most of them are either limited to unmanned aerial vehicles or low speed ground vehicles, or make extensive use of iterative or optimization processes to obtain near-optimum speed profiles, or can only be applied to path defined through specific geometric primitives.

With the aim of producing human-like speed profile for already determined paths, in this paper we propose a jerk-limited speed planning method for arbitrary paths. The proposed algorithm is able to compute a time-optimal speed profile that meet the given constrains with regard to speed, acceleration and jerk.

The main contributions of this work are threefold: (i) the proposed time-optimal speed planning algorithm is path primitive agnostic while considering kinodynamic constraints of the vehicle. Even curvature of the path is not needed if the speed limit profile already consider lateral acceleration constraint; (ii) neither interpolation on the path or speed profile nor optimization algorithms are required, resulting in a computational efficient solution; (iii) the proposed approach introduces fallback strategies to handle constraints when initial and final conditions cannot be met.

The paper is structured as follows: In section II similar state-of-the-art approaches are reviewed. Section III describes the acceleration-limited approach while section IV focuses on the jerk-limited speed profile generation algorithm. In section V, the results of the trials carried out in a real environment are presented and analyzed. Finally, section VI draws some concluding remarks.

II. Related work

Motion planning problem has been one of the main focus of the robotics research in last decades. Although some existing approaches jointly address both path and speed planning [31], the complexity in obtaining safe and feasible trajectories makes this problem often divided into two different procedures: (i) path planning in order to compute collision free paths and feasible in terms of kinematic constraints, and (ii) speed planning for computing speed profiles for a given path that satisfies dynamic feasibility and comfort requirements, taking into account vehicle constraints.

Rather than speed planning, most of the attention in motion planning research has focused on different aspects of path planning such as obtaining optimal paths with regard to path

Antonio Artuñedo, Jorge Villagra and Jorge Godoy are with the Centre for Automation and Robotics (CSIC-UPM), Ctra. M300 Campo Real, Km 0,200, Arganda del Rey - 28500 Madrid, Spain. {antonio.artunedo, jorge.villagra, jorge.godoy}@csic.es

length or curvature [2, 19, 9]. Nevertheless, speed planning approaches for on-road autonomous driving are recently gaining considerable attention.

Some jerk-limited speed planning methods can be found in literature for different robotic applications such as manipulators [17, 12, 23], or unmanned aerial vehicles [13]. In [16], an algorithm to find an achievable acceleration-limited speed profile considering the dynamics of the vehicle is applied to obtain the minimum time traversal of a given path, using optimization techniques. The authors obtained speed profiles each 10 ms on paths defined through 50 points. A similar approach is also used in [30], where a strategy to find a minimum time speed profile traveling along a fixed path subject to the vehicle dynamics constraints, slip circle constraints, and actuator limits is proposed. In [14], authors use quintic Bézier curves to generate speed profiles considering bounding speed and acceleration.

Alternatively, some works focus on jerk-limited speed planning. The approach proposed in [29] is able to compute jerk-limited speed profiles given piecewise paths composed of clothoids, arcs of circles and straight lines. S-curve equations [17] are applied to each section of the path to achieve speed and acceleration continuity. In [20], the jerk-limited speed planning problem is stated as a non-linear and nonconvex one. To efficiently solve it, authors propose to use linear approximations at different stages that can be solve easily solved. In [28], a set of speed profiles is generated using 3rd order polynomial spline in the time domain. Then, a selection procedure discards those speed profiles that does not meet the speed and longitudinal acceleration limits. A recent work [22] proposes an heuristic approach for jerk-limited speed planning in laser guided vehicles for warehouse environments. This algorithm needs to be run recursively to reach a time-optimal solution and it is limited to 9 constant-jerk intervals in a trajectory.

Instead of generating trajectories for subsequent tracking, other existing approaches rely on model-predictive control (MPC) techniques. These approaches jointly solve the problem of planning and control by generating a set of future control actions that satisfy a set of constraints [4, 27, 8, 18] with regard to safety and comfort. MPC approaches are often computationally expensive as an optimization problem must be solved for each prediction step. As a result, to achieve acceptable run times, time horizon cannot be too long, which reduces the anticipation capability of the trajectory. To address these limitations, other approaches [11, 3, 28, 30, 15] based on long-term motion planning are often used to provide larger trajectories that increase vehicle anticipation and thus prevent sudden events from abruptly changing the trajectory.

As reviewed above, most of the speed planning approaches for autonomous driving bound accelerations and try to minimize jerk instead of bounding it. Moreover, those that limit jerk do so in specific situations, obtain non-time-optimal solutions and often apply computationally expensive methods. Besides, the limited time horizon and computation cost of MPC-based approaches are not yet able to cope with jerk-limited speed profiles generation for predefined paths in a real-time setting.

III. Acceleration-limited time-optimal speed planning

Before calculating a jerk-limited speed profile for a given path, the proposed approach in this work computes an acceleration-limited speed for each point of the trajectory. This auxiliary speed profile will be used both by the jerk-limited speed planning algorithm and to perform anticipatory reachability estimation, as explained in section IV-C. Indeed, in addition to the speed planning approach proposed in [1], this algorithm includes a fallback strategy to avoid speed profile discontinuities when initial an final speeds cannot be met with specified acceleration constraints.

Based on Pontryagin's Maximum Principle, time-optimal trajectories are assumed to be bang-bang in this work (see e.g. [21] for a similar consideration). As a result, the time-optimal speed profile for a predefined path follows successively maximum and minimum acceleration profiles while considering the given speed and acceleration constraints.

The acceleration-limited speed planning strategy comprises the computation of a speed limit curve and a subsequent algorithm for acceleration limiting. Both steps are detailed in the following subsections.

A. Speed limit curve

The speed limit curve sets a maximum velocity value at each point of the path taking into account different speed and acceleration constraints that can come from regulatory limits (maximum speed), comfort constraints (maximum comfort speed, acceleration and jerk) and dynamic constraints (available engine torque, sliding, ground contact and tip-over constraints) [25].

In this work, the speed limit curve has been used to consider the maximum speed and maximum lateral acceleration. However, the remaining constraints mentioned above can be also used to impose further limits to be taken into account in the proposed speed planning algorithm.

B. Acceleration-limited speed planning algorithm

The speed profile is calculated over a given path $P(x, y, \kappa)$, composed of a coordinate list of n two-dimensional points and curvature. This speed profile must comply with the constraints listed in table I, i.e. limit both longitudinal and lateral accelerations and linear speed, as well as impose initial and final speed. In other words, the goal of this step is to find a mapping of a linear speed v_i and a longitudinal acceleration a_i for each point i of P, taking into account the following constraints:

$$\begin{aligned} 0 < v_i < v_{max} \\ a_{min}^{long} < a_i < a_{max}^{long} \\ |a_i^{lat}| < a_{max}^{lat} \\ v_0 = v_{ini} \\ v_n = v_{end} \end{aligned} \tag{1}$$

Algorithm 1 shows the complete procedure to obtain an acceleration-limited speed profile associated to a given path. The speed profile calculation is performed in several stages:

Symbol	Description
v_{ini}	Initial speed
v_{end}	Final speed
v_{max}	Maximum allowed speed along the path
a_{max}^{lat}	Maximum lateral acceleration
a_{max}^{long}	Maximum positive longitudinal acceleration
a_{min}^{long}	Maximum negative longitudinal acceleration

TABLE I
PARAMETERS FOR ACCELERATION-LIMITED SPEED PROFILE GENERATION

Algorithm 1 Acceleration-limited speed planning algorithm

1: **procedure** AL-SPEEDPLANNING(P, v_{ini}, v_{end}, v_{max}, a_{min}^{long}, a_{max}^{long}, a_{max}^{lat})
2: $v^{SLC} \leftarrow f(P(\kappa), v_{ini}, v_{end}, v_{max}, a_{max}^{lat})$
3: $\Delta s \leftarrow f(P(x,y))$ ▷ Distances between path points
4: $a_{1,\dots,n} \leftarrow f(v^{SLC}, \Delta s)$ ▷ Obtain acceleration
5: **for** $i \leftarrow 1, n-1$ **do** ▷ Limit maximum positive acceleration
6: **if** $a_i > a_{max}^{long}$ **then**
7: $v_{i+1} \leftarrow f(v_i, a_i, a_{i+1}, \Delta s_i)$
8: $a_i \leftarrow a_{max}^{long}$
9: $a_{i+1} \leftarrow f(v_{i+2}, v_{i+1}, \Delta s_{i+1})$
10: **end if**
11: **end for**
12: **if** $v_n < v_{end}$ **then**
13: $a_{fb} \leftarrow f(\Delta v_{fb}, \Delta s_{fb})$
14: FALLBACKFINALSECTION(a_{fb})
15: **end if**
16: **for** $i \leftarrow n-1, 1$ **do** ▷ Limit maximum negative acceleration
17: **if** $a_i < a_{min}^{long}$ **then**
18: $v_i \leftarrow f(v_{i+1}, a_i, a_{i+1}, \Delta s_i)$
19: $a_i \leftarrow a_{min}^{long}$
20: $a_{i-1} \leftarrow f(v_i, v_{i-1}, \Delta s_{i-1})$
21: **end if**
22: **end for**
23: **if** $v_0 < v_{ini}$ **then**
24: $a_{fb} \leftarrow f(\Delta v_{fb}, \Delta s_{fb})$
25: FALLBACKINITIALSECTION(a_{fb})
26: **end if**
27: **return** v, a ▷ Return speed and acceleration for P
28: **end procedure**

1) The first step of algorithm 1 is to compute the speed limit curve, based on the maximum speed v_{max} and the maximum lateral acceleration a_{max}^{lat}. The maximum allowed speed v^{SLC} is computed using centripetal force equation

$$v_i^{SLC} = \min\{v_{max}, \sqrt{\frac{a_{max}^{lat}}{|\kappa_i|}}\} \tag{2}$$

where (κ_i) is the curvature of the given path (P) at a specific point i.

