HAL
open science

An Algorithm for Supervised Driving of Cooperative Semi-Autonomous Vehicles

Florent Altché, Xiangjun Qian, Arnaud de La Fortelle

To cite this version:

Florent Altché, Xiangjun Qian, Arnaud de La Fortelle. An Algorithm for Supervised Driving of Cooperative Semi-Autonomous Vehicles. IEEE Transactions on Intelligent Transportation Systems, 2017, 18 (12), pp.3527 - 3539. 10.1109/TITS.2017.2736532 . hal-01691892

HAL Id: hal-01691892

https://minesparis-psl.hal.science/hal-01691892v1

Submitted on 24 Jan 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An Algorithm for Supervised Driving of Cooperative Semi-Autonomous Vehicles (Extended)

Florent Altché, Xiangjun Qian, and Arnaud de La Fortelle,

***Abstract*—Before reaching full autonomy, vehicles will gradually be equipped with more and more advanced driver assistance systems (ADAS), effectively rendering them semi-autonomous. However, current ADAS technologies seem unable to handle complex traffic situations, notably when dealing with vehicles arriving from the sides, either at intersections or when merging on highways. The high rate of accidents in these settings prove that they constitute difficult driving situations. Moreover, intersections and merging lanes are often the source of important traffic congestion and, sometimes, deadlocks. In this article, we propose a cooperative framework to safely coordinate semi-autonomous vehicles in such settings, removing the risk of collision or deadlocks while remaining compatible with human driving. More specifically, we present a *supervised coordination* scheme that overrides control inputs from human drivers when they would result in an unsafe or blocked situation. To avoid unnecessary intervention and remain compatible with human driving, overriding only occurs when collisions or deadlocks are imminent. In this case, safe overriding controls are chosen while ensuring they deviate minimally from those originally requested by the drivers. Simulation results based on a realistic physics simulator show that our approach is scalable to real-world scenarios, and computations can be performed in real-time on a standard computer for up to a dozen simultaneous vehicles.**

***Index Terms*—Semi-autonomous driving, safety, supervisor, supervised driving.**

I. Introduction

ADVANCED driver assistance systems (ADAS) are becoming increasingly complex as they spread across the automotive market. Although adaptive cruise control (ACC) [1] and automated emergency braking (AEB) [2] are the best-known examples of such systems, applications of ADAS have been broadened and now include pedestrian [3], traffic light [4] or obstacle detection [5] as well as lane keeping assistance [6]. The development of this new equipment allows drivers to delegate part of the driving task to their vehicles. As these systems keep getting more efficient and able to handle more complex situations, vehicles will gradually progress towards *semi-autonomous* driving, where drivers remain in charge of their own safety, while their errors can be seamlessly corrected to prevent potential accidents.

One of the challenges of semi-autonomous driving lies in efficiently handling vehicles on conflicting paths, for instance at an intersection or a highway entry lane. Traffic rules such as priority to the right can help determine whether to pass before or after another vehicle; however, many situations require driving experience to be handled efficiently. Learning-based approaches may eventually prove able to transfer driving experience to a computer, but such knowledge is very hard to implement in a safety system. In this article, we consider another possible solution, consisting in using vehicle-to-vehicle or vehicle-to-infrastructure communication for cooperative semi-autonomous driving. In this setting, vehicles negotiate with one another, or receive instructions from a centralized computer, allowing them to drive safely and efficiently.

In this article, we consider a method to ensure the safety of multiple semi-autonomous vehicles on conflicting paths, for instance crossing an intersection or entering a highway, while remaining compatible with the presence of human drivers. To this end, and inspired by earlier work in [7], [8], we propose a so-called *Supervisor* which monitors control inputs from each vehicle's driver, and is able to override these controls when they would result in an unsafe situation. More specifically, the role of the supervisor is twofold: first, knowing the current states of the vehicles, the supervisor should determine if the controls requested by the drivers would lead the vehicles into unsafe *inevitable collision states* [9]. In this case, the second task of the supervisor is to compute safe controls – maintaining the vehicles in safe states – which are as close as possible to those actually requested by the drivers. We say that such a control is *minimally deviating*.

This paper provides two main contributions: from a practical standpoint, we design and implement a mathematical framework allowing to simultaneously perform the safety verification of target control inputs, and the computation of minimally deviating safe controls if target inputs are unsafe. From a theoretical standpoint, we formally prove that verifying safety over a finite time horizon is enough to ensure infinite horizon safety, and we provide a sufficient condition on the verification horizon for this property to hold. Unlike previous work focusing on specific situations such as intersections [7], [8], our framework can be applied to a wide variety of driving scenarios including intersections, merging lanes and roundabouts.

The rest of the article is structured as follows: in Section II, we provide a review of the related literature. In Section III, we present our modeling of semi-autonomous vehicles and introduce the *Supervision problem* of verifying the safety of drivers control inputs and finding a minimally deviating safe control if necessary. In Section IV, we present an infinite horizon formulation based on constraints programming to solve this problem. In Section V, we derive a finite horizon formulation which we prove is equivalent to the infinite

F. Altché, X. Qian and A. de La Fortelle are with MINES ParisTech, PSL Research University, Centre for robotics, 60 Bd St Michel 75006 Paris, France (e-mail: `florent.altche@mines-paristech.fr`.)

F. Altché is also with École des Ponts ParisTech, Cité Descartes, 6-8 Av Blaise Pascal, 77455 Champs-sur-Marne, France.

horizon one. In Section VI, we use computer simulations to showcase the performance of the proposed supervisor in various driving situations. In Section VII, we present possible methods for real-world implementations of our approach. Finally, Section VIII concludes the study.

II. Related Work

In the last decade, a lot of research has been focused on coordinating fully autonomous vehicles in challenging settings such as crossroads, roundabouts or merging lanes, with the ambition of improving both safety and traffic efficiency. Naumann et al. [10], followed by Dresner and Stone [11] have seemingly pioneered the work of adapting traffic intersections management methods to fully autonomous vehicles, designing so-called *autonomous intersection management* algorithms. They propose that each approaching autonomous vehicle reserves a time interval to cross the intersection; collisions are prevented by ensuring that conflicting vehicles are assigned non-overlapping crossing times. Subsequent studies on this particular problem have led to other approaches. In [12], vehicles choose their control inputs based on navigation functions which include a collision avoidance term, allowing vehicles to react to maneuvers from other traffic participants. In [13], collision avoidance is ensured by assigning relative crossing orders to incoming vehicles; each vehicle then uses model predictive control to plan collision-free trajectories respecting these priorities. Other authors have considered different driving situations for autonomous vehicles, such as cooperative merging on a highway [14]–[16], or entering a roundabout [17].

