Cooperative Driving of Automated Vehicles Using B-Splines for Trajectory Planning

Robbin van Hoek, Jeroen Ploeg, and Henk Nijmeijer, *Fellow, IEEE*

***Abstract*—Vehicle automation is used to increase traffic throughput and road safety. Automated vehicles should be highly versatile to navigate on both highways and urban roads. To do so, fully automated autonomous vehicles often adopt motion planners to ensure collision-free and comfortable trajectories. In contrast, in cooperative driving, feedback control is often used to achieve string-stable vehicle-following at very short inter-vehicle distances. In this work, the gap between cooperative vehicle following and trajectory planning is bridged, by using a trajectory planner to generate cooperative trajectories, such that the versatility of an autonomous vehicle is retained, but cooperation is included. The presented method uses B-splines to generate cooperative trajectories. Simulations are used to demonstrate that even under large communication delays, string stable behaviour can be achieved. The use of parametrized trajectories in the form of B-splines ensures only a small number of parameters needs to be communicated between vehicles.**

***Index Terms*—Automated vehicle, Autonomous vehicles, B-splines, Connected vehicles, Cooperative driving, Motion planning, String stability, Trajectory planning.**

I. Introduction

VEHICLE automation is currently in the center of attention due to its potential to increase both road throughput and safety. Safety is increased by elimination of human error in the driving task and by decreasing the response time from the system. Additionally, it can enable closer vehicle following and can eliminate so called 'ghost traffic jams,' both of which increase the road throughput [1].

Generally, the distinction between two types of automated vehicles can be made [2]. The first type is cooperative vehicles. These vehicles use Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I) communication to share information in order to cooperate in a (partially) automated manner. Examples of cooperative systems that controls the longitudinal motion are Cooperative Adaptive Cruise Control (CACC) [1], Connected Cruise Control [3], and automated intersection handling [4]. Additionally, systems have been developed to control the lateral motion in a cooperative manner [5]. An important aspect in the analysis of performance of cooperative systems is string stability [6]. This refers to the attenuation of the effects of disturbances along the string in the upstream direction. String-stable behaviour is important to ensure that accelerations remain bounded and feasible, and in addition, result in preventing ghost traffic jams. While string-stability can be achieved without vehicle to vehicle communication [7], communication allows for shorter inter-vehicle distances, which increases road capacity [1].

The second type of automated vehicles is generally referred to as an autonomous vehicle, since it is able to autonomously navigate in traffic. The vehicle under consideration corresponds with the *automated driving system* (level 3 and above) in the taxonomy defined by the SAE [8], where the driver is no longer part of the driving task. Autonomous vehicles often use trajectory planners to create feasible, comfortable and collision-free trajectories in complex traffic scenarios [9]–[14] and references therein. Where cooperative vehicles obtain additional environmental information via communication, autonomous vehicles rely only on on-board sensors.

Ideally, both types of vehicle automation are integrated in one vehicle. Such a vehicle would benefit from the capability of close vehicle following, while still retaining the versatility of the autonomous trajectory planner framework. The task of keeping string-stable inter-vehicle distances is thus assigned to the motion planning layer, instead of the feedback controller. A multitude of such automated vehicles would be capable of short-distance vehicle-following in a string-stable manner. At the same time, the autonomous vehicles would be able to simply break formation as the formation is not predefined. This is useful, e.g., in case the platoon is driving too slow to be beneficial, or the platoon has a different destination than the host vehicle. Furthermore, the vehicle would also still be capable of navigating roads autonomously, or as lead vehicle, using the same framework.

In addition to on-board sensors, such as typically used in autonomous vehicles, inter-vehicle communication, as commonly used in cooperative vehicle automation, is also employed. Communication between vehicles typically allows for shorter safe inter-vehicle distances, in which latency of communication is important [1]. The required bandwidth for this communication, and thus message size, is ideally low. This increases the message delivery ratio [15], and in turn improves the reliability of the communication between vehicles.

Some trajectory planners are capable of vehicle following. However, in most cases, no attempt has been made to assess

Manuscript received November 28, 2019; revised August 3, 2020 and November 9, 2020; accepted April 7, 2021. Date of publication April 23, 2021; date of current version August 23, 2021. This work was supported in part by the research program i-CAVE with Project number 14893, and in part by the Netherlands Organisation for Scientific Research (NWO). *(Corresponding author: Robbin van Hoek.)*

Robbin van Hoek and Henk Nijmeijer are with Mechanical Engineering Department, Eindhoven University of Technology, AZ 5612 Eindhoven, The Netherlands (e-mail: r.b.a.v.hoek@tue.nl; h.nijmeijer@tue.nl).

Jeroen Ploeg is with 2getthere, AE 3543 Utrecht, The Netherlands, and also with the Department of Mechanical Engineering, Eindhoven University of Technology, AZ 5612 Eindhoven, The Netherlands (e-mail: jeroen@2getthere.eu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIV.2021.3072679

the string stability properties in case multiple autonomous vehicles are subject to a vehicle-following situations, i.e., drive in a platoon. In [11], quintic polynomials were used to create minimum jerk trajectories for vehicle following. These were demonstrated to be string-unstable in [16]. In [12], a more traditional ACC controller is used for vehicle following, by estimating an acceleration for the preceding vehicle; however, no string stability was considered or not mentioned explicitly. Additionally, this approach did not consider V2V and thus, resulted in conservative trajectories. Moreover, if V2V communication would have been considered, it would require a high bandwidth of the wireless link, due to the particular representation of the trajectories. Another approach is that of [17], [18], [19], which present a Model Predictive Control (MPC) approach to solve the vehicle following problem, specifically for heterogeneous platoons. Although not originally intended for a trajectory planner, the MPC approach provides access to future states, and thus can be integrated with such a planner. In this work, string stability is explicitly analysed. However, still a large number of samples needs to be transmitted between vehicles. In contrast, the method of planning polynomial trajectories presented in [16] only requires a few coefficients to be communicated between vehicles, rather than a sampled trajectory. These trajectories are capable of transitioning a string of vehicles in steady state toward another steady state. However, this method was not capable of handling perturbations of the initial state, which resulted in undesirable trajectories.

In this work, a novel approach for cooperative trajectory planning considering string stable behaviour is presented. With this novel approach, the objective of distance regulation is shifted from the feedback control layer to the trajectory planning layer. In doing so, string-stable cooperative trajectory planning can be incorporated into the trajectory planning framework of the autonomous vehicle. By merging both cooperative vehicle following and autonomous driving into the same framework, the automated vehicle is capable of seamlessly switching between autonomous driving and string-stable cooperative driving. This could for example be achieved by a user specified cost-function, such that the decision of forming a platoon or overtaking a vehicle can be made. The resulting vehicle can thus drive at close inter-vehicle distances, yet remain highly versatile for other road scenarios. The cooperative trajectories are planned using B-splines, which form a class of piecewise polynomial functions. Since the B-spline is a parametrized curve, only a small number of parameters is communicated between vehicles, such that the communication bandwidth is significantly lower than the case where a sampled trajectory needs to be communicated. B-splines have a local modification property, such that the curve is only changed locally if a control point is changed, which makes small modifications on the trajectory easier. In contrast, for Bézier curves and polynomials, the entire graph is changed when a single coefficient is changed. Also note that a Bézier curve is actually a special case of a B-spline. Moreover, even when significant communication delays may be present, string-stable behaviour can be retained, which shows a significant advantage over traditional CACC systems.

This paper is organized as follows. First, Section II presents the framework of the trajectory planner for the autonomous vehicle and formulates the cooperative trajectory planner problem. Next in Section III, a brief introduction to B-splines and their properties is presented, followed by the main contribution of this work, which outlines the methodology of using B-splines in cooperative vehicle control. In Section IV, the functionality of the planner is demonstrated using a simulation platform. Finally, Section V presents conclusions of this work and recommends on future research directions.

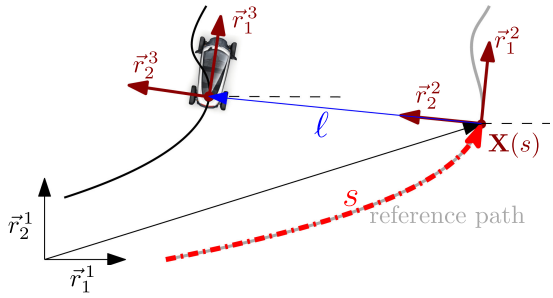


Fig. 1. Reference path described by $\mathbf{X}(s)$, deviated path constructed via lateral offset ℓ in the moving Frenet frame $\vec{r}^{\,2}(s)$, which is parametrized by curvilinear distance, s.

II. Framework and Problem Formulation

This section first presents the trajectory planning framework of the autonomous car. With the framework presented, it will outline how cooperation is included in this framework. Finally, the control objectives and performance criteria of the cooperative addition are formulated.