2) After that, initial and final speeds (v_{ini} and v_{end}) are imposed and the acceleration profile is computed assuming uniform acceleration between two consecutive points of the path:

$$v_i = \sqrt{v_{i-1}^2 + 2a_i\Delta s_i} \tag{3}$$

where Δs_i is the distance between points $i-1$ and i of the path P.

3) Then, the accelerations computed for each path point are traversed forward in order to verify that they are lower than the maximum acceleration value (a_{max}^{long}). In case the acceleration at point i overcomes the limit, it is bounded to the maximum value and the speed at point $i+1$ is recalculated using Eq. (3).

4) Finally, the same procedure followed in the previous step is performed backwards imposing a deceleration limit of a_{min}^{long} along the whole path.

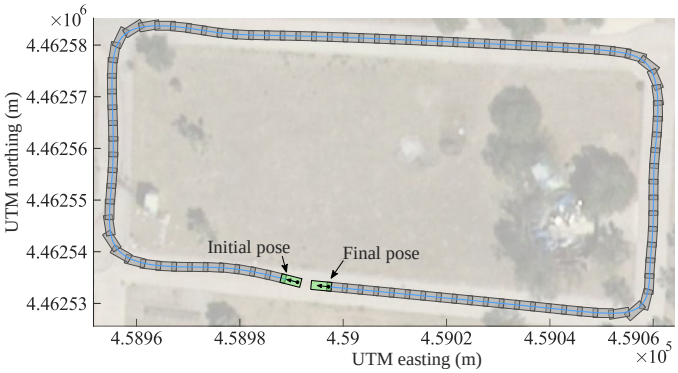


Fig. 1. Example of acceleration-limited speed planning

Fig. 2 shows an example of the acceleration-limited speed planning over a 289.2m path (see Fig 1) with the following constraints: $a_{max}^{lat} = a_{max}^{long} = 1.8m/s^2$, $a_{min}^{long} = -2.5m/s^2$, $v_{ini} = v_{end} = 0km/h$ and $v_{max} = 50km/h$. As can be seen, a smooth speed profile is generated while limiting longitudinal and lateral accelerations, as well as guaranteeing initial, final and maximum speed.

C. Fallback strategy for acceleration-limited speed planning

Lines 13-15 and 23-26 of Alg. 1 are devoted to verify if initial an final speeds can be reached with the given maximum acceleration constraints, avoiding thus discontinuities at the beginning or the end of the speed profile.

In the critical cases where initial and final speeds cannot be met, the longitudinal acceleration that allows to satisfy these end-point speed constraints (a_{fb}) is found applying Eq. 3:

$$a_{fb} = \frac{v_{ffb}^2 - v_{0fb}^2}{2\Delta s_{fb}} \tag{4}$$

where v_{0fb}, v_{ffb} and Δs_{fb} are the initial and final speeds and the length of the fallback section, respectively. Then a new acceleration-limited speed profile is computed using the calculated fallback acceleration.

In the schematic example of Fig. 3, v_{ini} cannot be met with the given acceleration constraints. In this case, a_{min}^{long} does not allow v_{ini} to be reached when performing the backwards correction of the algorithm. To guarantee speed continuity, a_{fb} is found and the initial section of the speed profile is computed. A analogous procedure is applied when v_{end} cannot be met.

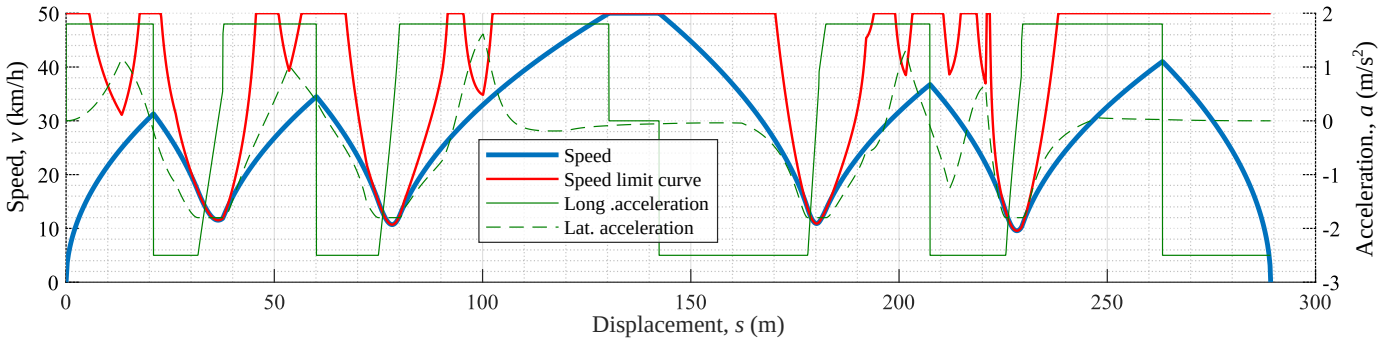


Fig. 2. Example of acceleration-limited speed planning for path of Fig. 1

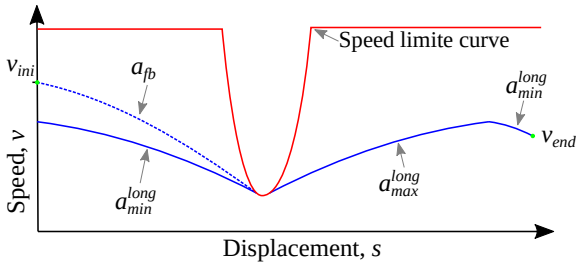


Fig. 3. Schematic example where v_{ini} cannot be met

Information about whether the initial and/or final section have been recomputed is stored and given an additional output of the algorithm. This information is helpful to identify fall-back actions in the jerk-limited speed planning, as is explained in section IV-C.

IV. Jerk-limited time-optimal speed planning

The acceleration limitation in the speed planning as presented in the previous section is a fast and effective method for speed planing while meeting kinodynamic constraints that can be considered in the speed limit curve. However, abrupt changes in acceleration could cause uncomfortable behaviour inside the vehicle. In these cases when comfort becomes more relevant (e.g. in passenger transportation), the jerk limitation becomes a need.

To meet the requirement of time-optimality when jerk is constrained, time-optimal speed profile must be generated successively applying maximum and minimum jerk profiles [21], while considering the given speed, acceleration and jerk constraints along the trajectory. However, it is reported in the literature [24, 21] that bang-bang approaches based on Pontryagin's Maximum Principle fails at singular arcs in which large amounts of continuous jerk switches are required, thus generating inaccurate results. The proposed approach deals with this kind of singularities by applying the maximum possible jerk, acceleration and speed at each of the points defining the trajectory.

In this section, an algorithm for jerk-limited speed planing is proposed. The objective of the algorithm is to find a mapping of speed v_i and acceleration a_i for each point i of the path P, taking into account the following constraints, whose parameters are listed in Table II:

$$\begin{aligned} 0 < v_i < v_{max} \\ a_{min}^{long} < a_i^{long} < a_{max}^{long} \\ |a_i^{lat}| < a_{max}^{lat} \\ j_{min} < j_i < j_{max} \\ v_0 = v_{ini} \\ v_n = v_{end} \\ a_0 = a_{ini} \\ a_n = a_{end} \end{aligned} \tag{5}$$

Symbol	Description
v_{ini}	Initial speed
v_{end}	Final speed
a_{ini}	Initial acceleration
a_{end}	Final acceleration
v_{max}	Maximum speed
a_{max}^{lat}	Maximum lateral acceleration
a_{max}^{long}	Maximum positive longitudinal acceleration
a_{min}^{long}	Maximum negative longitudinal acceleration
j_{min}	Minimum jerk
j_{max}	Maximum jerk

TABLE II
Parameters for jerk-limited speed profile generation

The following subsections focus on introducing the jerk-limited motion equations used, the detailed description of the algorithm and the explanations of the fallback strategy used when initial conditions cannot be met.