By contrast, relatively little work has considered semi-autonomous driving assistance, possibly because the presence of human drivers brings a lot of additional complexity. The goal of a semi-autonomous driving assistant is to help the driver avoid collisions, either by notifying of a potential danger [18] or by taking over vehicle control in dangerous situations [19]–[21]. To be accepted by human drivers, such systems should be as unobtrusive as possible, and in particular should only intervene when necessary. Most of the currently existing literature on semi-autonomous driving mostly focuses on highway driving [19]–[21], which presents relatively low difficulty as vehicles trajectories remain mostly parallel. The aim of this article is to bring semi-autonomy one step further, to allow cooperative driving between semi-autonomous vehicles in more complex conflict situations.

Some of these more complex problems have already been studied in the literature. In [22], the authors consider *semi-autonomous* driving at an intersection and propose that human drivers let an automated system control their vehicle while crossing said intersection. However, this scheme is rather intrusive as drivers completely relinquish control for a time, and handing back controls to a potentially distracted driver poses problems by itself. Colombo et al. [7], [8] introduced the idea of a supervisory instance (called *supervisor*) tasked with preventing the system of vehicles from entering undesirable states by overriding the controls of one or several vehicles. In this more human-friendly approach, overriding only occurs when necessary, *i.e.* if an absence of intervention would result in a crash. The question of determining whether overriding is needed or not, called *verification problem*, is NP-hard [23]; under several simplifying assumptions, it is shown in [7] to be equivalent to a *scheduling problem*. In this reformulation, vehicles are each assigned a time slot during which they are allowed inside the intersection, and assigned slots are mutually disjoint. If, due to vehicle dynamics, no feasible schedule exists, the initial state is deemed unsafe. This allows the authors to design a so-called *least restrictive* supervisor, which verifies the safety of the desired inputs and overrides them if necessary. However, the proposed supervisor is only suitable to simple intersection geometries with a single conflict point. Moreover, no additional property is required from the safe controls used for overriding, which can widely deviate from the desired ones.

Several variations have been proposed based on the equivalence demonstrated in [7]. Reference [24] designs a supervisor which is robust to bounded uncertainties by adding safety margins. Reference [25] leverages job-shop scheduling to develop a supervisor that considers several possible conflict points inside the intersection; however, vehicle dynamics are only modeled as first-order integrators, which is not realistic in a real-world setting. Campos et al. [8] proposed a Pareto-optimal supervisor leading to a minimally deviating formulation by recursively finding the most constrained vehicle, reserving its optimal crossing time, and scheduling the crossing of the remaining vehicles using the previous schedule as constraints. This method allows to minimize the deviation between the overridden and desired controls, but may be computationally intensive. Indeed, one of the major difficulties of performing optimization in this context lies in the necessity to consider all the possible orderings of the vehicles.

This problem is highly combinatorial; it has been shown that there exists up to $2^{n(n-1)/2}$ orderings for n vehicles [26]. Moreover, it is generally ignored by most authors studying motion planning problems, who either use simple heuristics such as first-come, first-served [11], [27] or rely on exhaustive search [8], [28]. A possible method to handle the combinatorial explosion is to use pruning techniques such as *branch-and-bound*, which avoid exploring branches of the decision tree that would provably yield suboptimal results. These methods are commonly used in mixed-integer linear (see, *e.g.*, [29], [30] for applications to motion planning) or quadratic programming (see, *e.g.*, [31]) problems, which combine continuous and discrete optimization. More general nonlinear methods have also been used in motion planning [32], [33], although their high computational difficulty generally requires linearization for effective resolution, as illustrated in [34]. To the best of the authors' knowledge, branch-and-bound methods have never been applied to semi-autonomous driving.

This article significantly differs from references [7], [8], [25]. Instead of using a scheduling approach, we formulate the supervision problem as a Mixed Integer Quadratic Programming (MIQP) problem, which can handle various geometries with multiple collision points such as multi-lane intersections, merging lanes or roundabouts. Our formulation only requires to consider a small, finite planning horizon, while previous

approaches [7], [8], [25] needed to schedule the crossing of all the considered vehicles. Furthermore, the MIQP formulation is highly flexible, allowing to take into account various constraints (*e.g.*, maximal turning speed) and different cost functions. Finally, the resolution of MIQP can leverage highly-optimized solvers [35], allowing real-time implementations even for a relatively large number of vehicles.

This article expands the results presented in the conference paper [36]; among the significant improvements made in this extension, we now give a more comprehensive model of our vision of semi-autonomous vehicles and adjust the modeling of the problem to handle bounded control errors. We provide a detailed discussion on how complex road geometries with multiply-intersecting paths can be handled, leading to a very versatile framework. Finally, we extend the theoretical results to continuous arrivals of vehicles, and provide possible ways for actual implementation as a roadside unit.

III. Supervision problem

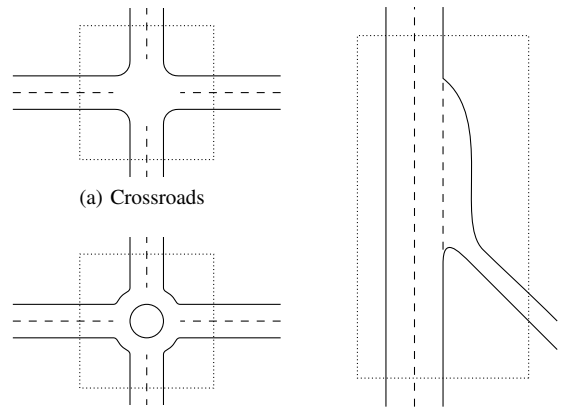
We consider the problem of safely coordinating multiple semi-autonomous vehicles on the road, in order to prevent collisions and deadlock situations where no vehicle is able to move forward. Since vehicles are human-driven, a form of outside supervision is necessary to prevent undesirable situations. This section presents our formulation of a so-called Supervision problem generalizing the work of Colombo et al. [7]; solving this problem yields a provably safe control, as close as possible to the original intentions of the drivers.

A. Modeling

1) Supervision area: We consider an isolated portion of a road infrastructure used by semi-autonomous vehicles, where some form of coordination is required to ensure vehicles safety. For instance, this could be a classical road intersection, a roundabout or an entry or weaving lane on a highway. We call this bounded portion of infrastructure the *supervision area* and we assume that vehicles can travel safely outside of the collision area using only their ACC capacities. In a real-world setting, different critical portions of infrastructure which are far enough apart can be considered individually, but need to be treated jointly if traffic from one can influence another. Figure 1 shows examples of roads configurations and the corresponding possible choice for a supervision area.