A. Framework

The same approach as presented in [11] is followed, since cooperation can easily be incorporated in this framework. A reference path $\mathbf{X}(s) = [x(s), y(s)]$ in inertial reference frame $\vec{r}^{\,1}$, with components $\vec{r}_1^{\,1}$, $\vec{r}_2^{\,1}$, as displayed in Fig. 1 is used. This reference path can be regarded as being the centreline of the road, with sufficient continuity properties. To allow the planner to deviate from the reference path, a moving Frenet frame $\vec{r}^{\,2}$, with components $\vec{r}_1^{\,2}$, $\vec{r}_2^{\,2}$ is defined. The planning problem is now defined as determining the curvilinear distance $s(t)$, of the moving Frenet frame, and the lateral distance $\ell(t)$, of the vehicle to the Frenet frame, parametrized to time t, such that $\{s(t), \ell(t)\}$ defines a trajectory. Using the closed form transformation as presented in [11] these trajectories can also be expressed in the vehicle frame $\vec{r}^{\,3}$ with components $\vec{r}_1^{\,3}$, $\vec{r}_2^{\,3}$. In particular, the longitudinal and lateral coordinates together with their time derivatives form state trajectories $\mathbf{S}(t) := [s(t), \dot{s}(t), \ddot{s}(t)]^\intercal$ and $\mathbf{\Lambda}(t) := [\ell(t), \dot{\ell}(t), \ddot{\ell}(t)]^\intercal$. The objective of the trajectory planner is to find feasible, collision free and comfortable trajectories, $\mathbf{S}(t), \mathbf{\Lambda}(t)$ over a time horizon $t \in [t_0, t_0 + T]$, while considering dynamic obstacles. These trajectories would start from initial state $\mathbf{S}(t_0), \mathbf{\Lambda}(t_0)$.

Instead of solving a possibly complex optimization problem that includes dynamic obstacles and constraints, a set of possible trajectories is generated and afterwards tested on possible violations of constraints and on the occurrence of collisions. This simplifies the problem significantly. It has been shown

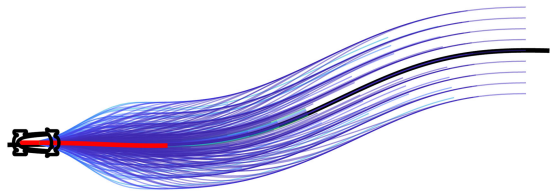


Fig. 2. Visualization of a set of generated trajectories (—) with a grid of terminal conditions, spatial reference path (—).

in [11] that using quintic polynomials for $s(t)$ and $\ell(t)$ results in minimum jerk manoeuvres for a given terminal state at a given terminal time. For normal autonomous operation, quintic polynomials are generated to connect initial state with a grid of terminal constraints in both lateral, longitudinal and temporal dimensions [16]. This is illustrated in Fig. 2.

It is important that the planner considers multiple potential trajectories simultaneously. The method generates a diverse set of trajectories that includes changing lanes, coming to a stop and maintaining velocity. As a result, the cooperative trajectory can simply be added to this set. A cost function is used to select the optimal (feasible and collision free) trajectory from the generated set. This allows the new framework to automatically select between for example cooperation in the form of platooning, or simply overtaking the vehicle if the velocity of the preceding vehicle is smaller than the desired velocity. CACC systems do not have this flexibility, since the preceding vehicle is always followed. Clearly, the resulting planner is capable of operating both autonomously and cooperatively and is capable of switching seamlessly between these modes, making the resulting automated vehicle highly versatile. In what follows, only the generation of the cooperative trajectory will be considered.

B. Control Objectives

The aim is to integrate string-stable vehicle-following in trajectory planning. To this extent, a string of vehicles indexed by $i \in \{0, 1, \ldots, M\}$ is considered, with the leading vehicle having index $i=0$. The trajectory of vehicle i at timestep k is denoted as $\mathbf{S}_{i,k}(t)$. The objective is now to plan a trajectory $\mathbf{S}_{i,k}(t), t \in [t_{k,i}, t_{k,i} + T_i]$, using communicated information, $\mathbf{S}_{i,k}, t \in [t_{k,i}, t_{k,i} + T_i]$, of the preceding vehicle. In this work, a predecessor-following configuration is used, where the host vehicle only uses information of the preceding vehicle. This makes the ad-hoc forming of platoons easier. Moreover, in [20], a multi-vehicle look-ahead was shown not to always be beneficial. A more general communication topology could potentially be integrated but is outside the scope of the work presented here. Only the curvilinear trajectories, $\mathbf{S}_{i,k}(t)$ will be considered, since the lateral trajectories $\mathbf{\Lambda}_{i,k}(t)$ are not critical to the performance of the string in the context of this work. Also note that, as can be seen from Fig. 2, each $\mathbf{S}_i(t)$ has a multitude of lateral trajectories associated with it.

Several aspects are important regarding this trajectory planning. First of all, this control system faces hard constrains on computation times [12], since it is operating in collision-critical situations. Hence, it is important that it can be guaranteed that a new trajectory is available for execution when the vehicle needs one. Another important aspect is the required bandwidth for communicating the planned trajectory to the following vehicle. Note that the trajectory $\mathbf{S}_{i,k}(t)$ can be transmitted in various ways. Intuitively, due to the discrete-time implementation originating from the real-time implementation on the host vehicle, it would make sense to simply communicate the sampled trajectory over the horizon. However, as the planning horizon and update rate of the real-time computer may be large, the amount of data that needs to be transmitted will also be large. Naturally, a larger data block that needs to be transmitted, may result in larger latency, which is undesirable.

Next, properties of the generated trajectories itself are considered. Clearly, safety is important, hence it is required that the cooperative trajectories are collision free, i.e., $s_i(t) \leq s_{i-1}(t)$ assuming vehicle lengths equal to zero. Another critical aspect of cooperative trajectories is string stability. String stability implies that a disturbance introduced in the system, does not grow unbounded in the upstream direction. Aside from preventing so called 'ghost-traffic jams,' this property prevents unbounded growth of the acceleration inputs beyond the accelerations constraints of the vehicle. Therefore string stability is a necessary condition to ensure feasibility of trajectories of vehicles in the upstream direction. The notion of string stability is described in [6] and the references contained therein, where $\mathcal{L}_p$ string stability is defined as

$$\frac{\|y_i(t)\|_\infty}{\|y_{i-1}(t)\|_\infty} \leq 1, \qquad \forall i \tag{1}$$

$$\|y_i(t)\|_p = \sqrt[p]{\int_{-\infty}^{\infty} y_i^p(t)\mathrm{d}t} \tag{2}$$

in which $y_i(t)$ is an output of the cascaded system under consideration. In this case, the acceleration of these vehicles are used as the outputs, because of the physical significance.

Another important aspect of trajectory planning is temporal consistency [11]. This refers to the recursive implementation of the trajectory planner, where the overlapping part of the time horizon of trajectories planned in two planning cycles matches in case the situation has not changed. This is important as a following vehicle uses a transmitted trajectory to construct its own trajectory. Additionally, if consecutive trajectories do not significantly differ, previous trajectories can be utilized in case of packet drop-out in the wireless communication for robustness. In this work, a performance indicator is used that indicates to what extent the trajectory $\mathbf{S}_{i,k}(t), t \in [t_{k,i}, t_{k,i} + T_i]$ of planning step k overlaps with the trajectory of another planning step. Planning times are updated with planning interval t_p, such that $t_{k+1,i} = t_{k,i} + t_\mathrm{p}$. As performance indicator, the maximum difference between planning step k and planning step j (with $k < j$) on the overlapping time horizon is used, which is defined as

$$\begin{bmatrix} \Gamma_{k,j,s} \\ \Gamma_{k,j,\dot{s}} \\ \Gamma_{k,j,\ddot{s}} \end{bmatrix} = \begin{bmatrix} \max_t |s_{i,k}(t) - s_{i,j}(t)| \\ \max_t |\dot{s}_{i,k}(t) - \dot{s}_{i,j}(t)| \\ \max_t |\ddot{s}_{i,k}(t) - \ddot{s}_{i,j}(t)| \end{bmatrix}, t \in [t_{j,i}, t_{k,i}+T_i] \tag{3}$$

From this definition two criteria can be derived. The first is the maximum difference between two consecutive trajectories $\Gamma_{k,k+1}$. The second is the maximum difference over all trajectories that share a part of their horizon, $\max_j \Gamma_{k,j}, j \in$

$\{k+1,\ldots,k+\lfloor T_i/t_\mathrm{p}\rfloor\}$. These performance indicators will be used to study temporal consistency of the planner.

III. Approach

In this work, B-splines are adopted to describe trajectories. Like polynomial trajectories, the entire trajectory can be elegantly parametrized by only a few parameters, which can then be communicated to other traffic participants. However, unlike polynomials, B-splines can be modified locally, giving more design freedom in constructing the trajectories. B-splines have been used in optimization schemes for trajectory planning, such as in [21], but have traditionally been used more often for graphics and curve fitting [22]. This section will describe how a B-spline is defined, and how it is used to construct trajectories. First, B-splines will be presented, together with some useful properties. With the general B-spline curve described, B-spline functions that are used as trajectories are explained. Finally, the construction of these trajectories is described in the context of cooperative automated driving.