A. Constant jerk motion

Since jerk must be limited along the whole path, a constant jerk motion is considered between two consecutive points ($i-1$ and i) of the given path P. This constant jerk motion along time is described, for each point i, by the following expressions:

$$s_i = s_{i-1} + v_{i-1}t + \frac{1}{2}a_{i-1}t^2 + \frac{1}{6}j_i t^3 \tag{6}$$

$$v_i = v_{i-1} + a_{i-1}t + \frac{1}{2}j_i t^2 \tag{7}$$

$$a_i = a_{i-1} + j_i t \tag{8}$$

where s, v, a and j are the displacement, speed, acceleration and jerk, respectively, and t the time. Nevertheless, these

equations cannot be directly applied to the stated problem as they are defined in function of time and time independence is required since the position of every point in path P is invariable (it is an input). Moreover, the speed limit curve also contains speed constraints associated to each point.

Hence, to avoid a explicit time dependence, Eq. (6) can be rearranged replacing time t by its value in (8):

$$\Delta s = \frac{(a_i - a_{i-1})^3}{6j_i^2} + \frac{a_{i-1}(a_i - a_{i-1})^2}{2j_i^2} + \frac{v_{i-1}(a_i - a_{i-1})}{j_i} \tag{9}$$

The problem is reduced to find a value a_i in (9) for each pair of points ($i-1$ and i), given: $\Delta s = s_i - s_{i-1}$, v_{i-1}, a_{i-1} and j_i, such that the constraints specified in (5) are met.

Rectilinear motion is assumed between P_{i-1} y $P_i : i \in [1, n]$. As a result, the path P is handled as a set of $n-1$ rectilinear motions with constant jerk, subject to speed, acceleration and jerk constraints.

B. Jerk-limited speed planning algorithm

To solve the stated problem, algorithm 2 is proposed.

Algorithm 2 Jerk-limited speed planning algorithm

procedure JL-SPEEDPLANNING(P, v_{ini}, v_{end}, v_{max}, a_{ini}, a_{end}, a_{min}^{long}, a_{max}^{long}, a_{max}^{lat}, j_{max}, j_{min})
  v, a $\leftarrow$ AL-SPEEDPLANNING(P, v_{ini}, v_{end}, v_{max},a_{min}^{long}, a_{max}^{long}, a_{max}^{lat})
  AL-CONSTRAINTSCHECK(v, v_{ini}, v_{end})
  $j \leftarrow f(v, a, \Delta s)$
  $i_{local-min} \leftarrow$ COMPUTELOCALMINIMA(v)
  for all $i \in i_{local-min}$ **do**
    v, a, $j \leftarrow$ FIXLOCALMINIMUM(v, a, j, i)
  end for
  $i_{local-max} \leftarrow$ COMPUTELOCALMAXIMA(v)
  for all $i \in i_{local-max}$ **do**
    v, a, $j \leftarrow$ FIXLOCALMAXIMUM(v, a, j, i)
  end for
  return v, a, j ▷ Return speed, acceleration and jerk for P
end procedure

The first step of the proposed algorithm is to find local minima of the speed limit curve in order to build up a jerk-limited sections of the speed profile. To that end, the acceleration-limited speed planner (Alg. 1) is initially used, leveraging on its simplicity and computational efficiency. This step is not strictly necessary, as local minima could be extracted directly from the speed limit curve. However, the geometry of the path can result in a great variety of possibilities, leading to significant complexity differences when using the speed limit curve. The output of Alg. 1 typically provides a lower amount of local minima and consequently a lower and more stable total execution time while achieving the same results. To illustrate that effect, Fig. 2 shows a case in which 10 local minima would be obtained from the speed limit curve (red line), while after applying Alg. 1, only 4 local minima are attained (blue line).

Once every local minima is identified, they are used to generate jerk and acceleration limited sections of the speed

Algorithm 3 Procedure for local minimum fix

procedure FIXLOCALMINIMUM(v, a, j, i_{min})
  $flag_{sol} \leftarrow false$
  while $flag_{sol} = false$ **do**
    for $i \leftarrow i_{min}, n-2$ **do**
      if $a_i < a_{max}^{long}$ **then**
        $j_{aux} \leftarrow j_{max}$
        $a_{aux}, v_{aux} \leftarrow f(j_i, a_i, v_i, \Delta s_i)$
      else
        $j_{aux} \leftarrow 0$
        $a_{aux} \leftarrow a_i$
        $v_{aux} \leftarrow f(v_i, ai+1, \Delta s_i)$
      end if
      if $v_{aux} > v_{i+1}$ **then** ▷ Speed profile reached
        break
      else
        $v_{i+1} \leftarrow v_{aux}$
        $a_{i+1} \leftarrow a_{aux}$
        $j_i \leftarrow j_{aux}$
        $j_{j+1} \leftarrow f(a_{i+1}, a_{i+2}, v_{i+1}, \Delta s_{i+1})$
      end if
    end for
    if $i = n-2$ & $v_{i+1} < v_{end}$ **then** ▷ v_{end} not met
      $j_{max} \leftarrow j_{max} + \Delta j_{fb}$
    else
      $flag_{sol} \leftarrow true$
    end if
  end while
  $flag_{sol} \leftarrow false$
  while $flag_{sol} = false$ **do**
    for $i \leftarrow i_{min}, 2$ **do**
      if $a_i > a_{min}^{long}$ **then**
        $j_{aux} \leftarrow j_{max}$
        $a_{aux}, v_{aux} \leftarrow f(j_i, a_{i+1}, v_{i+1}, \Delta s_{i+1})$
      else
        $j_{aux} \leftarrow 0$
        $a_{aux} \leftarrow a_{i+1}$
        $v_{aux} \leftarrow f(v_{i+1}, ai, \Delta s_i)$
      end if
      if $v_i > v_i^{SLC}$ **then** ▷ Speed profile reached
        break
      else
        $v_i \leftarrow v_{aux}$
        $a_i \leftarrow a_{aux}$
        $j_i \leftarrow j_{aux}$
        $j_{i-1} \leftarrow f(a_i, a_{i-1}, v_i, \Delta s_i)$
      end if
    end for
    if $i = 0$ & $v_i < v_{ini}$ **then** ▷ v_{ini} not met
      $j_{min} \leftarrow j_{min} - \Delta j_{fb}$
    else
      $flag_{sol} \leftarrow true$
    end if
  end while
  return v, a, j
end procedure

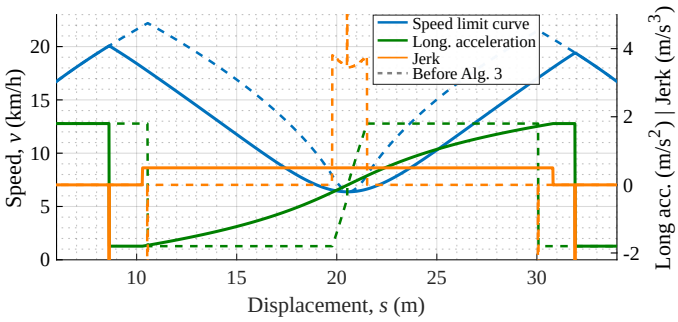


Fig. 4. Acceleration discontinuity fix at speed local minimum

profile (Alg. 3). An illustrative example of this procedure is shown in the speed profile in Fig. 4, where dashed lines shows the speed profile before computing Alg. 3 on the minimum at 20.5 m, and the result after this procedure is shown in continuous lines. This procedure sets null acceleration at the location of each local minimum speed (except at the initial and final point of the path) and computes two speed profile sections from local speed minimum, one forward and another one backward. The maximum positive allowed jerk (j_{max}) is applied until the maximum allowed accelerations are reached (maximum positive acceleration a_{max} in forward section and maximum negative acceleration a_{min} in backward section) or the acceleration limited speed profile is reached. Thus, the maximum jerk of the whole trajectory is limited to j_{max}. In the example in Fig. 4, the maximum positive allowed jerk (j_{max}) is set to $0.5m/s^3$, $a_{max} = 1.8m/s^2$ and $a_{min} = -1.8m/s^2$. It can be seen that, on the forward speed profile computation, the maximum positive acceleration is reached at 31 m and the speed reaches acceleration limited speed profile at 32 m. Starting from local minima at 20.5 m, the backward computation is carried out until the computed speed reaches acceleration limited speed profile at 8.6 m. In this case, the maximum negative acceleration is reached at 10.5 m. In this example it can also be seen that, although the acceleration has been limited to the maximum allowed values when computing Alg. 1, the positive jerk values exceeded j_{max} before computing Alg. 3. Nevertheless, as a result of applying this algorithm, it is ensured that the maximum jerk obtained in this section is limited to j_{max}. By applying this procedure to all identified local minima, it is consequently ensured that j_{max} is not exceeded across the entire speed profile.