In this article, we present an embodiment of a Supervisor working over a spatially static supervision area over time, that can be thought of as a dedicated computer on the infrastructure or in the cloud. Vehicles are assumed to establish a connection to the supervisor when they enter the supervision area (using, for instance, V2I communication), and maintain it until they exit this region. We denote by $\mathcal{N}_t$ the set of vehicles currently inside the supervision area at a time t.

2) Semi-autonomous vehicles: We consider *semi-autonomous* vehicles equipped with advanced driver assistance systems, many of which are already commercially available, and Vehicle to Infrastructure (V2I) communication capacities. In particular, vehicles are assumed to have advanced cruise control, automated braking and lane keeping assistance



(a) Crossroads

(b) Roundabout

(c) Highway merging

Fig. 1. Examples of considered road configurations, and corresponding supervision areas (interior of the dotted rectangles).

systems such that accelerating, braking and steering can be actuated by an on-board computer. Moreover, we suppose that vehicles have access to reliable cartographic data and are capable of precisely measuring their current position, orientation and velocity with reference to a unique global frame, for instance using GNSS and inertial navigation.

Since the vehicles are not assumed to have advanced environment-sensing capacities, for instance based on LIDAR data, they are not able to handle all situations and still require a human driver to safely navigate, for instance in the case of on-road obstacles or loss of GNSS signal. Moreover, lateral collisions or deadlock situations can happen due to human error, justifying the need for supervision.

3) Parametrization: In the remainder of this article, we only consider the two-dimensional kinematics and dynamics of the vehicles. We denote by E_i a bounding polygon for the shape of vehicle $i \in \mathcal{N}_t$, and by c_i the center of E_i.

We assume that the geometry and lane markings of the roads inside the supervision area define a finite number of reference paths across this region, as exemplified in fig. 2. Due to the presence of a lane keeping assistance system, we assume that every vehicle is able to follow one of these reference paths with a small bounded lateral error. Noting γ_i the reference path of a vehicle i, we assume that the distance of c_i from γ_i is bounded from above by $\xi_i \geq 0$. Moreover, we assume that γ_i is at least $\mathcal{C}^2$-continuous, and that ξ_i is small enough to ensure, for all $x \in \mathbb{R}^2$,

$$d(x, \gamma_i) \leq \xi_i \Rightarrow \exists!\ y \in \gamma_i,\ ||x - y|| = d(x, \gamma_i). \qquad (1)$$

This condition allows to use the curvilinear position of the point of γ_i closest to c_i to uniquely encode the longitudinal position of vehicle i along γ_i. We denote by s_i this curvilinear position, with the convention that $s_i = 0$ when the front bumper of i first enters the supervision area and increases when i goes forward; we let s_i^{out} be the longitudinal position at which the rear bumper of i fully exits the supervision area.

4) Vehicle dynamics: In this article, we mostly focus on the longitudinal dynamics of the vehicles, and we let $x_i = (s_i, v_i)^T$ be the state of vehicle i, where s_i and v_i are respectively its longitudinal position and longitudinal speed. We

assume that vehicles follow second-order integrator dynamics with a bounded longitudinal error, and that the control input u_i corresponds to the longitudinal acceleration as:

$$\dot{x}_i = Ax_i + Bu_i, \tag{2}$$

where $A = \left(\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right)$ and $B = \left(\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right)$. Since we mostly consider situations with conflicting vehicles, we assume that human drivers maintain a relatively low speed (compared to the curvature of their path), which allows neglecting lateral dynamics and slip [37].

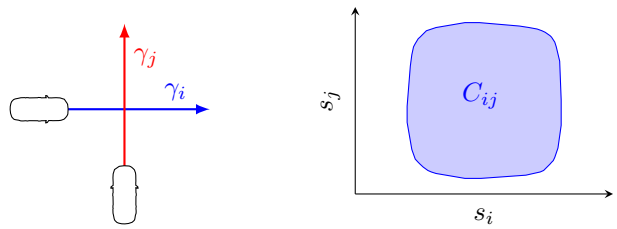
To account for speed limitations on the vehicles, each vehicle i is supposed to have a bounded non-negative velocity, so that $v_i \in [0, \overline{v}_i]$ (with $\overline{v}_i > 0$) at all times. Moreover, we assume that the acceleration u_i of each vehicle is bounded as $u_i \in [\underline{u}_i, \overline{u}_i]$, with $\underline{u}_i < 0 < \overline{u}_i$. These bounds can differ between vehicles, thus allowing heterogeneous vehicle performance. At a given time t, we let $\mathbf{U}_t = \prod_{i \in \mathcal{N}_t} [\underline{u}_i, \overline{u}_i]$ be the set of admissible accelerations for the vehicles of $\mathcal{N}_t$. We denote bt boldface $\mathbf{x} = (x_i)_{i \in \mathcal{N}_t}$ and $\mathbf{u} = (u_i)_{i \in \mathcal{N}_t}$ the state and control for the system of vehicles.

In what follows, we let $v_{max} > 0$ be a global upper bound for $\overline{v}_i$, $u_a > 0$ a lower bound for $\overline{u}_i$ and $u_b < 0$ an upper bound for $\underline{u}_i$ such that for all $t \geq t_\kappa$ and all $i \in \mathcal{N}_t$, $\overline{v}_i \leq v_{max}$ and $\underline{u}_i \leq u_b < 0 < u_a \leq \overline{u}_i$. Therefore, all vehicles are capable of braking with u_b and accelerating with u_a; finally, we let u_{max} be a global upper bound for $\overline{u}_i$.

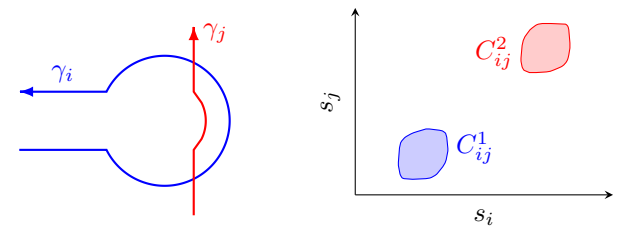
5) Collision regions: Finally, we assume that the angle between the orientation of vehicle i and the tangent to γ_i at its point closest to c_i is also bounded. With these hypotheses, for any pair of vehicles (i, j), we can compute the bounded set $\mathcal{C}_{ij}$ of curvilinear positions $(s_i, s_j) \subset [0, s_i^{out}] \times [0, s_j^{out}]$ for which a collision could happen between i and j. Note that these sets are "inflated" to take into account the bounded control errors. We call $\mathcal{C}_{ij}$ the *collision region* between i and j; fig. 2 shows examples of paths and corresponding computed collision regions for different driving situations. Note that collision regions can be empty or have one or multiple connected components. If $\mathcal{C}_{ij} \neq \emptyset$, we say that vehicles i and j are *conflicting*; when $\mathcal{C}_{ij}$ has multiple connected components, we denote by $\mathcal{C}_{ij}^p$ its p-th component, using the convention $\mathcal{C}_{ij}^p = \mathcal{C}_{ji}^p$.