A. B-Splines

A B-spline, $\mathbf{C}(u)$, is a piecewise polynomial of degree p, with parametrisation variable u. A general multidimensional spline is defined as

$$\mathbf{C}(u)=\sum_{j=0}^{n} N_{j,p}(u)\mathbf{P}_j,\quad u\in[u_0,u_m] \tag{4}$$

where $\boldsymbol{\mathcal{P}}=[\mathbf{P}_0,\mathbf{P}_1,\ldots,\mathbf{P}_n]^\intercal$ are $n+1$ control points, corresponding to basis functions $N_{j,p}(u)$, of degree p, which are recursively defined as [22]:

$$N_{j,0}(u)=\begin{cases}1 & \text{if}\quad u_j\le u<u_{j+1}\\ 0 & \text{otherwise}\end{cases},$$

$$N_{j,p}(u)=\frac{u-u_j}{u_{j+p}-u_j}N_{j,p-1}(u)+\frac{u_{j+p+1}-u}{u_{j+p+1}-u_{j+1}}N_{j+1,p-1}(u) \tag{5}$$

via a so-called knot vector $\mathbf{U}=[u_0,u_1,\ldots,u_m]$ with $m+1$ elements, where $u_{i-1}\le u_i$, such that zero-order functions are step functions and nonzero on a single knot interval, first-order functions are linear and nonzero on two knot intervals, second-order functions are quadratic and nonzero on three knot intervals and so on. These basis functions form a partition of unity such that $\sum_{j=0}^{n}N_{j,p}(u)=1,\forall u$. The knot vector determines the shape of the basis functions and is a nondecreasing vector. The number of knots, $m+1$, together with the number of control points, $n+1$, uniquely determine the degree of the curve via $p=m-n-1$. Typically, so-called *clamped* knot vectors are used, in which the first and last $p+1$ knots are identical, resulting in the B-spline coinciding with the first and last control point.

The control points can be connected to form a control polygon. This control polygon dictates the shape of the B-spline due to the strong convex hull property. This property follows from the fact that if $u\in[u_j,u_{j+1})$ then $\mathbf{C}(u)$ is in the convex hull of control points $\{\mathbf{P}_k\mid k\in\{j,j-1,\ldots,j-p\}\}$. When $\mathbf{P}_j$ is indeed multidimensional, this control polygon can be plotted in the same dimensional space as the curve itself. However, the control polygon can also be visualized in the parameter space. This is useful when working with a one dimensional curve, i.e., P_j is a scalar, as is done in this work. To find the parameter coordinates, μ_j, of the control points, P_j, the following formula is used [23]

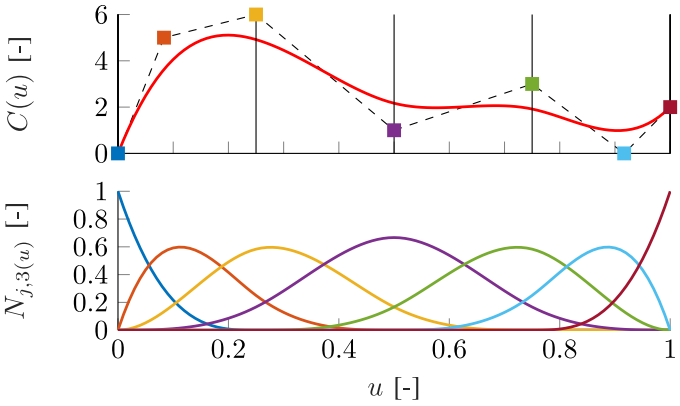


Fig. 3. B-spline ($n=6$, $p=5$) $C(u)$ (—), knots (|), Control polygon (- -), control points (μ_j,P_j) (■) and and corresponding Basis functions, $N_{j,p}(u)$ in the same color.

$$\mu_j=\frac{1}{p}\sum_{k=1}^{p}u_{j+k},\qquad j\in\{0,\ldots,n\} \tag{6}$$

These points are also referred to as Greville abscissae or Marsen-Schoenberg points. Using these points, it can be demonstrated that the first and last piece of the control polygon are tangential to the start and end of the B-spline. An example of this is illustrated in Fig. 3, where the control polygon is illustrated by the dashed line. This control polygon is constructed via the control points of which the corresponding basis functions $N_{j,p}$, are given in the lower plot. It can be seen that for all three curves, the spline follows the direction changes of the control polygon. Moreover, the control points form a convex hull for the B-spline.

Another important feature is that the derivative of a B-spline is a B-spline itself. The derivative of a B-spline of degree p is a B-spline of degree $p-1$, referred to as the hodograph. The control points of the kth hodograph (i.e., the k-th derivative) are denoted as $\mathbf{P}_j^{(k)}$ and are determined by the following recursion

$$\mathbf{P}_j^{(k)}=\begin{cases}\mathbf{P}_j & \text{for } k=0\\ \frac{p-k+1}{u_{j+p+d}-u_{j+d}}\left(\mathbf{P}_{j+1}^{(k-1)}-\mathbf{P}_j^{(k-1)}\right)\end{cases} \tag{7}$$

for $j\in\{0,\ldots,n-k\}$. Note that the number of control points decreases by one for each derivative. Also note that (7) can be rewritten as a matrix multiplication

$$\boldsymbol{\mathcal{P}}^{(k)}=\underbrace{\boldsymbol{\Delta}_{k,k-1}(\mathbf{U})\cdot\boldsymbol{\Delta}_{k-1,k-2}(\mathbf{U})\cdot\ldots\cdot\boldsymbol{\Delta}_{1,0}(\mathbf{U})}_{\boldsymbol{\Delta}_{k0}}\boldsymbol{\mathcal{P}} \tag{8}$$

where the dependency on $\mathbf{U}$ in $\mathbf{\Delta}_{k0}$ is omitted in the notation for brevity.

$$\mathbf{\Delta}_{k,k-1}(\mathbf{U}) = \begin{bmatrix} \delta_0 & & \\ & \ddots & \\ & & \delta_{n-d} \end{bmatrix} \begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix}$$

$$\delta_i = \frac{p+1-k}{u_{i+p+k} - u_{i+k}} \tag{9}$$

In addition, the knot vector corresponding to the k-th derivative of a B-spline loses $2k$ knots (according to $p = m - n - 1$), to ensure that the degree of the derivative curve is indeed lower than that of the original curve. In fact, the knot vector for the k-th derivative curve is $\mathbf{U}^{(k)} = [u_k, u_{k+1}, \ldots, u_{m-k-1}, u_{m-k}]$, where the knots are from the original knot vector. Note that in (7), the knots of the original curve are used as well to determine the control points.

B. B-Spline Trajectories

With the basics of B-splines explained, the use of B-splines for describing trajectories is covered. The objective is to describe state trajectories $\mathbf{S}_i(t)$, using splines. To this extent, the time t is utilized as the parametrisation variable. Then a one dimensional B-spline is used to describe a trajectory $s_i(t)$ for vehicle i:

$$s_i(t) = \sum_{j=0}^{n} N_{j,p}(t) P_{j,i}, \qquad t \in [t_{k,i}, t_{k,i} + T_i] \tag{10}$$

where $\boldsymbol{\mathcal{P}}_i = [P_{0,i}, P_{1,i}, \ldots, P_{n,i}]^\intercal$ are $n+1$ control points. The knots in the knot vector $\mathbf{U}_i$ are also expressed in the time domain. To ensure that the trajectory is defined over the desired time horizon, the first knot should coincide with the current time, $t_{k,i}$, and the final knot should extend to the time horizon $t_{k,i} + T_i$. A uniform clamped knot vector is used,

$$\mathbf{U}_i = [\underbrace{t_{k,i}, \ldots, t_{k,i}}_{p+1}, \underbrace{u_{p+1}, \ldots, u_n}_{n-p}, \underbrace{T_i + t_{k,i}, \ldots, T_i + t_{k,i}}_{p+1}],$$

$$u_j = \frac{j-p}{n-p+1} T_i + t_{k,i}, \qquad j = \{p+1, \ldots, n\} \tag{11}$$

Note that once the trajectory for $s_i(t)$ is defined, the hodograph can simply be computed to find $\dot{s}_i(t)$ and $\ddot{s}_i(t)$. This means $\mathbf{S}_i(t)$ can be evaluated using only control points $\boldsymbol{\mathcal{P}}_i$, current planning time $t_{k,i}$ and time horizon T_i. This is illustrated in Fig. 4, which shows the B-spline for $s_i(t)$ and corresponding spline trajectories for $\dot{s}_i(t)$ and $\ddot{s}_i(t)$.