Once the maximum jerk has been limited, a subsequent computation is carried out to find local maxima of the speed profile. Alg. 4 is then applied at each local maximum (labelled as i_{max}) to correct the speed profile by limiting the jerk to the minimum allowed jerk (j_{min}).

Alg. 4 is illustrated in Fig. 5, which depicts the speed, longitudinal acceleration and jerk before (in dashed lines) and after (continuous lines) applying Alg. 4. This algorithm consist of an iterative procedure that, starting from point $i_{max} - 1$, imposes the minimum allowed jerk j_{min} to find the closest point before i_{max} where to start applying j_{min} while satisfying acceleration and speed continuity at the end

Algorithm 4 Procedure for local maximum fix

procedure FIXLOCALMAXIMUM(v, a, j, i_{max})
  $k \leftarrow i_{max} + 1$
  for $i'_o \leftarrow i_{max} - 1, 2$ **do**
    while $i_k \in [i_{max}, n]$ **do**
      $a_{aux} \leftarrow f(j_{min} ...)$
      $v_{aux} \leftarrow f(v_{i+1}, a_{aux}, j_i)$
      if $v_{aux} < v_k^{SLC}$ & $a_{aux} < a_k$ **then**
        $flag_{sol} \leftarrow true$
        $i_o = i'_o$
        break
      end if
      $i_k \leftarrow i_k + 1$
    end while
  end for
  if $flag_{sol}$ **then**
    COMPUTESECTION(i_o,i_k)
  else ▷ Initial/final conditions can not be met
    $j_{fb} \leftarrow j + \Delta j_{fb}$
    FIXLOCALMAXIMUM(v, a, j', i_{max})
  end if
  return v, a, j
end procedure

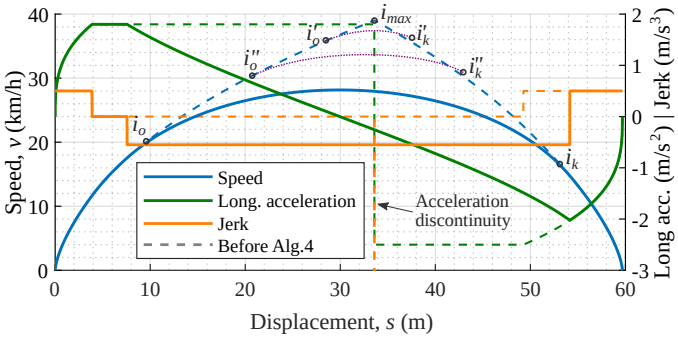


Fig. 5. Acceleration discontinuity fix at speed local maximum

of the section being computed (index i_k in Alg. 4). Three iterations of this procedure are represented in Fig. 5. The first iteration shown starts from point i'_o and reach the current speed profile at i'_k (dotted line), where an acceleration discontinuity is found. A subsequent iteration starts from i''_o and reach the current speed profile at i''_k but still the acceleration presents a discontinuity. Finally, a solution that meets the acceleration and speed continuity is found from i_o to i_k and the final jerk, acceleration and speed for that section is computed. Thus, it is guaranteed that j_{min} is not exceeded when a solution for that section is found.

Note that it is possible to adapt the speed planning behaviour over the path based on comfort (e.g. considering passengers driving preferences allowing them to choose the driving smoothness level) or safety criteria (e.g. considering high deceleration values to automatically avoid collisions) by varying the speed planner parameters (a_{max}^{lat}, a_{max}^{long}, a_{min}^{long} and v_{max}).

In some cases, initial and/or final conditions may not be

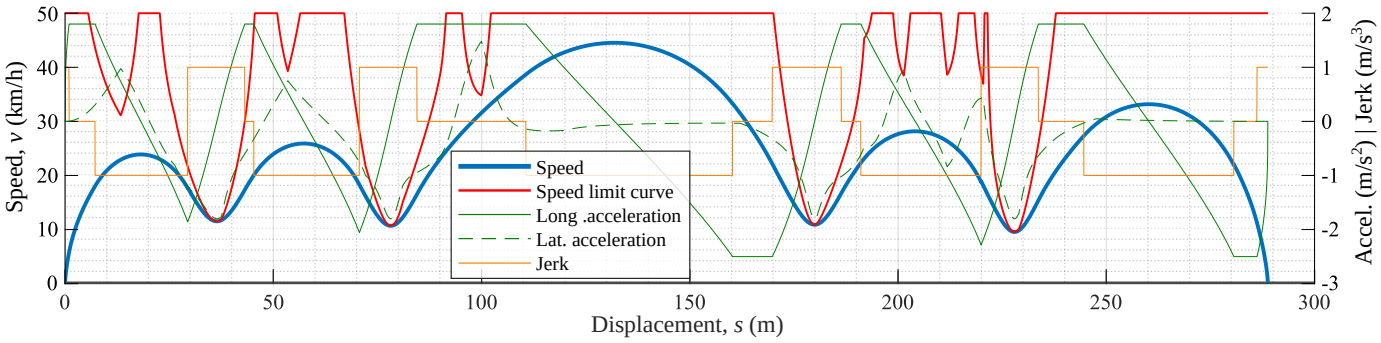


Fig. 6. Example of jerk-limited speed planning for path of Fig. 1

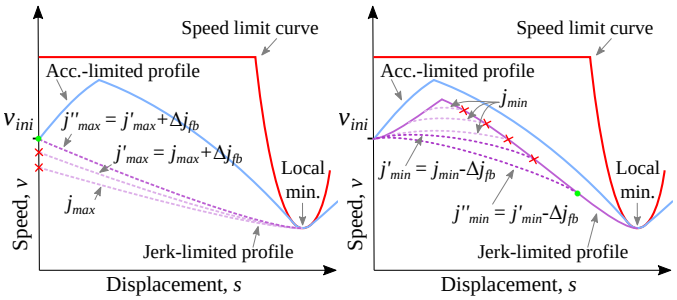
satisfied depending on the given constraints for jerk-limited speed planning. In these cases, fallback constraints are used to compute the initial/final section. The fallback strategy is detailed in the following subsection.

C. Fallback strategy in jerk-limited speed planning

As indicated in line 3 of Alg. 2, the jerk-limited planning algorithm runs Alg. 1 at the beginning, thus ensuring that initial and final speeds are met. However, in some specific case initial/final speeds and accelerations could not be met after applying jerk limitations. Thus, when the initial or final path point is reached while calculating a speed profile section with positive jerk (Alg. 3), it is firstly checked that the section being computed has not been previously computed by applying a fallback acceleration as described in section III-C. If it is not the case, it is assumed that initial acceleration (a_{ini}) and jerk limits cannot be met in this section and the speed profile in this section remains the same. In other words, if a less restrictive acceleration has had to be imposed in the fallback calculation when limiting acceleration (as shown in Fig. 3), lower speeds would be computed if jerk were limited in that section, making it impossible to reach the initial/final speed. Otherwise, the jerk-limited calculation is carried out. After that, it is checked that initial and final speeds are greater than v_{ini}, v_{end}, respectively. If not, $flag_{sol}$ in Alg. 3 is not activated and a fallback maximum jerk is used to recompute that section. The fallback maximum jerk is computed by increasing the maximum allowed jerk a given amount Δj_{fb} so that $j_{max} = j_{max} + \Delta j_{fb}$. This fallback procedure is iteratively computed until a solution is found or a maximum j_{ax} is reached (j_{jb}^{max}). In the latter case, the acceleration-limited profile for this section is conserved, assuming that jerk cannot be limited in this section.

An illustrative example of this strategy is shown in the speed profile section in Fig. 7a. It can be seen that the initial attempt to compute the backward section of the local minimum end with an initial speed lower than v_{ini}. Then, a fallback maximum jerk j'_{max} is used to recompute that section but again the initial speed is lower than v_{ini}. A second fallback maximum jerk j''_{max} is once again increased and the v_{ini} is met.

While computing the procedure for local maximum fix (Alg. 3), it is also possible that a solution cannot be found



(a) Fallback strategy to increase j_{max} (b) Fallback strategy to decrease j_{min}

Fig. 7. Fallback strategies in jerk-limited speed planning

with the given jerk and acceleration limits. The strategy in this case is to decrease the minimum allowed jerk a given amount Δj_{fb}, so that $j_{min} = j_{min} - \Delta j_{fb}$ and recompute it again. The procedure is recursively called (line 19 in Alg. 4) until a solution is found or a minimum j_{min} value is reached (j_{jb}^{min}).

An example of this strategy is shown in Fig. 7b. In this case, the first three represented iterations, where j_{min} is applied, are part of the normal operation of the algorithm. However, the first point of the trajectory is reached and a solution is not found using that value for j_{min}, meaning that a lower value is needed. The first fallback minimum jerk j'_{min} reaches the speed profile but acceleration continuity is not found in that point. A second iteration reduces again the minimum allowed jerk (j''_{min}) and a solution is found.