6) No-stop regions: To prevent creating deadlock situations, vehicles are not allowed to stop when doing so would block traffic in other directions. To this extent, we define a *no-stop region* (see fig. 3) $\mathcal{D}_i$ for each vehicle $i \in \mathcal{N}_t$ as the smallest interval $\mathcal{D}_i = [\underline{s}_i^\perp, \overline{s}_i^\perp]$ containing all $\min\left(\Pi_{s_i} \mathcal{C}_{ij}^p\right)$ for all $t' \geq t$, $j \in \mathcal{N}_{t'}$ and all p such that $(0,0) \notin \mathcal{C}_{ij}^p$; in this formula, Π_{s_i} is the projection operator on the first coordinate. The no-stop region corresponds to the part of the supervision area where a vehicle may have to yield to another; if $\mathcal{C}_{ij}^p$ contains $(0,0)$, then either i or j enters the supervision area behind the other, in which case the relative ordering of the vehicles is given and the $\mathcal{C}_{ij}^p$ does not count in $\mathcal{D}_i$.

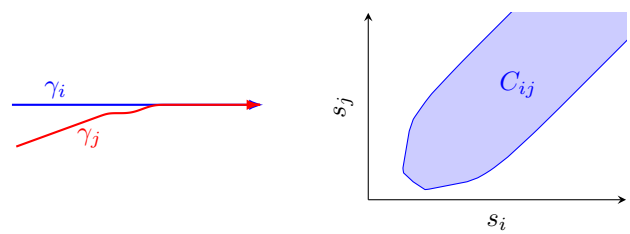
Note that, although this definition theoretically requires knowledge of all future vehicles, $\mathcal{D}_i$ can be computed off-line as a finite intersection of intervals provided that there only exists a finite number of possible paths inside the supervision area. In what follows, we let $v_{min} > 0$ be a minimum allowed speed for any vehicle inside its no-stop region, and we assume that $v_{min} \leq \overline{v}_i$ for all vehicles.



(a) Simple orthogonal intersection situation with example vehicle shapes



(b) Roundabout situation with multiple connected components in $\mathcal{C}_{ij}$



(c) Highway merging situation

Fig. 2. Examples of paths (left) and corresponding collision regions (right) for vehicles with the polygonal shape shown in fig. 2a.

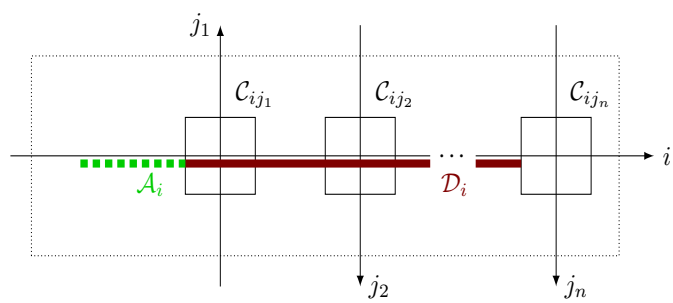


Fig. 3. Illustration of the no-stop region $\mathcal{D}_i$ and acceleration region $\mathcal{A}_i$ inside the supervision area (dotted rectangle).

For a no-stop region $\mathcal{D}_i$, we define the corresponding acceleration region $\mathcal{A}_i = [s_i^{acc}, \underline{s}_i^\perp]$ such that, if vehicle i is stopped at s_i^{acc}, it can reach a speed v_{min} before reaching $\underline{s}_i^\perp$. More specifically, we require that $0 \leq s_i^{acc} \leq \underline{s}_i^\perp - \frac{v_{min}}{2u_a}$ for all i. Inside the acceleration region, vehicles are only allowed to accelerate; this condition prevents vehicles from stopping right before the entrance of the no-stop region, leaving them unable to proceed forward due to the minimum speed requirement. Figure 3 illustrates an example of the no-stop regions and the corresponding acceleration regions.

7) Time discretization: Drivers continuously change the control input of their vehicle; however, due to computational

and communication constraints, it is impractical to handle functions of a continuous variable. In the remainder of this article, we choose a constant time step duration $\tau > 0$, and we assume that all vehicles use piecewise-constant controls with step τ, typically $0.5\,\mathrm{s}$. To simplify the formulation, we further assume that vehicles update their control simultaneously at times $t_\kappa = \kappa\tau$ for $\kappa \in \mathbb{N}$, and we denote by $\mathcal{U}_\tau(t_\kappa)$ the set of piecewise-constant admissible controls for the vehicles of $\mathcal{N}_{t_\kappa}$. By definition, for all $t \geq t_\kappa$ and all $\mathbf{u} \in \mathcal{U}_\tau(t_\kappa)$, $\mathbf{u}(t) \in \mathbf{U}_{t_\kappa}$.

B. Problem statement

Before presenting the so-called *supervision problem*, we first define the safety criterion for the vehicles inside the supervision area at a given time.

Definition 1 (Safe state)**.** *We say that the supervision area is in a safe state $\mathbf{x}^\kappa$ at time t_κ if there exists an admissible piecewise-constant control $\mathbf{u} \in \mathcal{U}_\tau(t_\kappa)$ defined over $[t_\kappa, +\infty[$ such that, under this control and starting from $\mathbf{x}^\kappa$, for all $t \geq t_\kappa$ and all $i, j \in \mathcal{N}_{t_\kappa}$, $(s_i(t), s_j(t)) \notin \mathcal{C}_{ij}$. Such a control is said to be a safe control.*

With this definition, the supervision area is in a safe state when all the vehicles inside this area can apply a dynamically admissible, infinite horizon control without a risk of collision. This safety condition corresponds to a contraposition of the notion of "inevitable collision state" proposed by Fraichard et al. [9]. In what follows, we denote by $\mathcal{U}_\tau^{safe}(t_\kappa)$ the set of safe and dynamically admissible piecewise-constant controls for the vehicles in $\mathcal{N}_{t_\kappa}$; by definition, a control $\mathbf{u} \in \mathcal{U}_\tau^{safe}(t_\kappa)$ is a piecewise-constant function from $[t_\kappa, +\infty[$ to $\mathbf{U}_{t_\kappa}$. We now define the safety condition for vehicles entering the supervision area.