C. Vehicle Model

Note that for a trajectory to be feasible, it is required to start from the current vehicle state, which is achieved by letting the current position, $s_i(t_{k,i})$, coincide with the first control point, $P_{0,i}$, due to the clamped knot vector. The initial velocity $\dot{s}_i(t_{k,i})$ is simply found via the hodograph relation, with control points according to (7), to coincide with the first control point, $P^{(1)}_{0,i}$, which is related to the difference in the first two control points $P_{0,i}, P_{1,i}$. Similarly, the initial acceleration, $\ddot{s}_i(t_{k,i})$ coincides with $P^{(2)}_{0,i}$. As such, the current state $\mathbf{S}_i(t_{k,i})$ is fixed via the

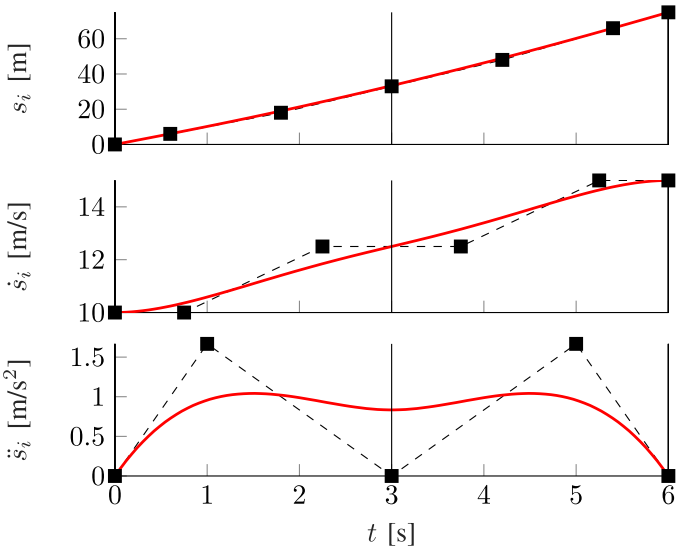


Fig. 4. B-spline trajectory ($n = 7, p = 5$) $s_i(t)$ (—), control points $(\mu_j, P_{j,i})$ (■), control polygon (- -) and knots (|).

first three control points $P_{j,i}, j \in \{0, 1, 2\}$. By including the acceleration in the state vector, continuity of the acceleration is thus ensured. This is needed as the drive-train of vehicles is not capable of instantaneous changes in driving torque. In fact, the same model as used in [1], [18], [19], [24] can be adopted, which represents the experimental demonstrator platform that will be used in future work. This model uses first-order dynamics for the drive-line

$$\dot{a}_i(t) = \frac{1}{\eta_i} \left(u_i(t) - a_i(t) \right), \tag{12}$$

in which $a_i(t)$ is the longitudinal acceleration, $u_i(t)$ the commanded longitudinal acceleration, and η_i the time constant of the drive-line for vehicle i. A new augmented input to the system $\nu_i(t) := \frac{1}{\eta_i}(u_i(t) - a_i(t))$ is defined, such that a virtual triple integrator system is obtained.

$$\dot{q}_i(t) = v_i(t) \tag{13}$$

$$\dot{v}_i(t) = a_i(t) \tag{14}$$

$$\dot{a}_i(t) = \nu_i(t) \tag{15}$$

In which $q_i(t)$, $v_i(t)$ and $a_i(t)$ are defined in the vehicle frame $\vec{r}^3$. Note the jerk of the planned trajectory can simply be substituted for $\nu_i(t)$. Then the input $u_i(t)$ to the original system can be computed as $u_i(t) = \eta_i \nu_i(t) + a_i(t)$. This shows another benefit of explicitly planning trajectories, rather than using a feedback strategy, as the drive-line dynamics can be compensated. For the longitudinal B-spline trajectories this planned jerk can be computed as long as $p \geq 3$. In addition, following the closed form transformations (that maps $[\mathbf{S}_i(t), \mathbf{\Lambda}_i(t), \mathbf{X}(s)] \rightarrow [q_i(t), v_i(t), a_i(t)]$) presented in [11], it is required that $\ell(t)$ is thrice continuously differentiable and the curvature of the reference path $\mathbf{X}(s)$ is twice continuously differentiable. In practice however, the contributions to motion in the vehicle frame $[q_i(t), v_i(t), a_i(t)]$, of the lateral trajectories $\mathbf{\Lambda}_i(t)$ are much smaller than that of $\mathbf{S}_i(t)$. Note that in case of model uncertainty in η_i, the receding horizon implementation of the

planner constitutes the feedback mechanism of the proposed planner.

D. Desired Trajectory

In order to enable safe short-distance vehicle following, a spacing policy that is commonly used in CACC applications is used [1]. This spacing policy describes the desired inter vehicle distance, $d_{i,r}(t)$ between the rear bumper of vehicle i with the rear bumper of vehicle $i-1$

$$d_{i,r}(t) = c_i + L_i + h_i \dot{s}_i(t), \tag{16}$$

$$\begin{aligned} e_i(t) &= d_i(t) - d_{i,r}(t) \\ &= s_{i-1}(t) - s_i(t) - (\tilde{c}_i + h_i \dot{s}_i(t)) \end{aligned} \tag{17}$$

in which $\tilde{c}_i = c_i + L_i$ combines the standstill distance c_i and vehicle length L_i, h_i the constant headway time and e_i the distance error. Note that these trajectories and spacing policies are defined in the moving frame $\vec{r}^2$.

It is assumed that information of the planned trajectory of the preceding vehicle, $s_{i-1}(t)$, is available in the form of a B-spline. The $n+1$ control points of this trajectory are denoted by $\boldsymbol{\mathcal{P}}_{i-1}$. In addition to the control points, the time instance, $t_{k,i-1}$, at which the preceding vehicle has planned its trajectory, as well as the time horizon, T_{i-1}, are received such that the clamped uniform knot vector as in (11) can be reconstructed, assuming that all vehicles use a uniformly distributed knot vector, with the same degree p of the B-spline. Hence the complete trajectory of the preceding vehicle, $\mathbf{S}_{i-1}(t)$ can be evaluated, by communicating only $n+3$ values. In contrast, methods using Model Predictive Control or sampled trajectories, assuming a control frequency f_c, would require instead to communicate $f_c \cdot T_i$ values to achieve the same. Typical values are $f_c = 100$ Hz, $T_i = 5$ s and $n = 6$, resulting in 500 values that need to be communicated in the case of MPC and only 9 values for the proposed planning method.

Next, to construct the desired trajectory for the host vehicle i, spacing policy (16) is used. In traditional CACC applications, a feedback controller is used to regulate the error, e_i, towards zero. Along the same line of thought, the B-spline is manipulated to ensure the planned trajectories minimize this error as well. For a given time, the value of the basis functions for a given order can be computed and combined into a row vector

$$\boldsymbol{\mathcal{B}}_{n,p}(t) = \begin{bmatrix} N_{0,p}(t) & N_{1,p}(t) & \dots & N_{n,p}(t) \end{bmatrix} \tag{18}$$

such that (10) can be rewritten to

$$s_i(t) = \boldsymbol{\mathcal{B}}_{n,p}(t)\boldsymbol{\mathcal{P}}_i \tag{19}$$

$$\dot{s}_i(t) = \boldsymbol{\mathcal{B}}_{n-1,p-1}(t)\boldsymbol{\Delta}_{10}\boldsymbol{\mathcal{P}}_i \tag{20}$$

$$\ddot{s}_i(t) = \boldsymbol{\mathcal{B}}_{n-2,p-2}(t)\boldsymbol{\Delta}_{20}\boldsymbol{\mathcal{P}}_i \tag{21}$$

Substituting (19) and (20) in (17) yields

$$e_i(t) = s_{i-1}(t) - \tilde{c}_i - [\boldsymbol{\mathcal{B}}_{n,p}(t) + h_i \boldsymbol{\mathcal{B}}_{n-1,p-1}(t)\boldsymbol{\Delta}_{10}]\boldsymbol{\mathcal{P}}_i \tag{22}$$

Time instances are used

$$t_j^* \in [t_{k,i}, t_{k,i} + T_i], \quad j = \{1, 2, \dots\} \tag{23}$$

to construct a vector $\mathcal{T}$ for which holds that $t_j^* < t_{j+1}^*$. For each element in this vector the error in (22) is computed and the objective is to minimize the cumulative error

$$J = \sum_{\mathcal{T}} e_i(t_j^*)^2 \tag{24}$$

To solve this minimization problem, (22) can be equated to zero for each time instance in $\mathcal{T}$, to construct a linear system of equations which can be solved for $\boldsymbol{\mathcal{P}}_i$. Note that in order for the solution to be unique, t^* should satisfy the Schoenberg-Whitney's Condition [25], [23],

$$u_j \leq t_j^* \leq u_{j+p} \tag{25}$$

Thus, the selection of the time instances in $\mathcal{T}$ is important. Since the error of (22) should be small over the entire time horizon, samples over the entire horizon should be included. Note that the coordinates of the control points of (6), satisfy both the Schoenberg-Whitney's condition, and span the entire spline. Therefore, $t_j^* = \mu_j$ is used in the vector of time instances $\mathcal{T}$.