As in the previous case, when a solution is not found and j_{jb}^{min} is reached, the acceleration-limited profile for this section is conserved, assuming that jerk cannot be limited in this section. Please note that, for the sake of clarity, this mechanism is not included in the pseudocode in Alg. 4. Also note that, although the examples in Fig. 7 show the fallback strategies applied to the initial section when the initial conditions cannot be met, an analogous mechanism is applied when the final conditions cannot be met with the original jerk limits.

V. Experimental Results

The performance of the proposed speed planning algorithm has been extensively evaluated by computing speed profiles for a large set of paths with different geometries and lengths.

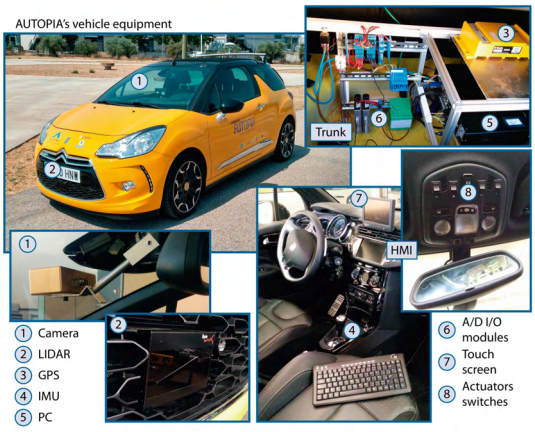


Fig. 8. Experimental platform

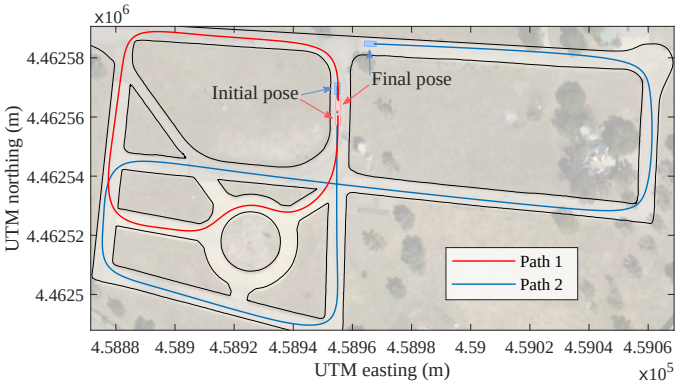


Fig. 9. Paths used in trials

Moreover, some of the generated trajectories have been tested on an instrumented vehicle.

The experimental platform used for testing and the evaluation details are respectively described in the following subsections.

A. Experimental platform

The proposed algorithm has been validated by performing different experiments using one of the automated vehicles of the AUTOPIA Program (see Fig. 8) at the test track of the Centre for Automation and Robotics (CSIC-UPM) in Arganda del Rey, Spain.

The localization of the vehicle used in these tests relies on a RTK-DGPS receiver and on-board sensors to measure vehicle speed, accelerations and yaw rate. The vehicle also includes a computer with an Intel Core i7-3610QE and 8Gb RAM, which is used to run the control architecture [1].

The trajectory tracking system used in this work, relying on fuzzy logic, is designed and behaves in a decoupled manner: on the one hand, the lateral controller uses both lateral and angular errors measured from the ego-vehicle pose with respect to the reference path, to compute the steering wheel position. On the other hand, the longitudinal controller computes the positions of throttle and brake pedals from the reference speed profile and the speed error (resulting from the difference of that reference and the measured speed of the vehicle). Further details about the trajectory tracking can be found in [10].

B. Trials on test track

Two different paths were used to test the proposed algorithm in real scenarios (see Fig. 9). While Path 1 includes a route with a high concentration of curves in a section of 100 meters, Path 2 (of 200 meters) includes a straight section of 180 meters that allows to reach higher speeds. Both paths were generated by applying the approach described in [3], i.e. path points, orientation and curvature are obtained from the analytic equations of the Bézier curves.

For each of the paths defined, a set of 7 speed profiles were generated considering different speed planning parameters, as

TABLE III
RESULTS OBTAINED FOR ALL SETUPS AND PATHS

Setup	j_{max}	j_{min}	Path 1		Path 2	
			MSJ	MSE_a	MSJ	MSE_a
1	0.1	-0.1	1.52	0.0510	1.87	0.0425
2	0.2	-0.2	1.79	0.0920	2.30	0.0787
3	0.3	-0.3	2.07	0.0870	2.56	0.0839
4	0.5	-0.5	2.46	0.0899	2.93	0.1236
5	0.8	-0.8	2.81	0.1804	3.98	0.2670
6	1.0	-1.0	3.07	0.2594	4.42	0.3144
7	–	–	4.21	0.5703	4.75	0.4937

indicated in Table III. While the maximum speed allowed in Path 1 was $v_{max} = 40km/h$, $v_{max} = 50km/h$ was set for Path 2 trajectories. In all setups the following values have been the same: $a_{max}^{long} = 1.2m/s^2$, $a_{min}^{long} = -2m/s^2$ and $|a_{max}^{lat}| = 1.2m/s^2$. In addition, the following values used in fallback strategy (section IV-C) were considered: $\Delta j_{fb} = -0.5m/s^3$ and $j_{jb}^{max} = -3m/s^3$.

For the sake of clarity, only some representative setups are depicted for each path. The generated speed profiles for setups 5 ($j_{max} = 0.8m/s^3$) and 7 (no jerk bounds) are shown in Fig. 10. Fig. 11 shows the speed profile computed in setups 3 ($j_{max} = 0.3m/s^3$) and 7. In both figures, continuous blue, green and yellow lines show the jerk-limited speed, acceleration and jerk, respectively. As can be seen, maximum jerk and acceleration limits are respected along the speed profile. Note that no abrupt acceleration changes occurs in jerk-limited profiles in contrast to just acceleration-limited profiles (depicted with dashed lines). For example, at 205 m in the speed profile in Fig. 10, an acceleration discontinuity in the acceleration-limited profile is corrected by the jerk-limited speed planning algorithm while respecting the maximum accelerations: starting from the maximum positive acceleration at 187 m, the minimum jerk is applied until the maximum negative acceleration is reached at 219 mj. Likewise, note that jerk-limited speed profiles have a smoother speed evolution along the path, which makes driving more comfortable when compared to acceleration-limited profiles.

The proposed approach does not necessarily impose maximum, minimum or null jerk on each point of the trajectory but intermediate values can be obtained in sections where the imposed jerk limits are not exceeded. This effect can be observed around 25 m and 55 m of the Path 1 results shown

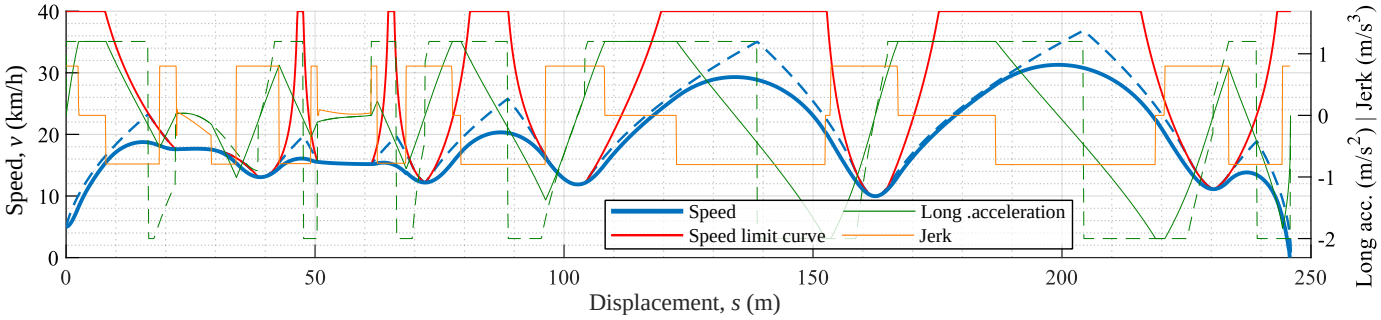


Fig. 10. Speed profiles of Path 1. Continuous lines: Setup 5 ($j_{max} = 0.8m/s^3$, $j_{min} = -0.8m/s^3$,). Dashed lines: setup 7 (no jerk limited)