Definition 2 (Safe entry)**.** *Consider a safe state $\mathbf{x}^\kappa$ at time t_κ and let $t_1 > t_\kappa$ be the first time at which a new vehicle enters the supervision area. We say that the vehicles of $\mathcal{N}_{t_1} \setminus \mathcal{N}_{t_\kappa}$ safely enter the supervision area with a margin τ if $t_1 \geq t_\kappa + \tau$, or if any safe control $\mathbf{u} \in \mathcal{U}_\tau^{safe}(t_\kappa)$:*

- *keeps the system of the vehicles of $\mathcal{N}_{t_1}$ safe at time $t_\kappa + \tau$ and*
- *remains safe over $[t_\kappa, +\infty[$ for the vehicles of $\mathcal{N}_{t_\kappa}$,*

regardless of the control applied by the vehicles of $\mathcal{N}_{t_1} \setminus \mathcal{N}_{t_\kappa}$ over $[t_1, t_1 + \tau]$.

This definition ensures that a safe control computed for the vehicles of $\mathcal{N}_{t_\kappa}$ remains safe after new vehicles enter, *i.e.* the entry of new vehicles does not invalidate previously safe controls. Moreover, we assume that we can safely exclude vehicles departing the supervision area from the safety verification problem, *i.e.* that drivers are able to safely follow the previously departed vehicles without supervision. We will show in Section IV-C that these hypotheses allow discrete-time supervision with continuous vehicle arrival.

In the remainder of this article, we consider a centralized supervisor working in discrete time steps of duration τ, and we assume that new vehicles always enter safely with a margin τ. At the beginning of each time step κ, the supervisor receives an information about the desired longitudinal control of each vehicle for the next time step, denoted by $u_{i,des}^\kappa$. The collection of these desired controls for the vehicles of $\mathcal{N}_{t_\kappa}$ defines a constant desired system control $\mathbf{u}_{des}^\kappa$ defined over $[t_\kappa, t_\kappa + \tau[$.

This control may, or may not, lead the system of vehicles into an unsafe state. The supervisor is tasked with preventing the system from entering an unsafe state, by overriding the desired control if necessary. To remain compatible with human drivers, it is desirable that the supervisor has several properties, namely being *least restrictive* and *minimally deviating*. Letting $\mathcal{U}_{\tau,\kappa}^{safe}(t_\kappa)$ be the restriction of the functions of $\mathcal{U}_\tau^{safe}(t_\kappa)$ to $[t_\kappa, t_\kappa + \tau[$, we define the *least restrictive* supervision problem:

Definition 3 (Least restrictive supervision)**.** *Consider a safe state $\mathbf{x}^\kappa$ at time $t_\kappa = \kappa\tau$, a desired system control $\mathbf{u}_{des}^\kappa$ and assume that all new vehicles enter the supervision area safely with a margin τ. The least restrictive supervision problem (SP) is that of finding a control $\mathbf{u}_{safe}^\kappa \in \mathcal{U}_{\tau,\kappa}^{safe}(t_\kappa)$ such that $\mathbf{u}_{safe}^\kappa = \mathbf{u}_{des}^\kappa$ if $\mathbf{u}_{des}^\kappa \in \mathcal{U}_{\tau,\kappa}^{safe}(t_\kappa)$.*

Note that this definition corresponds to that of [7] in our generalized setting. Such a supervisor is *least restrictive* because overriding only occurs if the initially requested control would lead the vehicles in an unsafe state. However, it is also desirable that the control used for overriding is chosen close to the drivers' desired control. Extending the work in [8], we define the *minimally deviating* supervision problem as follows:

Definition 4 (Minimally deviating supervision)**.** *Consider a safe state $\mathbf{x}^\kappa$ at time $t_\kappa = \kappa\tau$, a desired system control $\mathbf{u}_{des}^\kappa$ and assume that all new vehicles enter the supervision area safely with a margin τ. The minimally deviating supervision problem (SP^*) is that of finding a constant control ${\mathbf{u}^*}_{safe}^\kappa$ such that:*

$$ {\mathbf{u}^*}_{safe}^\kappa = \underset{\mathbf{u} \in \mathcal{U}_{\tau,\kappa}^{safe}(t_\kappa)}{\arg\min} \; ||\mathbf{u}^\kappa - \mathbf{u}_{des}^\kappa|| \tag{3} $$

where $||\cdot||$ is a norm defined over $\mathbf{U}_{t_\kappa}$.

Note that, from this definition, any solution to SP^* is a solution to SP.

This concept of minimally deviating supervision follows a different fail-safety paradigm that could be found in, *e.g.*, rail transportation where all trains in an area should perform an emergency braking when an incident occurs. The reasoning behind definition 4 is that, to improve efficiency without sacrificing safety, intervention is only performed on vehicles which are actually at risk, and does not necessarily result in a full stop. However, at individual vehicle level, the safe overriding control ${u^*}_{safe}^\kappa$ may differ greatly from the driver's input, *e.g.* braking instead of accelerating.

IV. Infinite Horizon Formulation of the Supervision Problem

In this section, we present an extension of the work in [36] allowing to reformulate the generalized minimally deviating supervision problem using mixed-integer quadratic programming (MIQP) in Section IV-A. As the supervisor works in discrete time steps of duration τ, we consider the beginning of a step κ, corresponding to a time $t_\kappa = \kappa\tau$ and formulate an infinite-horizon MIQP problem. Assuming the initial state

is safe, we will show in Section IV-B that this formulation can be used to find a minimally deviating safe control for the vehicles in $\mathcal{N}_{t_\kappa}$. We will show in Section IV-C that, if the vehicles of $\mathcal{N}_{t_\kappa}$ follow the corresponding control, our formulation expressed at $t_{\kappa+1} = (\kappa+1)\tau$ remains feasible for the vehicles of $\mathcal{N}_{t_{\kappa+1}}$, provided that all new vehicles enter safely with a margin τ. These properties ensure that our infinite horizon MIQP formulation can be solved in a receding horizon fashion, to ensure safety for all future vehicles.

A. Model variables and constraints

In what follows, we present the variables and constraints used in our model. Unless specified otherwise, these constraints are enforced at all time steps $k \geq \kappa$, and for all vehicles of $\mathcal{N}_{t_\kappa}$.