Planned trajectories start in the current vehicle state as described earlier. By construction, $N_{0,p}(t_{k,i}) = 1, \forall p$ and $N_{k,p}(t_{k,i}) = 0, k > 0$. The control points $P_{j,i}, j \in \{0,1,2\}$ are fixed to correspond to a given initial condition via (19), (20), (21) and (9) and can be computed via

$$\begin{aligned} \begin{bmatrix} P_{0,i} & P_{1,i} & P_{2,i} \end{bmatrix}^\intercal &= \mathbf{C}^{-1}\mathbf{S}_i(t_{k,i}) \\ \begin{bmatrix} \mathbf{C} & \mathbf{0}_{3\times(n-2)} \end{bmatrix} &= \begin{bmatrix} \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} \\ \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}\boldsymbol{\Delta}_{10} \\ \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}\boldsymbol{\Delta}_{20} \end{bmatrix} \end{aligned} \tag{26}$$

with $\mathbf{C} \in \mathbb{R}^{3\times3}$. These first three control points largely control the initial response. Therefore, only the time instances $\mathcal{T} = [\mu_3, \mu_4, \dots, \mu_n]$ are used, as the coordinates of the free control points, such that the cost function $J = \sum_{j=3}^{n} e_i(\mu_j)^2$ is minimized. The control points can be found by solving a system of equations of (22)

$$\begin{aligned} \begin{bmatrix} P_{3,i} \\ \vdots \\ P_{n,i} \end{bmatrix} &= \boldsymbol{\Omega}_{12}^{-1}\left(\begin{bmatrix} s_{i-1}(\mu_3) - \tilde{c}_i \\ \vdots \\ s_{i-1}(\mu_n) - \tilde{c}_i \end{bmatrix} - \boldsymbol{\Omega}_{11} \begin{bmatrix} P_{0,i} \\ P_{1,i} \\ P_{2,i} \end{bmatrix} \right) \\ \begin{bmatrix} \boldsymbol{\Omega}_{11} & \boldsymbol{\Omega}_{12} \end{bmatrix} &= \begin{bmatrix} \boldsymbol{\mathcal{B}}_{n,p}(\mu_3) + h_i \boldsymbol{\mathcal{B}}_{n-1,p-1}(\mu_3)\boldsymbol{\Delta}_{10} \\ \vdots \\ \boldsymbol{\mathcal{B}}_{n,p}(\mu_n) + h_i \boldsymbol{\mathcal{B}}_{n-1,p-1}(\mu_n)\boldsymbol{\Delta}_{10} \end{bmatrix} \end{aligned} \tag{27}$$

where $\boldsymbol{\Omega}_{11}$, $\boldsymbol{\Omega}_{12}$ are matrix partitions of corresponding sizes. More timestamps could be included in the optimization, in which case a Moore-Penrose inverse should be used. Because using the selection for t^* for minimization results in the B-spline minimizing the error over the entire horizon, overshoot is likely to be prevented for our application. This makes it suitable to obtain string stability.

E. Design Considerations

In practice, the planner will be implemented in a receding horizon fashion that updates at fixed time intervals t_p, such that $t_{k+1,i} = t_{k,i} + t_\mathrm{p}$. The design objective of temporal consistency can be achieved by means of a suitable minimization criterium, due to the principle of optimality. Ideally, this minimization remains identical when the time horizon is extended. Note that (17) indeed satisfies this criteria.

Additionally, exact temporal consistency can only be realised if the basis functions are identical. This is generally not the case, unless the new knot vector is shifted backwards over t_p to coincide with the knot vector of the previous planning step. However, this would require additional overhead in keeping track of how the basis functions should be defined. Hence, exact temporal consistency is sacrificed in favour of simple implementation.

An important consideration is the degree p of the B-spline, and the number of control points $n+1$ that is used. The smoothness properties of [16] are retained for comfort reasons, by choosing $p = 5$. For n, note that for a given horizon T_i, using more control points allows the method to regulate the error faster at the expense of larger control inputs and a higher computational burden on the CPU.

Note that the usual problem of gap closing, where aggressive acceleration is commanded by a CACC controller, can be easily avoided in this framework. A more comfortable approaching trajectory can be constructed by modifying the spacing policy as shown in [26]. However, in the remainder of this paper, the spacing policy is kept constant to more easily evaluate the behavior.

F. Communication Delay

Traditionally, one of the major bottlenecks in CACC controller design has been the communication delay. This delay impedes the performance of the feedback controller, since information about the desired acceleration of the preceding vehicle is available to the host vehicle later. Naturally, using a different control strategy does not remove this delay as it is the result of the hardware that is used in the system. However, as will be demonstrated, the trajectory planning approach is more robust against a communication delay $\zeta(t)$. This is due to the communicated information not only containing the current desired acceleration, but also the desired acceleration over an entire time horizon. Nevertheless, it is still required to take communication delay into account, as the communicated trajectory now no longer contains the complete time horizon that the host vehicle needs to plan over. This is illustrated in Fig. 5, where the communicated trajectory of the preceding vehicle can be seen. Note that the horizon of the host vehicle cannot simply be truncated to match that of the preceding vehicle, as this would result in vehicles upstream ending up with ever smaller time horizons. The same problem occurs if the planning horizon of the preceding vehicle is smaller than that of the host vehicle (e.g., $T_i > T_{i-1}$).

To ensure that the host vehicle is still capable of constructing a trajectory over its time horizon, $t \in [t_{k,i}, t_{k,i} + T_i]$, some options are available. One option would be to change the time vector t^*, which contains the samples over which the problem

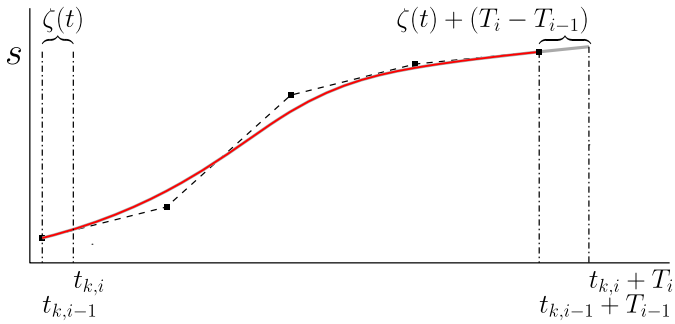


Fig. 5. B-spline trajectory of predecessor is extended in case of communication delay $\zeta(t)$, as information is not included over the entire time horizon of the host vehicle, $t \in [t_{k,i}, t_{k,i} + T_i]$.

given by (22) is minimized. However, for large mismatches in time horizon $T_i - T_{i-1}$, this could result in a violation of the Schoenberg-Withney relation (25). The other option is to simply extend the B-spline trajectory by means of prediction. A constant acceleration is assumed for the extended part of the trajectory. A simple prediction $\hat{s}_{i-1}(\theta)$, for time samples θ for which holds $(t_{k,i-1} + T_{i-1}) < \theta$, can be computed as:

$$\tau_i = t_{k,i} + T_i, \qquad \Delta t = \theta - \tau_{i-1}$$

$$\hat{s}_{i-1}(\theta) = s_{i-1}(\tau_{i-1}) + \Delta t \dot{s}_{i-1}(\tau_{i-1}) + \frac{1}{2}\Delta t^2 \ddot{s}_{i-1}(\tau_{i-1})$$

where (9) can be used to compute

$$\dot{s}_{i-1}(\tau_{i-1}) = [0 \quad \dots \quad 0 \quad 1]\boldsymbol{\Delta}_{10}\boldsymbol{\mathcal{P}}_{i-1}$$
$$\ddot{s}_{i-1}(\tau_{i-1}) = [0 \quad \dots \quad 0 \quad 1]\boldsymbol{\Delta}_{20}\boldsymbol{\mathcal{P}}_{i-1}$$

IV. Simulation Results

This section demonstrates the effectiveness of this algorithm for path planning. To this extent, various important scenarios for a platoon of vehicles are implemented. The lead vehicle uses B-spline trajectories, that transition to a new velocity with minimum jerk. Note that these correspond with the quartic polynomial trajectories from [11], [16]. This trajectory for the lead vehicle is represented by a B-spline with $p=5$, $T_0=5$ s, knot according to (11) and control points $\boldsymbol{\mathcal{P}}_{0,0} = [0 \quad 10 \quad 30 \quad 48.75 \quad 65 \quad 80 \quad 87.5]$. This lead vehicle trajectory corresponds to a velocity change from $\dot{s}(0)=20\mathrm{ms}^{-1}$ to a velocity of $\dot{s}(5)=15\mathrm{ms}^{-1}$. Note that this trajectory updates every planning cycle, such that $\boldsymbol{\mathcal{P}}_{0,k}$ updates every timestep k, to correspond with a new initial condition. This is done in a similar manner as for all the following vehicles, except that for the lead vehicle, the desired trajectory is modified to match the trajectory planned in the previous planning cycle as closely as possible. Additionally, a terminal constraint of $\ddot{s}(t_{k,0} + T_0) = 0$ is applied to prevent oscillations in spline fitting. As such, the control points $\boldsymbol{\mathcal{P}}_{0,k}$ for the lead vehicle are solved by means of a minimization with equality constraint

$$\min \left\| \begin{bmatrix} \boldsymbol{\mathcal{B}}_{n,p}(\mu_0) \\ \vdots \\ \boldsymbol{\mathcal{B}}_{n,p}(\mu_n) \end{bmatrix} \boldsymbol{\mathcal{P}}_{0,k} - \begin{bmatrix} s_{0,k-1}(\mu_0) \\ \vdots \\ s_{0,k-1}(\mu_n) \end{bmatrix} \right\|_2$$

$$\text{s.t.} \begin{bmatrix} \mathbf{C} \\ \begin{bmatrix} 0 & \dots & 0 & 1 \end{bmatrix} \mathbf{\Delta}_{20} \end{bmatrix} \boldsymbol{\mathcal{P}}_{0,k} = \begin{bmatrix} \mathbf{S}_0(t_{k,0}) \\ 0 \end{bmatrix}, \quad (28)$$

where $\mu_j, j \in [3, \dots, n]$ is related to timestep k, $s_{0ak-1}(t)$ represents the values of planning step $k-1$ and where $\mathbf{C}$ is identical to that in (26). Also note that similarly to the case of communication delay, the trajectory planned in the previous timestep needs to be extrapolated. This is done by assuming a constant velocity at the value to which the velocity transition is made.

Note that in [16] the performance of the polynomial planner of [11] was already demonstrated and shown to be inherently string-unstable. A potential solution was also demonstrated in [16], however, this required to shift the time horizon for each vehicle in the string which would not be practical for long platoons. This is due to vehicle in the upstream direction taking a longer time to compensate for initial perturbations the further upstream these vehicles are.