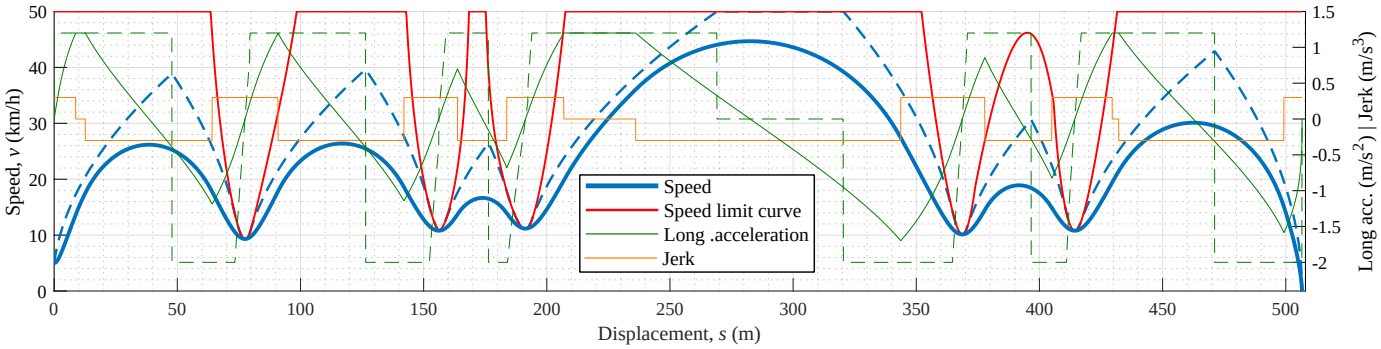


Fig. 11. Speed profiles of Path 2. Continuous lines: Setup 3 ($j_{max} = 0.3m/s^3$, $j_{min} = -0.3m/s^3$,). Dashed lines: setup 7 (no jerk limited)

in Fig. 10, where the speed limit curve in the first two curves present almost flat sections that do not cause high jerk values exceeding the imposed limits.

C. Performance evaluation

All the setups listed in Table III were tested for both paths in the experimental platform. Aiming at quantifying the jerkiness to compare the resulting vehicle performance while tracking the generated trajectories, the mean square jerk metric (MSJ) [5] is used to represent the jerkiness of each test is defined as follows:

$$MSJ = \frac{1}{t_f - t_0} \sum_{t_0}^{t_f} \frac{\Delta \hat{a}_{long}^2}{\Delta t} \tag{10}$$

where $\hat{a}_{long}$ is the longitudinal acceleration measured in the vehicle during the trials and t_0 and t_f are the initial and final time of the test.

The values of MSJ obtained for each setup and path are shown in Table III. As can be observed, the jerkiness in the vehicle increases when the maximum allowed jerk grows. Moreover, the maximum values of MSJ are reached in setup 7 for both paths, i.e. when there is no jerk constraint in the speed profile.

This effect can be graphically seen in Fig. 12. This figure shows density graphs of the actual longitudinal and lateral accelerations measured along the trials on Path 2. It can be noticed that greater negative (acceleration applied in left direction) than positive lateral acceleration values are measured. This is due to the fact that the vehicle must negotiate more right than left turns to reach the destination. Moreover, it is also observed that a broader spectrum is covered by the measured acceleration when jerk is limited (setups 1-6) in contrast to setup 7. For example, in setup 5 (Fig. 12-e), intermediate acceleration values are measured between maximum allowed longitudinal accelerations ($a_{max}^{long} = 1.2m/s^2$, $a_{min}^{long} = -2m/s^2$) unlike setup 7, causing a smoother driving.

It can be noticed that while most of the measured acceleration values are under the limits imposed in the speed planning, some of them go further. This effect is mainly caused by the lack of gravity compensation in accelerometers measurements. Indeed, road camber and longitudinal slope affect lateral and longitudinal acceleration measurements, respectively. The section of the test track traversed by Path 1 and 2 presents negligible longitudinal slopes. However, road camber is noticeable (around 6%), causing lateral acceleration reach up to $0.6m/s^2$ of measurements when the vehicle remains stopped in the steepest section of the track. Furthermore, lateral acceleration measurements are also influenced by vibrations induced by road imperfections and by positive speed tracking errors when driving on curves, thus generating greater centrifugal forces than planned. Consequently, greater disturbances are perceived in lateral acceleration measurements rather than in longitudinal acceleration ones.

When maximum allowed jerk is set to higher values (Figs. 12e-g), some positive longitudinal accelerations beyond the allowed limit can be found. These higher acceleration values are originated by the longitudinal controller reaction to the speed error caused by that the lack of acceleration during the shift from first to second gear. This effect can be easily seen in the temporal evolution of speed and accelerations of the trial on Path 2 with setup 7 depicted in Fig. 13 (at

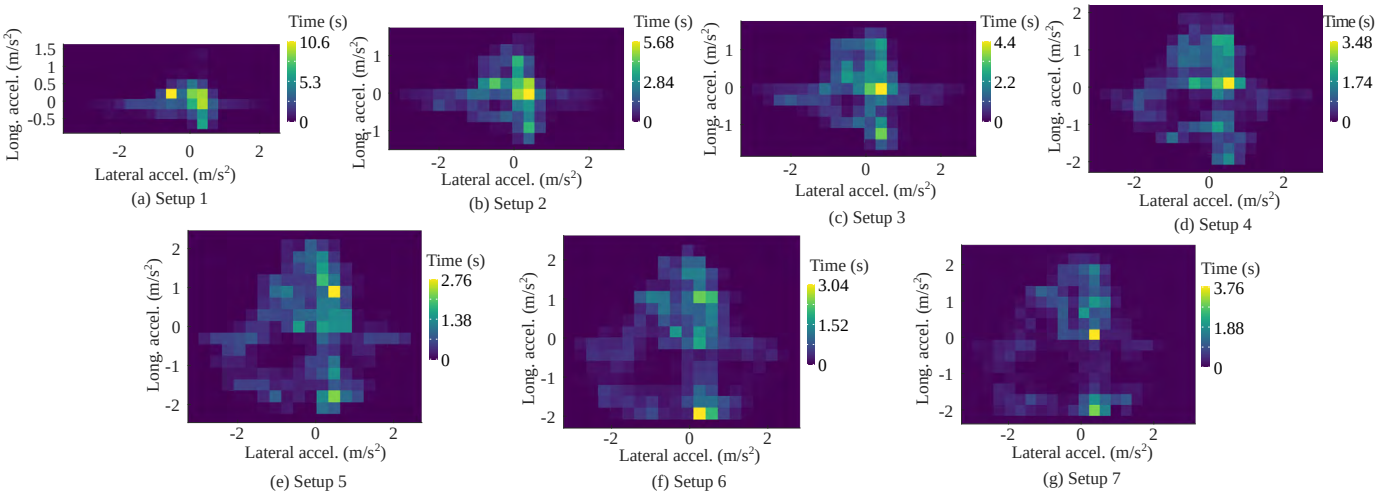


Fig. 12. Density graph of measured acceleration in the vehicle during the trials in Path 2

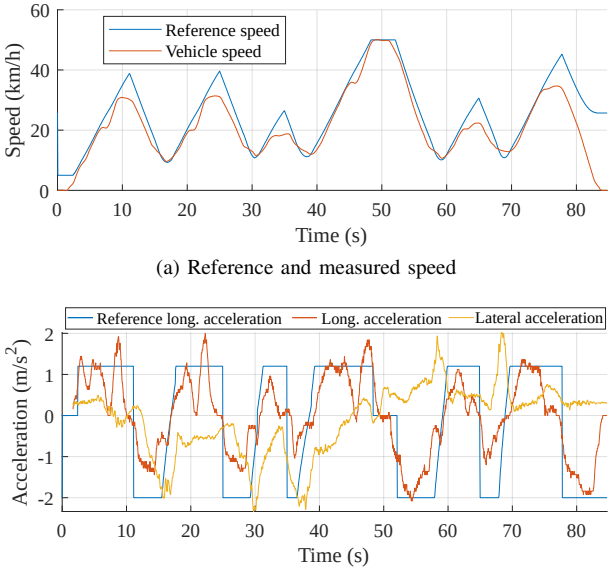


Fig. 13. Speed and accelerations during trial on Path 2 with setup 7 (no jerk limited)

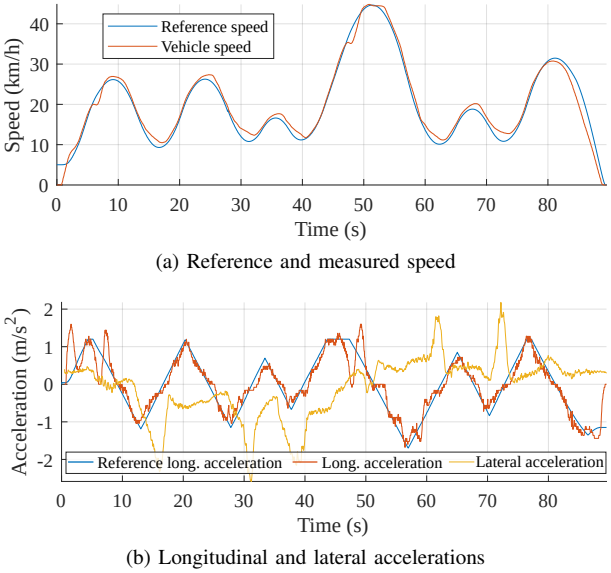


Fig. 14. Speed and accelerations during trial on Path 2 with setup 3

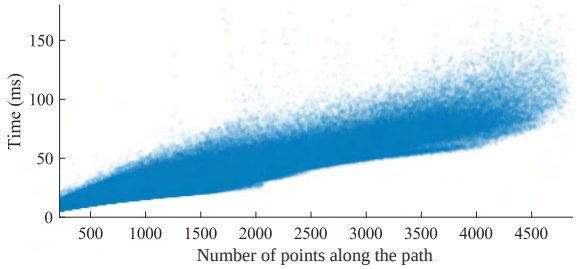
time $t = 6s$, $t = 20s$ and $t = 45s$). However, when the maximum allowed jerk is below $0.8m/s^3$, the vehicle is able to smoothly perform the gear shift as can be observed in the speed and acceleration evolution of setup 3 in Path 2 shown in 14. In order to quantify this effect, the mean squared error of the acceleration tracking (MSE_a) is shown in Table III. As reflected, the MSE_a increases when the allowed jerk does. Moreover, the greatest value is obtained for setup when no jerk limited. This highlights the positive impact of jerk bounding in the performance of trajectory tracking controllers.