1) Vehicle dynamics: When they evolve inside the supervision area, vehicles use a piecewise-constant control, which is updated every τ seconds. For a vehicle $i \in \mathcal{N}_{t_\kappa}$ at step k, we introduce the variables s_i^k, $v_i^k \in [0, \overline{v}_i]$ and $u_i^k \in [\underline{u}_i, \overline{u}_i]$, respectively denoting its curvilinear position and longitudinal speed at t_k, and longitudinal acceleration over $[t_k, t_k+\tau[$. The following constraints enforce vehicle dynamics:

$$s_i^{k+1} - s_i^k = \frac{1}{2}\left(v_i^k + v_i^{k+1}\right)\tau \tag{4}$$

$$v_i^{k+1} - v_i^k = u_i^k \tau \tag{5}$$

2) Logical constraints: In [38], we showed that it is possible to enforce logical constraints on continuous and integer variables with linear inequalities using a "big-M" formulation. More specifically, if b is a binary variable and x a continuous or integer variable bounded so that $|x| < M$, then the logical constraint: $(b = 0 \Rightarrow x \leq a)$ is equivalent to the linear inequality constraint $x \leq a + bM$. This method can be used to define indicator binary variables for a given semi-infinite interval: for a continuous variable x and a constant $a \in \mathbb{R}$, we denote by $b = \chi_{[a,+\infty[}(x)$ the constraints $(b = 0 \Rightarrow x \leq a) \wedge (b = 1 \Rightarrow x \geq a)$, where $\wedge$ denotes the binary conjunction; we use $\neg$ to denote the binary negation.

3) Collision avoidance: As presented in Section III-A3, the *collision region* between two vehicles $i, j \in \mathcal{N}_{t_\kappa}$, $\mathcal{C}_{ij}$, can be computed off-line. As it was already presented in [38], it is possible to compute a minimal bounding convex polygon for each connected component $\mathcal{C}_{ij}^p$ of $\mathcal{C}_{ij}$. A good compromise between accuracy and complexity is to use a bounding hexagon with edges either parallel to the $s_i = 0$, $s_i = s_j$ or $s_j = 0$ lines; such a polygon is uniquely defined by six parameters, as shown in fig. 4.

To ensure that vehicles do not enter any of the collision regions, we introduce a set of binary variables to encode the discrete decisions arising from the choice of an ordering of vehicles, as presented in [38]. For all conflicting vehicles $i, j \in \mathcal{N}_{t_\kappa}$, we let $\pi_{ij}^p = 1$ if vehicle i passes the p-th collision region before j, and $\pi_{ij}^p = 0$ otherwise; moreover, we introduce the binary indicator variables for all $k \geq \kappa$:

$$\varepsilon_{ij,p}^{\parallel}(k) = \chi_{[s_{ij,p}^{\parallel},+\infty[}(s_i^k), \tag{6}$$

$$\varepsilon_{ij,p}^{\perp}(k) = \chi_{[\overline{s}_{ij,p}^{\perp},+\infty[}(s_i^k). \tag{7}$$

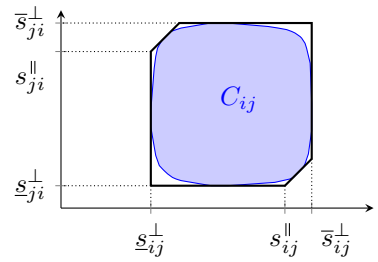


Fig. 4. Minimum bounding hexagon for the collision region presented in fig. 2a.

We enforce the collision avoidance constraints for all conflicting vehicles $i, j \in \mathcal{N}_{t_\kappa}$ and $k \geq \kappa$ as:

$$\left(\pi_{ij}^p \wedge \neg\, \varepsilon_{ij,p}^{\parallel}(k)\right) \Rightarrow s_j^{k+1} \leq \underline{s}_{ji,p}^{\perp} \tag{8}$$

$$\left(\pi_{ij}^p \wedge \varepsilon_{ij,p}^{\parallel}(k) \wedge \neg\, \varepsilon_{ij,p}^{\perp}(k)\right) \Rightarrow s_i^{k+1} \geq s_j^{k+1} + d_{ij,p} \tag{9}$$

$$\left(\pi_{ij}^p \wedge \varepsilon_{ij,p}^{\parallel}(k) \wedge \neg\, \varepsilon_{ij,p}^{\perp}(k)\right) \Rightarrow$$
$$s_i^{k+1} \geq s_j^{k+1} + d_{ij,p} + \frac{\tau}{2}\left(v_j^{k+1} - v_i^{k+1}\right) \tag{10}$$

where $d_{ij,p} = s_{ij,p}^{\parallel} - \underline{s}_{ji,p}^{\perp}$. Constraint (8) corresponds to "crossing situations", where a vehicle has to wait for another to pass; constraints (9) and (10) correspond to "following situations", where a vehicle needs to maintain a certain longitudinal distance from another.

Note that constraints (8) to (10) use the values of the indicator variables at step k to force the positions of the vehicles at step $k+1$ in order to avoid a "corner cutting" phenomenon; the additional constraint (10) prevents collisions between two time steps. These constraints are very slightly stronger than that of collision avoidance, *i.e.* for all $t \geq t_\kappa$, $(s_i(t), s_j(t)) \notin \mathcal{C}_{ij}$. Consequently, the results in the rest of this article are to be understood replacing the exact collision avoidance constraints in definition 1 by conditions (8)-(10).

Finally, to ensure the consistency of the formulation, we add the mutual exclusion constraint for all conflicting vehicles $i, j \in \mathcal{N}_{t_\kappa}$:

$$\pi_{ij}^p + \pi_{ji}^p = 1. \tag{11}$$

4) Deadlock avoidance: As described in Section III-A6, we require all vehicles to maintain a minimum speed inside their no-stop region $\mathcal{D}_i = [\underline{s}_i^{\perp}, \overline{s}_i^{\perp}]$. This requirement is enforced by defining additional binary variables, for all $i \in \mathcal{N}_{t_\kappa}$ and all $k \geq \kappa$:

$$\zeta_i^{acc}(k) = \chi_{[\underline{s}_i^{acc},+\infty[}(s_i^k) \tag{12}$$

$$\zeta_i^{in}(k) = \chi_{[\underline{s}_i^{\perp},+\infty[}(s_i^k) \tag{13}$$

$$\zeta_i^{out}(k) = \chi_{[\overline{s}_i^{\perp},+\infty[}(s_i^k) \tag{14}$$

$$\eta_i(k) = \chi_{[v_{min}-u_a\tau,+\infty[}(v_i^k) \tag{15}$$

and using the constraints:

$$\left(\zeta_i^{acc}(k) \wedge \neg\, \zeta_i^{in}(k) \wedge \neg\, \eta_i(k)\right) \Rightarrow v_i^{k+1} \geq v_i^k + u_a\tau, \tag{16}$$

$$\left(\zeta_i^{in}(k) \wedge \neg\, \zeta_i^{out}(k)\right) \Rightarrow v_i^k \geq v_{min}. \tag{17}$$

As long as the acceleration regions $\mathcal{A}_i$ are large enough, constraint (16) prevents vehicles from remaining blocked due

to the minimum speed requirement (17). We will show in the next section that these conditions effectively prevent deadlocks for all future times.