A. Temporal Consistency

One of the control objectives noted in Section II-B was that of temporal consistency. This refers to how similar consecutively planned trajectories are. As already stated, in general, exact temporal consistency cannot be achieved due to the basis functions being shifted over time t_p. However, in practice it suffices to require that consecutive trajectories are close to each other (i.e., requiring that $\Gamma_{k,k+1}$ in (3) is 'small').

A simulation study was performed to analyse the performance indicators from Section II-B for a scenario in which the lead vehicle brakes from $\dot{s}(0) = 20\ \mathrm{ms}^{-1}$ to a velocity of $\dot{s}(5) = 15\ \mathrm{ms}^{-1}$. As spacing policy parameters, a time gap of $h_i = 0.5$ s, and standstill distance $c_i = 5$ m is used for all vehicles; all vehicle lengths are set to zero without loss of generality. For the B-spline trajectories, $p = 5$ and $n = 6$ are used. The trajectories are planned over a time horizon of $T_i = 5$ s $\forall i$. This simulation is depicted in Fig. 6 for a planning update interval of $t_\mathrm{p} = 0.2$ s. The first vehicle is illustrated by a black line, and upstream vehicle responses are indicated by an increasingly blue color. For each vehicle, all the planned trajectories are plotted over the entire horizon to show discrepancies between the consecutive planning cycles. The actual response of each vehicle, which is determined by the consecutive planned B-spline trajectories on time spans $t \in [t_{k,i}, t_{k,i} + t_\mathrm{p}]$, updated at intervals t_p, is illustrated by the color corresponding to that vehicle.

As can be seen from Fig. 6, the response from all vehicles is a smooth transition to the new velocity. The string of vehicles can be seen to not demonstrate string-unstable behaviour, due to the magnitude of acceleration decreasing over the vehicle index. Moreover, the consecutive planning cycles show (close to) temporal consistency. The new knots do not coincide with the previous knots since $u_{j+1} - u_j \neq t_\mathrm{p}$. Therefore the transitions of the lowest order basis functions also occur at slightly different times. Hence exact temporal consistency is not possible. However, the figure shows that both position and velocity remain close to the original trajectory.

This is confirmed in Fig. 7, which illustrates the aforementioned performance indicators for various values of planning update time t_p, for a platoon of 6 vehicles (figure illustrates the maximum over all vehicles). Clearly, for smaller update times, the difference between two consecutive planning cycles are smaller as can be seen from the black crosses. Additionally, since more trajectories span the same time instance, the largest difference between any of the planned trajectories is larger. Note that the difference in position for this manoeuvre is only 0.8% of the inter-vehicle distance of the presented manoeuvre, which is sufficiently small to be acceptable.

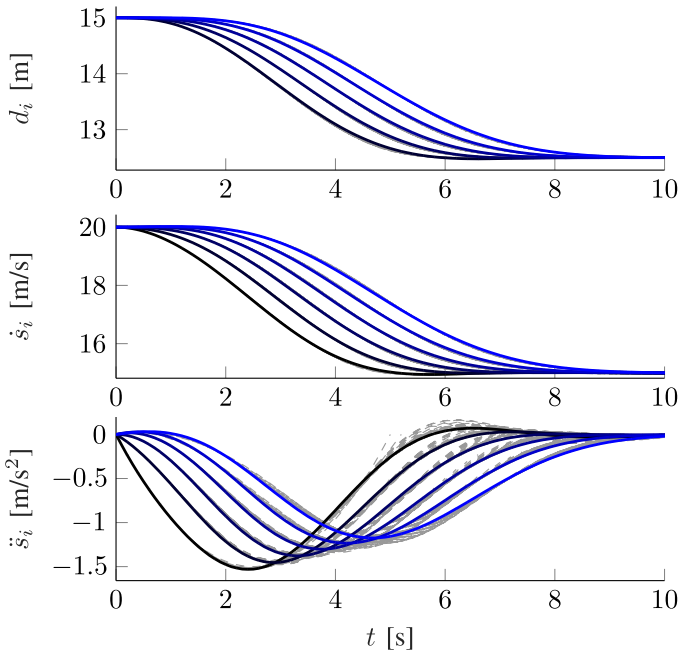


Fig. 6. Results of B-spline based trajectories for a velocity change in a platoon with six vehicles, $t_\mathrm{p} = 0.2$ s, lead vehicle (—), sixth vehicle (—), planned trajectories (- - -).

It should be noted that the update time t_p does not only affect temporal consistency, but will also affect the planning delay. The expected value of plan delay due to vehicles planning in an asynchronous fashion increases linearly with the value of t_p. Hence, t_p is best chosen as small as the on-board CPU allows in terms of computational capabilities. In the remainder of this section $t_\mathrm{p} = 0.2$ s.

B. Initial Perturbation

A second important test scenario is where the lead vehicle is driving at a constant velocity and parameters are identical to that of Section IV-A, but the platoon is not in steady state, since the position of the second vehicle has an offset with respect to its desired position. The proposed method of using polynomials [16] was not capable of handling these initial perturbations well, even resulting in collisions. However, as can be seen from Fig. 8, the newly proposed method of using B-splines handles the initial disturbance well. The initial errors are all regulated towards zero. Moreover, it can clearly be seen that the disturbance attenuates in upstream direction, with again the magnitude in acceleration decreasing for vehicles upstream. Hence, the response does not demonstrate string-unstable behaviour.

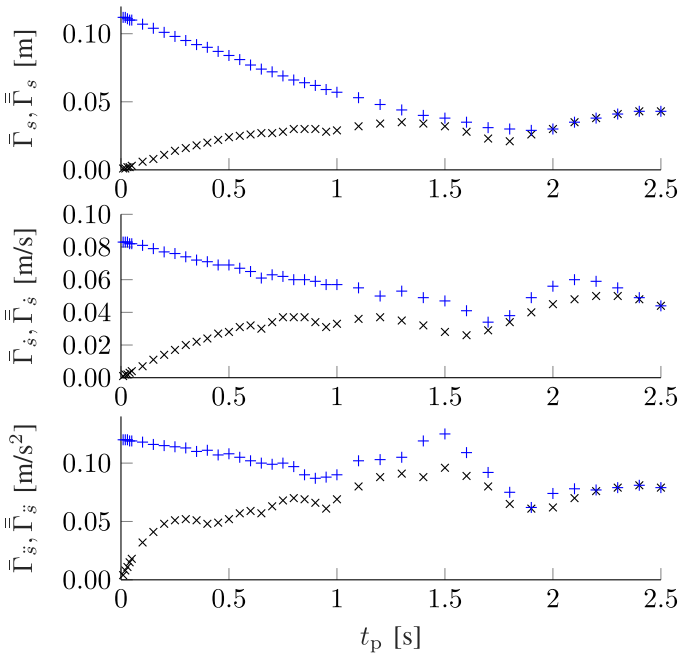


Fig. 7. Sensitivity of t_p on temporal consistency for $T_i = 5$ s. Maximum difference per element between consecutive trajectories $\bar{\Gamma}_l = \max_k \Gamma_{k,k+1,l}$ ($\times$), and all overlapping trajectories $\bar{\bar{\Gamma}}_l = \max_k(\max_j \Gamma_{k,j,l}), j \in \{k+1, \ldots, \lfloor k + T_i/t_\mathrm{p} \rfloor\}$ (+).

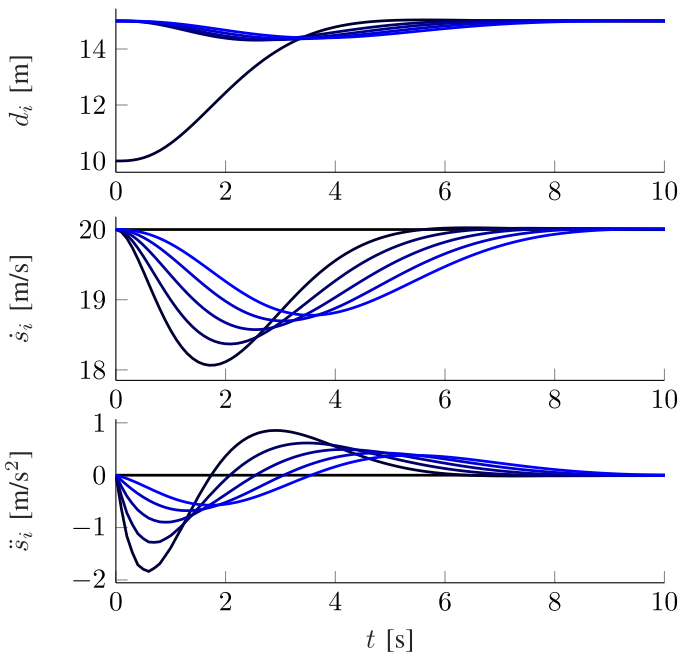


Fig. 8. Results of B-spline based trajectories for disturbance rejection, disturbance of $e_1(0) = 5$ m, in a platoon with six vehicles. lead vehicle (—), sixth vehicle (—).