D. Computation time analysis

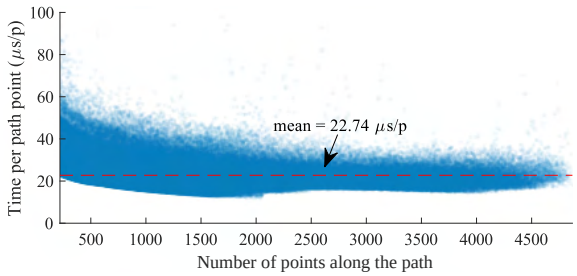
To perform a computation time analysis of the proposed speed planning algorithm, a set of 8 long paths with lengths between 100 m and $500m$ were generated over the test track at the Centre for Automation and Robotics. From these long paths, a big set of shorter paths were automatically generated by splitting each of them into path sections with incremental lengths by $1m$. Thus, a total amount of 423622 paths were generated. Finally, speed profiles were computed for each of the generated paths and the computing times were saved. The speed planning parameters for these tests were: $a_{max}^{long} = 1.2m/s^2$, $a_{min}^{long} = -2m/s^2$, $|a_{max}^{lat}| = 1.2m/s^2$, $j_{max} = 0.5m/s^3$ and $j_{min} = -0.5m/s^3$.

As expected, the run time of the speed planning algorithm highly depends on the amount of points of the path. This effect is shown in Fig. 15a, where it can be noticed a proportional relationship between the amount of points in the path and the run time of the jerk-limited speed planning algorithm. In Fig. 15 the microseconds per path point according to the number of points of each path is shown. A mean value of $22.71\mu s/p$ is obtained.

The separation of points of the paths (Δs) used for the this timing analysis is fixed along the paths and is set to $\Delta s =$



(a) Run time according to number of points of the path



(b) Run time per point according to number of points of the path

Fig. 15. Computation time results

$0.1m$. Considering this separation, we obtain a mean time according to the path length of $0.23ms/m$. Based on these results, and taking into account that the real-time planning architecture in which the proposed speed planing algorithm is integrated typically generates trajectories with path lengths around $50m$, an approximated average computing time for the speed planning is $11.35ms$.

It is important to note that $\Delta s = 0.1m$ is a generous discretization value for the path points even at low speed scenarios. In most situations $\Delta s = 0.2m$, or even greater values, are assumable. If $\Delta s = 0.2m$ is adopted,the number of path points is halved and so is the computation time consequently, thus obtaining $5.68ms$ for the speed planning computation on a $50m$ length path.

VI. Conclusions

The proposed algorithm is able to compute time-optimal speed profiles for given paths while meeting speed, acceleration and jerk constraints. The proposed algorithm does not need to apply iterative interpolations between time and displacement. Optimization algorithms are not involved either, allowing low computational cost of the proposed method. Furthermore, the proposed fallback strategies allows to maintain speed profile profile continuity in critical driving situations where initial or final conditions cannot be met with the given acceleration and jerk constraints.

An experimental platform has been used to test and validate the proposed approach through different trials in real environments. Moreover, the computing time required by the proposed algorithm has been analysed. The results show the capability of the proposed approach to be used in real automated driving applications.

Future works will focus on increasing the responsiveness of the speed planning in critical driving situations, further fallback strategies and their integration with a high-level decision-making system will be explored.

Acknowledgment

This work has been partially funded by the Spanish Ministry of Science, Innovation and Universities with National Projects COGDRIVE, PRYSTINE and NEWCONTROL (DPI2017-86915-C3-1-R, PCI2018-092928, PCI2019-103791, respectively), the Community of Madrid through SEGVAUTO 4.0-CM Programme (S2018-EMT-4362), and by the European Commission and ECSEL Joint Undertaking through the Projects PRYSTINE (783190) and NEWCONTROL (826653)

References

[1] A. Artuñedo, J. Godoy, and J. Villagra. “A decision-making architecture for automated driving without detailed prior maps”. In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. 2019, pp. 1645–1652.

[2] A. Artuñedo, J. Godoy, and J. Villagra. “A Primitive Comparison for Traffic-Free Path Planning”. In: *IEEE Access* 6 (2018), pp. 28801–28817.

[3] A. Artuñedo, J. Villagra, and J. Godoy. “Real-Time Motion Planning Approach for Automated Driving in Urban Environments”. In: *IEEE Access* 7 (2019), pp. 180039–180053.

[4] Mithun Babu et al. “Model Predictive Control for Autonomous Driving Based on Time Scaled Collision Cone”. In: Institute of Electrical and Electronics Engineers Inc., Nov. 2018, pp. 641–648. ISBN: 9783952426982. DOI: 10.23919/ECC.2018.8550510.

[5] S. Balasubramanian, A. Melendez-Calderon, and E. Burdet. “A Robust and Sensitive Metric for Quantifying Movement Smoothness”. In: *IEEE Transactions on Biomedical Engineering* 59.8 (2012), pp. 2126–2136.

[6] I. Batkovic et al. “Real-Time Constrained Trajectory Planning and Vehicle Control for Proactive Autonomous Driving With Road Users”. In: *2019 18th European Control Conference (ECC)*. 2019, pp. 256–262.

[7] Hanna Bellem et al. “Comfort in automated driving: An analysis of preferences for different automated driving styles and their dependence on personality traits”. In: *Transportation Research Part F: Traffic Psychology and Behaviour* 55 (2018), pp. 90 –100. ISSN: 1369-8478. DOI: https://doi.org/10.1016/j.trf.2018.02.036. URL: http://www.sciencedirect.com/science/article/pii/S1369847817301535.

[8] Gianluca Cesari et al. “Scenario Model Predictive Control for Lane Change Assistance and Autonomous Driving on Highways”. In: *IEEE Intelligent Transportation Systems Magazine* 9 (3 Sept. 2017), pp. 23–35. ISSN: 19411197. DOI: 10.1109/MITS.2017.2709782.

[9] L. Claussmann et al. “A Review of Motion Planning for Highway Autonomous Driving”. In: *IEEE Transactions on Intelligent Transportation Systems* 21.5 (2020), pp. 1826–1848.

[10] Jorge Godoy et al. “A driverless vehicle demonstration on motorways and in urban environments”. In: *Transport* 30.3 (2015), pp. 253–263. DOI: 10.3846/16484142.2014.1003406. eprint: https://doi.org/10.3846/16484142.2014.1003406. URL: https://doi.org/10.3846/16484142.2014.1003406.

[11] Tianyu Gu, John M. Dolan, and Jin-Woo Lee. “On-Road Trajectory Planning for General Autonomous Driving with Enhanced Tunability”. In: *Advances in Intelligent Systems and Computing*. 2016, pp. 247–261. ISBN: 9783319083377. DOI: 10.1007/978-3-319-08338-4_19. URL: http://link.springer.com/10.1007/978-3-319-08338-4{_}19.

[12] Dominik Kaserer, Hubert Gattringer, and Andreas Muller. “Nearly Optimal Path Following with Jerk and Torque Rate Limits Using Dynamic Programming”. In: *IEEE Transactions on Robotics* 35 (2 Apr. 2019), pp. 521–528. ISSN: 15523098. DOI: 10.1109/TRO.2018.2880120.

[13] Shu peng Lai et al. “Safe navigation of quadrotors with jerk limited trajectory”. In: *Frontiers of Information Technology and Electronic Engineering* 20.1 (2019), pp. 107–119. ISSN: 20959230. DOI: 10.1631/FITEE.1800719.