5) Initial conditions: The supervision problem is used in a receding horizon fashion, and we consider that the state of each vehicle of $\mathcal{N}_{t_\kappa}$ at time t_κ is known before solving the problem. Therefore, we use the following initial condition for all $i \in \mathcal{N}_{t_\kappa}$:

$$(s_i^\kappa, v_i^\kappa) = (s_i(t_\kappa), v_i(t_\kappa)) \tag{18}$$

B. Objective function

Any piecewise-constant control verifying constraints (4) to (18) for all $k \geq \kappa$ is dynamically admissible and prevents collisions for all future times, and is therefore in $\mathcal{U}_{\tau,\kappa}^{safe}(t_\kappa)$. To remain compatible with human driving, we now formulate an objective function allowing to find a least restrictive and minimally deviating control given a desired control $\mathbf{u}_{des}^\kappa = (u_{i,des}^\kappa)_{i\in\mathcal{N}_{t_\kappa}}$. In what follows, we let $(w_i^\kappa)_{i\in\mathcal{N}_{t_\kappa}}$ be a set of strictly positive weights, $\mathbf{X}$ be the tuple of all the problem variables, and we define:

$$J^\kappa(\mathbf{X}) = \sum_{i\in\mathcal{N}_{t_\kappa}} w_i^\kappa \left(u_i^\kappa - u_{i,des}^\kappa\right)^2. \tag{19}$$

Noting $\pi_{\mathbf{u}^\kappa}$ the projection operator such that $\pi_{\mathbf{u}^\kappa}(\mathbf{X}) = \mathbf{u}^\kappa$, we deduce the following theorem:

Theorem 1. *The solution of the optimization problem:*

$$\mathbf{u}^* = \pi_{\mathbf{u}^\kappa} \arg\min_{\mathbf{X}} J^\kappa(\mathbf{X}) \tag{IH-SP}$$
$$\textit{subj. to} \quad \forall k \geq \kappa,\ (4)-(18)$$

is a solution to the minimally deviating supervision problem SP^* at time t_κ, for the norm associated with $J^\kappa \circ \pi_{\mathbf{u}^\kappa}$.

Note that the weighting terms w_i^κ allow distinguishing between different types of agents, for instance to prioritize emergency services or high-occupancy vehicles. More complex cost functions can also be used, for instance to penalize a forced acceleration more than a forced braking.

C. Receding horizon properties

We now assume that there exists a solution to IH-SP at time t_κ, that the vehicles of $\mathcal{N}_{t_\kappa}$ follow this solution control over $[t_\kappa, t_\kappa + \tau]$, and that the vehicles of $\mathcal{N}_{t_{\kappa+1}}$ enter safely with a margin τ. From Definitions 1 and 2, we have the following theorem:

Theorem 2 (Recursive feasibility)**.** *Let $\tau > 0$, $\kappa \geq 0$, $t_\kappa = \kappa\tau$ and $t_{\kappa+1} = t_\kappa + \tau$. Assume that:*

- *there exists a solution to IH-SP at time t_κ for the vehicles of $\mathcal{N}_{t_\kappa}$,*
- *the vehicles of $\mathcal{N}_{t_\kappa}$ follow this solution control over $[t_\kappa, t_{\kappa+1}]$,*
- *the vehicles of $\mathcal{N}_{t_{\kappa+1}} \setminus \mathcal{N}_{t_\kappa}$ enter safely with a margin τ.*

Then there exists a solution to IH-SP at time $t_{\kappa+1}$ for the vehicles of $\mathcal{N}_{t_{\kappa+1}}$.

Proof. From definitions 1 and 2, and using theorem 1, we know that the first two hypotheses guarantee that the vehicles in $\mathcal{N}_{t_\kappa}$ are in a safe state at time $t_{\kappa+1}$. Moreover, the third hypothesis ensures that the vehicles in $\mathcal{N}_{t_{\kappa+1}}$ also are in a safe state at $t_{\kappa+1}$ regardless of the control applied by the vehicles of $\mathcal{N}_{t_{\kappa+1}} \setminus \mathcal{N}_{t_\kappa}$ up to time $t_{\kappa+1}$. By definition 1, there exists a feasible solution to IH-SP thus proving the theorem. □

We now state that the IH-SP formulation effectively prevents the apparition of deadlocksThe proof of this theorem can be found in Appendix A-A.

Theorem 3 (Deadlock avoidance)**.** *Let $\kappa \geq 0$ and assume that, for all $\kappa \leq k \leq \kappa_0$, the conditions of theorem 2 remain satisfied at time t_k. There exists a feasible solution of IH-SP at time t_{κ_0} in which all the vehicles in $\mathcal{N}_{t_{\kappa_0}}$ exit the supervision area in finite time.*

Note that theorem 3 only ensures that, at all times, there exists a solution where all the vehicles inside the supervision at this particular time eventually exit. However, there is no guarantee that such a solution will actually be selected, for instance if one driver wishes to stop although there is no other vehicle. There is also no fairness guarantee, *i.e.* it is possible that one vehicle is forced to remain stopped for an arbitrarily long time, for instance if there is a very heavy traffic coming from another direction. Future developments will focus devising more complex objectives function to take traffic efficiency and fairness into account.

D. Multiple paths choices

The above formulation assumes that the path of each vehicle is known in advance. However, this may not be realistic in the context of semi-autonomous cars where drivers can decide to change paths, for instance to avoid an obstacle on the road or use another itinerary. Using additional variables to indicate the path to which a vehicle is assigned, our formulation can be extended to handle multiple possible paths for each vehicle. Due to length limitations, this extension will be detailed in future work.

V. Finite Horizon Formulation

In Section IV-C, we presented an infinite horizon formulation to solve the minimally deviating supervision problem. However, due to the infinite number of variables, this formulation is not suitable for practical resolution. In this section, we derive an equivalent finite horizon formulation that can be implemented and solved using standard numerical techniques.