C. Communication and Planning Delay

In reality, a time delay will always be present, due to communication delay and asynchronous planning updates. Therefore the same simulation as before is performed, but now information to the following vehicles is only available after a communication

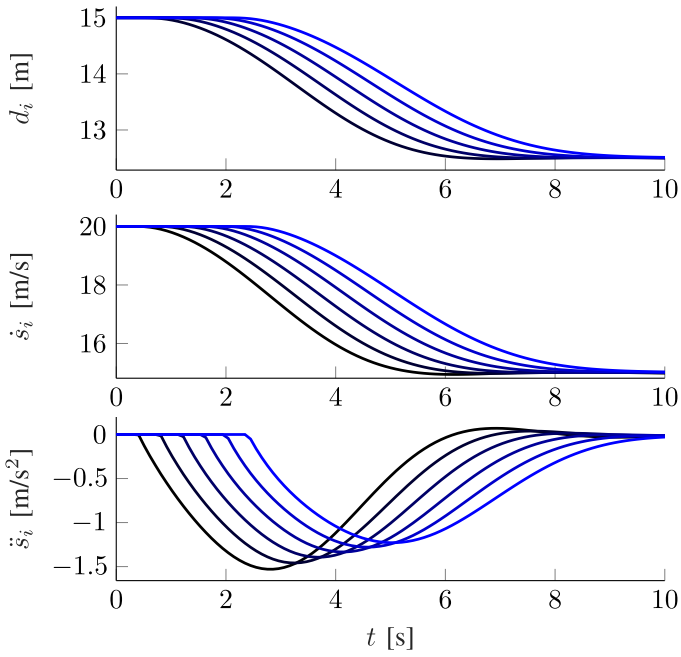


Fig. 9. Results of B-spline based trajectories for a velocity change, $\zeta(t) = 0.4$ s, $\forall t$, $h_i = 0.5$ s. Lead vehicle (—), sixth vehicle (—).

delay $\zeta(t)$. In the simulation, a constant delay is assumed, although this is not required for the planner, since the time instance at which the planning was computed is always communicated. In this case, $\zeta(t)$ does not only include the communication delay, but also the delay due to asynchronicity in the planning of consecutive vehicles, thus representing the total time between the plan update time, $t_{k,i}$, of the host vehicle, and the time instance, $t_{k,i-1}$, of the planning of the preceding vehicle that is available. This is illustrated in Fig. 5. The simulation is set up in such a way that the following vehicle is able to directly update its planning, once the communicated information becomes available.

A delay of $\zeta = 0.4$ s is used and the remaining parameters again those used in Section IV-A. The results are given in Fig. 9, which shows that the response still does not demonstrate string unstable behaviour, and gives nearly the same results as the non delayed case of Fig. 6. This result emphasizes the strength of this approach, as string-unstable behaviour is not observed despite significant time delays. It can be seen that the following vehicles initially keep planning to remain driving at the initial velocity, when the information of the velocity change has not been received yet.

It was noticed that the following three parameters are important in the resulting behaviour of the string: the planning horizon T_i, the time gap h_i, and the delay time ζ. To evaluate the performance, a numerical parameter study is performed to see for which combinations the platoon does not show string-unstable behaviour for the selected manoeuvres. As the time horizon is often the result of requirements for the autonomous planner, it is fixed to $T_i = 5$ s $\forall i$ as before, with an additional simulation study performed for a longer horizon $T_i = 10$ s $\forall i$ separately. For this study, several manoeuvres are simulated with a string of 50 vehicles. The manoeuvres that are tested are as follows. Velocity transitions:

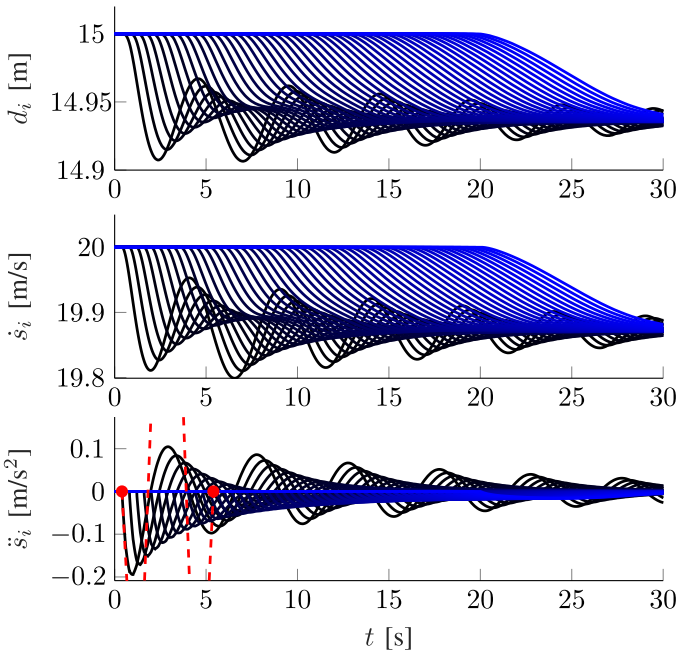
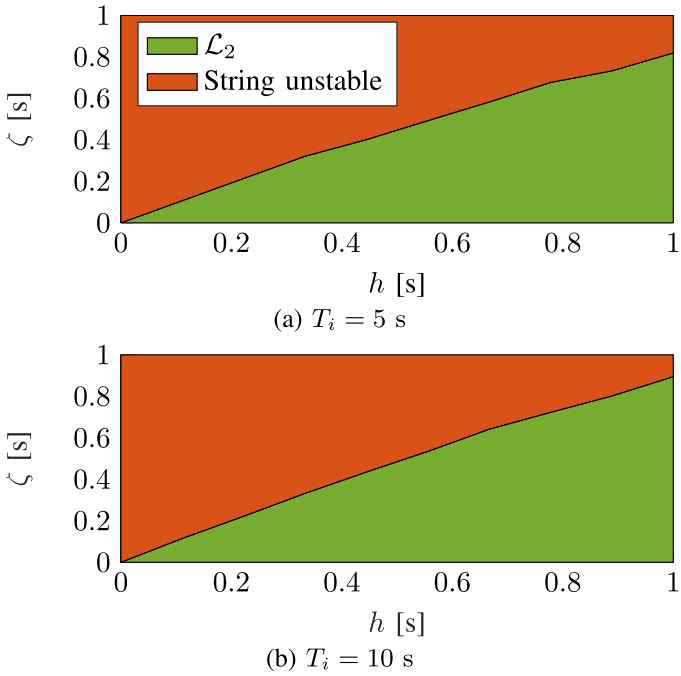


Fig. 10. Single scenario of 6, $\zeta(t) = 0.4$ s, $\forall t$, $h_i = 0.5$ s, control polygon of the random input $P^{(2)}_{j,1} \sim \mathcal{N}(0,1)$ (- • -). Lead vehicle (—), 50th vehicle (—).



(a) $T_i = 5$ s

(b) $T_i = 10$ s

Fig. 11. Behaviour of the vehicle string in scenarios 1-5 and thirty random scenarios of 6, for various delays ζ, and time gaps $h_i = h, \forall i$.

i) $\dot{s}_i(0) = 20$ ms^{-1}$\rightarrow \lim_{t\to\infty} \dot{s}_i(t) = 15$ ms^{-1}, $\forall i$, as illustrated in Fig. 9;
ii) $\dot{s}_i(0) = 20$ ms^{-1}$\rightarrow \lim_{t\to\infty} \dot{s}_i(t) = 1$ ms^{-1}, $\forall i$;
iii) $\dot{s}_i(0) = 5$ ms^{-1}$\rightarrow \lim_{t\to\infty} \dot{s}_i(t) = 15$ ms^{-1}, $\forall i$;

Driving at velocity $\dot{s}_i(0) = 20$ ms^{-1}, with an initial error for the second vehicle of:

iv) $e_1(0) = 5$ m (vehicle i=1 is further away than desired), as illustrated in Fig. 8;
v) $e_1(0) = -5$ m (vehicle $i = 1$ is closer than desired).
vi) Random acceleration input lead vehicle:
vi) $P^{(2)}_{j,1} \sim \mathcal{N}(0,1), j \in [1,\ldots,n-2]$, with $P^{(2)}_{n-1,1} = 0$

Scenarios i–iii are used to cover multiple velocities. Scenarios iv–v demonstrate more real world scenarios, where the velocity profile is determined by a vehicle correcting an error. Finally, twenty different random scenarios, described by vi, are used as worst case scenarios, covering a larger frequency range, where the acceleration control points of the first plan step are taken from the normal distribution $\mathcal{N}(0,1)$, with zero mean, and variance of one. The combination of the scenarios i–vi thus gives reliable indication of string-stability. After this first planning step, (28) is used for the remainder of the simulation. An example of such a scenario is illustrated in Fig. 10 , which shows the response of 50 vehicles, where the vehicle index is indicated by the color of the curve: black for the first vehicle, blue for vehicle number 50. It can be seen that the behavior in the scenario is string stable for the particular combination of delay ζ and headway time h. It should be noted that since the lead vehicle is assumed to plan trajectories by means of B-splines, the random choice of control points still results in smooth trajectories, as can also clearly be observed in Fig. 10.