[14] Ray Lattarulo et al. “A Speed Planner Approach Based on Bézier Curves Using Vehicle Dynamic Constrains and Passengers Comfort”. In: *Proceedings - IEEE International Symposium on Circuits and Systems*. Vol. 2018-May. IEEE, May 2018, pp. 1–5. ISBN: 9781538648810. DOI: 10.1109/ISCAS.2018.8351307. URL: https://ieeexplore.ieee.org/document/8351307/.

[15] Xiaohui Li et al. “Real-time trajectory planning for autonomous urban driving: framework, algorithms, and verifications”. In: *IEEE/ASME Transactions on Mechatronics* 21.2 (2016), pp. 740–753.

[16] Thomas Lipp and Stephen Boyd. “Minimum-time speed optimisation over a fixed path”. In: *International Journal of Control* 87.6 (2014), pp. 1297–1311. ISSN: 13665820. DOI: 10.1080/00207179.2013.875224. URL: http://dx.doi.org/10.1080/00207179.2013.875224.

[17] Steven Liu. “An on-line reference-trajectory generator for smooth motion of impulse-controlled industrial manipulators”. In: *International Workshop on Advanced Motion Control, AMC*. 2002, pp. 365–370. DOI: 10.1109/amc.2002.1026947.

[18] Marcus Nolte et al. “Model predictive control based trajectory generation for autonomous vehicles-An architectural approach”. In: Institute of Electrical and Electronics Engineers Inc., July 2017, pp. 798–805. ISBN: 9781509048045. DOI: 10.1109/IVS.2017.7995814.

[19] Brian Paden et al. “A survey of motion planning and control techniques for self-driving urban vehicles”. In: *IEEE Transactions on Intelligent Vehicles* 1.1 (Mar. 2016), pp. 33–55. ISSN: 23798858. DOI: 10.1109/TIV.2016.2578706. arXiv: 1604.07446. URL: http://arxiv.org/abs/1604.07446http://ieeexplore.ieee.org/document/7490340/.

[20] Simone Perri, Corrado Guarino Lo Bianco, and Marco Locatelli. “Jerk bounded velocity planner for the online management of autonomous vehicles”. In: *IEEE International Conference on Automation Science and Engineering* 2015-October.October (2015), pp. 618–625. ISSN: 21618089. DOI: 10.1109/CoASE.2015.7294147.

[21] H. Pham and Q. Pham. “On the structure of the time-optimal path parameterization problem with third-order constraints”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 679–686. DOI: 10.1109/ICRA.2017.7989084.

[22] Marina Raineri and Corrado Guarino Lo Bianco. “Jerk limited planner for real-time applications requiring variable velocity bounds”. In: *IEEE International Conference on Automation Science and Engineering*. Vol. 2019-August. 2019, pp. 1611–1617. ISBN: 9781728103556. DOI: 10.1109/COASE.2019.8843215.

[23] Pedro Reynoso-Mora, Wenjie Chen, and Masayoshi Tomizuka. “On the time-optimal trajectory planning and control of robotic manipulators along predefined paths”. In: Institute of Electrical and Electronics Engineers Inc., 2013, pp. 371–377. ISBN: 9781479901777. DOI: 10.1109/acc.2013.6579865.

[24] Z. Shiller and S. Dubowsky. “On computing the global time-optimal motions of robotic manipulators in the presence of obstacles”. In: *IEEE Transactions on Robotics and Automation* 7.6 (1991), pp. 785–797. DOI: 10.1109/70.105387.

[25] Zvi Shiller and Yu Rwei Gwo. “Dynamic Motion Planning of Autonomous Vehicles”. In: *IEEE Transactions on Robotics and Automation* 7.2 (1991), pp. 241–249. ISSN: 1042296X. DOI: 10.1109/70.75906.

[26] C. Sohn, J. Andert, and R. N. Nanfah Manfouo. “A Driveability Study on Automated Longitudinal Vehicle Control”. In: *IEEE Transactions on Intelligent Transportation Systems* 21.8 (2020), pp. 3273–3280. DOI: 10.1109/TITS.2019.2925193.

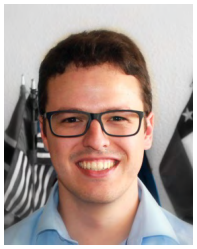
[27] Jongsang Suh, Heungseok Chae, and Kyongsu Yi. “Stochastic Model-Predictive Control for Lane Change Decision of Automated Driving Vehicles”. In: *IEEE Transactions on Vehicular Technology* 67 (6 June 2018), pp. 4771–4782. ISSN: 00189545. DOI: 10.1109/TVT.2018.2804891.

[28] Jordi Pérez Talamino and Alberto Sanfeliu. “Anticipatory kinodynamic motion planner for computing the best path and velocity trajectory in autonomous driving”. In: *Robotics and Autonomous Systems* 114 (Apr. 2019), pp. 93–105. ISSN: 09218890. DOI: 10.1016/j.robot.2018.11.022. URL: https://www.sciencedirect.com/science/article/pii/S0921889018301957?via{\%}3Dihub.

[29] Jorge Villagra et al. “Smooth path and speed planning for an automated public transport vehicle”. In: *Robotics and Autonomous Systems* 60.2 (Feb. 2012), pp. 252–265. ISSN: 0921-8890. DOI: 10.1016/J.ROBOT.2011.11.001. URL: https://www.sciencedirect.com/science/article/pii/S092188901100203X.

[30] Yu Zhang et al. “Hybrid Trajectory Planning for Autonomous Driving in Highly Constrained Environments”. In: *IEEE Access* 6 (2018), pp. 32800–32819. ISSN: 21693536. DOI: 10.1109/ACCESS.2018.2845448. URL: https://ieeexplore.ieee.org/document/8375948/.

[31] Julius Ziegler and Christoph Stiller. “Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios”. In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*. 2009. ISBN: 9781424438044. DOI: 10.1109/IROS.2009.5354448.



Antonio Artuñedo received a B.Sc. in Electrical Engineering from the Universidad de Castilla – La Mancha, Spain in 2011 and a M.Sc. in Industrial Engineering from the Universidad Carlos III de Madrid in 2014. In 2019, he received his PhD in Automation and Robotics at the Technical University of Madrid (UPM), Spain in the AUTOPIA Program. His PhD Thesis was awarded with the IDOM Prize for best Doctoral Thesis in Intelligent Control and was also the runner-up in the Robotnik Prize for the best Doctoral Thesis in Robotics, organized by the Spanish Committee of Automation. During his predoctoral period, he made a research stay at the Integrated Vehicle Safety group at TNO, Netherlands, in 2017. He joined the Centre for Automation and Robotics (CSIC-UPM) in 2013, where he has been working on both national and European reseach projects in the scope of autonomous vehicles. Antonio has published and peer-reviewed multiple journal and conference articles focused in this field. His research interests include system modelling and simulation, intelligent control, motion planning and decision-making systems.

Jorge Villagra graduated in Industrial Engineering at the Universidad Politécnica of Madrid in 2002. He received his PhD in Real-Time Computer Science, Robotics and Automatic Control at the École des Mines de Paris (France) in 2006. He was first granted with a 3 years CIFRE Program in PSA-Peugeot-Citroën and then with a post-doctoral fellowship at a Joint Research Unit INRIA-Mines ParisTech (France). The results of the PhD were granted with the Prize for the Best dissertation in Automatic Control in France in 2006. From 2007 to 2009 he held a position of Visiting Professor at the University Carlos III (Spain). He then received a 3 year JAEDoc fellowship at the AUTOPÍA Program in the Center for Automation and Robotics UPM-CSIC (Spain), where he spent one additional year funded by a research contract. From 2013 until August 2016 he led the Department of ADAS and Highly Automated Driving Systems at Ixion Industry & Aerospace SL, where he also coordinated all the activities in the EU R&D Funding Programmes. He is leading AUTOPIA Program at CSIC since October 2016. He has developed his research activity in 6 different entities with a very intense activity in project setup and management, through over 30 international and national R&D projects , where he is or has been IP of 10 of these projects. He has published over 85 papers in international journals and conferences on autonomous driving, intelligent transportation systems, dmodel-free control and new probabilistic approaches for embedded components in autonomous vehicles.

Jorge Godoy was born in Maracay, Venezuela, in 1986. He received the electronic engineer degree from the Universidad Simón Bolívar in 2008. In 2009 he was granted with a predoctoral JAE fellowship from CSIC for researching on autonomous vehicles at the Centre of Automation and Robotics (UPM-CSIC). In 2011, he received his M.E. degree in Automation and Robotics from the Universidad Politécnica de Madrid, finishing his PhD degree in the same area on 2013. From 2013 to 2017 he was the technical coordinator of the AUTOPIA Program, funded by research contracts from National and European research projects. In November 2017 he was granted with a Juan de la Cierva fellowship for postdoctoral research at the Universidad Politécnica de Madrid. His research interest includes intelligent transportation systems, autonomous driving, path planning and embedded AI-based control for autonomous vehicles.