A grid over h and ζ is used in combination with a bisection method to find parameter values for which the platoon does not demonstrate $\mathcal{L}_2$ string-unstable behaviour for all simulated manoeuvres i–iv, and twenty scenarios of vi, according to (1) for the acceleration $\ddot{s}_i(t)$. For a given time gap h, the maximum delay ζ for which the simulated platoon does not demonstrate string-unstable behavior, is then shown in Fig. 11. For very large delays, the error that has accumulated over the delay becomes too large to be compensated with a sufficiently small acceleration, resulting in string unstable behaviour. However, the delay at which this happens is far larger then in the traditional CACC controller from for example [1]. Clearly, by utilizing future information, string-unstable behaviour can still be prevented with large delays as indicated by the green area. For a larger time horizon, a small advantage can be seen, as string unstable behaviour can be prevented with a slightly larger time delay as was the case for the smaller time horizon.

V. Conclusion and Outlook

This paper presents a novel method of using B-splines in trajectory planning for cooperative automated vehicles. The presented method allows us to merge the framework of autonomous vehicles, in which trajectories are explicitly planned and checked before executing, with the framework of cooperative vehicle following, for which string stability is realized aided by communication. The presented method shows that B-splines can be used to plan trajectories for cooperative vehicles, and as such, shift the task of satisfying string stability from the feedback controller to the trajectory planner layer. The method does not need an iterative solver for optimization and as such, is likely to satisfy constraints on the computational time. Moreover the desired trajectory can be communicated efficiently due to the representations as B-splines, which keeps the required bandwidth low. Practical temporal consistency was shown to occur for the

selected cost function. The planning algorithm shows promising results in the presented simulations in terms of string stability.

Even in the presence of large communication delays, the proposed method shows string-stable behaviour in the presented simulations. This is due to every communicated message not only containing the current desired acceleration, but instead a complete planned trajectory. A more rigorous analysis of string-stability is needed to formally claim string-stability of the proposed planner framework. This could also provide insight whether or not string stability in terms of acceleration attenuation also implies string stability with respect to velocity. Experimental validation on full scale vehicles is left for future work.

REFERENCES

[1] J. Ploeg, B. T. Scheepers, E. van Nunen, N. van de Wouw, and H. Nijmeijer, "Design and experimental evaluation of cooperative adaptive cruise control," in *Proc. 14th Int. IEEE Conf. Intell. Transp. Syst.*, 2011, pp. 260–265.

[2] T. van der Sande and H. Nijmeijer, "From cooperative to autonomous vehicles, in *Sens. Control Auton. Veh., Ser. Lecture Notes Control Inf. Sci*. Berlin, German: Springer, 2017, pp. 435–452.

[3] J. I. Ge and G. Orosz, "Dynamics of connected vehicle systems with delayed acceleration feedback," *Transp. Res. Part C: Emerg. Technol.*, vol. 46, pp. 46–64, 2014. [Online]. Available: https://www.sciencedirect.com/journal/transportation-research-part-c-emergingtechnologies/issues

[4] A. I. M. Medina, N. van de Wouw, and H. Nijmeijer, "Automation of a T-intersection using virtual platoons of cooperative autonomous vehicles," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, 2015, pp. 1696–1701.

[5] A. Bayuwindra, Ø. L. Aakre, J. Ploeg, and H. Nijmeijer, "Combined lateral and longitudinal CACC for a unicycle-type platoon," in *Proc. IEEE Intell. Veh. Symp.*, 2016, pp. 527–532.

[6] J. Ploeg, N. van de Wouw, and H. Nijmeijer, "Lp string stability of cascaded systems: Application to vehicle platooning," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 2, pp. 786–793, Mar. 2014.

[7] P. Ioannou and C. Chien, "Autonomous intelligent cruise control," *IEEE Trans. Veh. Technol.*, vol. 42, no. 4, pp. 657–672, Nov. 1993.

[8] "Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems," SAE Int., Warrendale, PA, USA, 2018.

[9] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A review of motion planning for highway autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 1826–1848, May 2020.

[10] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Trans. Intell. Veh.*, vol. 1, no. 1, pp. 33–55, Mar. 2016.

[11] M. Werling, S. Kammel, J. Ziegler, and L. Gröll, "Optimal trajectories for time-critical street scenarios using discretized terminal manifolds," *Int. J. Robot. Res.*, vol. 31, no. 3, pp. 346–359, 2012.

[12] M. Mcnaughton, "Parallel algorithms for real-time motion planning." Ph.D dissertation, Carnegie Mellon Univ., The Robotics Inst., Pittsburgh, PA, USA, 2011.

[13] J. Ziegler and C. Stiller, "Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2009, pp. 1879–1884.

[14] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1135–1145, Apr. 2016.

[15] S. Gao, A. Lim, and D. Bevly, "An empirical study of DSRC V2V performance in truck platooning scenarios," *Digit. Commun. Netw.*, vol. 2, no. 4, pp. 233–244, 2016.

[16] R. van Hoek, J. Ploeg, and H. Nijmeijer, "Motion planning for automated connected vehicles," in *Proc. 14th Int. Symp. Adv. Veh. Control*, 2018, pp. 1–6.

[17] R. Kianfar, P. Falcone, and J. Fredriksson, "A receding horizon approach to string stable cooperative adaptive cruise control," in *Proc. 14th Int. IEEE Conf. Intell. Transp. Syst.*, 2011, pp. 734–739.

[18] E. van Nunen, J. Verhaegh, E. Silvas, E. Semsar-Kazerooni, and N. van de Wouw, "Robust model predictive cooperative adaptive cruise control subject to V2V impairments," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst.*, 2017, pp. 1–8.

[19] E. van Nunen, J. Reinders, E. Semsar-Kazerooni, and N. van de Wouw, "String stable model predictive cooperative adaptive cruise control for heterogeneous platoons," *IEEE Trans. Intell. Veh.*, vol. 4, no. 2, pp. 186–196, Jun. 2019.

[20] J. Ploeg, D. P. Shukla, N. van de Wouw, and H. Nijmeijer, "Controller synthesis for string stability of vehicle platoons," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 854–865, Apr. 2014.

[21] T. Mercy, R. V. Parys, and G. Pipeleers, "Spline-based motion planning for autonomous guided vehicles in a dynamic environment," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 6, pp. 2182–2189, Nov. 2018.

[22] L. Piegl and W. Tiller, *The NURBS Book*, 2nd ed., Ser. *Monographs in Visual Communication*. Berlin, Germany: Springer-Verlag, 1997, pp. 1–646.

[23] C. de Boor, *A Practical Guide to Splines*, vol. 27, Revised ed., Ser. *Applied Mathematical Sciences*. New York, NY, USA: Springer, 2001, pp. 1–348.

[24] J. Ploeg, "Analysis and design of controllers for cooperative and automated driving," Ph.D dissertation, Eindhoven Univ. of Technol., Eindhoven, The Netherlands, 2014.

[25] R. Goldenthal and M. Bercovier, "Spline Curve Approximation and Design by Optimal Control Over the Knots," in *Geometric Modelling*, S. Hahmann *et al.*, Eds., Vienna: Springer, 2004, pp. 53–64.

[26] R. van Hoek, J. Ploeg, and H. Nijmeijer, "Gap closing for cooperative driving in automated vehicles using B-splines for trajectory planning," in *Proc. 31st IEEE Intell. Veh. Symp.*, 2020, pp. 370–375.

Robbin van Hoek received the B.Sc. and M.Sc. degrees in 2013 and 2016, respectively, in mechanical from the Eindhoven University of Technology, Eindhoven, The Netherlands, where he is currently working toward the Ph.D. degree in mechanical engineering. His research interests include vehicle dynamics, vehicle control, and control systems for cooperative and automated vehicles. His current work focusses on the integration of vehicle cooperation in automated vehicles.

Jeroen Ploeg received the M.Sc. degree in mechanical engineering from the Delft University of Technology, Delft, The Netherlands, in 1988 and the Ph.D. degree in mechanical engineering on the control of vehicle platoons from the Eindhoven University of Technology, Eindhoven, The Netherlands, in 2014. He is currently with 2getthere B.V., Utrecht, The Netherlands, were he leads the research and development activities in the field of cooperative automated driving for automated transit systems, in particular platooning. Since 2017, he has been a part-time Associate Professor with the Mechanical Engineering Department, Eindhoven University of Technology.

From 1999 until 2017, he was with TNO, Helmond, The Netherlands, ultimately as a Principal Scientist in the field of vehicle automation and road safety assessment. His research interests include control system design for cooperative and automated vehicles, in particular string stability of vehicle platoons, the design of interaction protocols for complex driving scenarios, and motion control of wheeled mobile robots.

Henk Nijmeijer (Fellow, IEEE) was born in 1955. In January 2015, he was the Scientific Director of the Dutch Institute of Systems and Control, Delft, The Netherlands,. He is currently a Full Professor with Eindhoven University of Technology, Eindhoven, The Netherlands, and he Chairs the Dynamics and Control Group. He has authored or coauthored a large number of journal and conference papers, and several books, and he is or was on the Editorial Board of numerous journals. Since 2011, he has been an IFAC Council Member. He is a Member of the Mexican Academy of Sciences. In 1990, he received the IEE Heaviside Premium. He was the recipient of the 2015 IEEE Control Systems Technology Award. In 2011, he was appointed as an Honorary Knight of the Golden Feedback Loop. He is the Editor of the *Communications in Nonlinear Science and Numerical Simulations*. He is the Graduate Program Director of the TU/e Automotive Systems program.