In what follows, we let $K \geq 1$ and we denote by FH-SP$_K$ the restriction of IH-SP at time t_κ to the variables at steps k with $\kappa \leq k \leq \kappa + K$, and we only consider the constraints (4) to (18) up to step $\kappa + K$. The objective function is unchanged. A solution to FH-SP$_K$ at time t_κ allows to compute a control preventing collisions up to time $t_\kappa + K\tau$; however, due to the dynamics of the vehicles, the state reached at $t_\kappa + K\tau$ may not be safe. Since FH-SP$_K$ only has a subset of the constraints of IH-SP, we can formulate the following proposition:

Proposition 1. *Let $K \geq 1$ and let $\mathbf{X}$ be a solution of IH-SP at step κ. The restriction of $\mathbf{X}$ to the first $K+1$ time steps is a feasible solution to FH-SP$_K$.*

Using the global bounds u_a, u_b, u_{max} and v_{max} defined in section III-A4, we will now prove a reciprocal implication to proposition 1: if K is chosen large enough, any solution of FH-SP$_K$ can be used to construct a solution of IH-SP.

As presented in [36], the key idea of the proof lies in the choice of a planning horizon long enough to allow any vehicle to fully stop. The structure of the demonstration is as follows: lemma 1 gives a lower bound on the time horizon to allow a single isolated vehicle to stop using discrete dynamics, although with a potential risk of rear-end collisions from following vehicles. In proposition 2, we give a slightly higher bound on the time horizon ensuring that all vehicles in a line can all safely stop without rear-end collisions. Finally, in proposition 3 we give a bound on K ensuring the recursive feasibility of FH-SP$_K$; this allows formulating theorem 4, stating the equivalence of FH-SP$_K$ and IH-SP. In this section, we only present sketches of proofs for each result; detailed demonstrations can be found in Appendix A-B.

Lemma 1. *At a time t_κ, consider a horizon $T = K\tau$ with $T \geq \frac{v_{max}}{|u_b|} + \tau$. Let $i \in \mathcal{N}_{t_\kappa}$ be a vehicle for which there exists a piecewise-constant control $(u_i^k)_{\kappa \leq k < \kappa+K}$ such that, for all $\kappa \leq k < \kappa+K$, $u_i^k \in [\underline{u}_i, \overline{u}_i]$, corresponding to a dynamically feasible trajectory $s_i(t)$ over $[t_\kappa, t_\kappa + T + \tau]$.*
There exists a discrete control $(\tilde{u}_i^k)_{\kappa \leq k \leq \kappa+K}$ such that for all $\kappa \leq k \leq \kappa+K$, $u_i^k \in [\underline{u}_i, \overline{u}_i]$ and $\tilde{u}_i^\kappa = u_i^\kappa$, and for which the corresponding dynamically feasible trajectory $t \mapsto \tilde{x}_i(t) = (\tilde{s}_i(t), \tilde{v}_i(t))$ verifies $\tilde{s}_i(t_\kappa + T + \tau) \leq s_i(t_1 + T)$ and $\tilde{v}_i = 0$ over $[t_\kappa + T, t_\kappa + T + \tau]$.

Sketch of proof. $\frac{v_{max}}{|u_b|}$ is an upper bound on the required time for any vehicle to stop by applying a control u_b, which by definition is dynamically feasible. The additional τ accounts for the fact that we require $\tilde{u}_i^\kappa = u_i^\kappa$ at the first time step. □

In the following proposition and noting $\lceil \cdot \rceil$ the ceiling function, we prove a bound ensuring that a line of vehicles can safely stop before the leader reaches its final computed position at the end of the time horizon, without risk of rear-end collisions:

Proposition 2. *At a time t_κ, suppose that p vehicles of $\mathcal{N}_{t_\kappa}$ (denoted by $1, \ldots, p$ from rear to front) are following one another. Consider a horizon $T_{stop} = K_{stop}\tau \geq \frac{v_{max}}{|u_b|} + (p-1)\left(1 + \left\lceil \frac{u_{max}}{|u_b|} \right\rceil\right)\tau + \tau$, and assume that every vehicle $i \in \{1, \ldots, p\}$ has a safe discrete control $(u_i^k)_{\kappa \leq k < \kappa+K}$ such that, for all $\kappa \leq k < \kappa+K$, $u_i^k \in [\underline{u}_i, \overline{u}_i]$. We let $t \mapsto x_i(t)$ be the trajectory over $[t_\kappa, t_\kappa + T]$ for vehicle i under control (u_i^k).*
For all $i \in \{1, \ldots, p\}$, there exists a safe discrete control $(\hat{u}_i^k)_{\kappa \leq k \leq \kappa+K}$ such that for all $\kappa \leq k \leq \kappa+K$, $u_i^k \in [\underline{u}_i, \overline{u}_i]$, $\hat{u}_i^\kappa = u_i^\kappa$ and for which the corresponding dynamically feasible and safe trajectory $t \mapsto \hat{x}_i(t) = (\hat{s}_i(t), \hat{v}_i(t))$ verifies $\hat{s}_i(t_\kappa + T + \tau) \leq s_i(t_1 + T)$ and $\hat{v}_i = 0$ over $[t_\kappa + T, t_\kappa + T + \tau]$.

Sketch of proof. The worst case that needs to be taken into account corresponds to a situation where the initial states of the vehicles require each of them to accelerate in order to avoid a rear-end collision from the vehicle behind. This rather extreme situation happens when a vehicle goes faster than the one it is following, and the two are too close to allow a safe deceleration. In this case, the rearmost vehicle can always brake with the control from lemma 1, until it decelerates below the speed of the vehicle in front of it. The second rearmost vehicle can then decelerate, then the third and up to the front-most vehicle. The term $\left(1 + \left\lceil \frac{u_{max}}{|u_b|} \right\rceil\right)\tau$ arises from the piecewise-constant control hypothesis, and vanishes as τ goes to 0. Note that the condition $K_{stop}\tau \geq \frac{v_{max}}{|u_b|} + \frac{v_{max}}{u_a} + \tau$ also provides the same guarantees; depending on the value of p, this second bound might be more efficient. □

Remark 1. *The bound from proposition 2 depends on the number of vehicles in a line, and can become quite high when p is large. It can be proven that the condition $K_{stop}\tau \geq \frac{v_{max}}{|u_b|} + \frac{v_{max}}{u_a} + \tau$ also provides the same guarantees; depending on the value of p, this second bound might be more efficient.*

We can now prove the recursive feasibility of FH-SP$_K$ for a large enough K, as follows:

Proposition 3. *Consider a time t_κ, and assume that at most p vehicles are following one another at all times $t \geq t_\kappa$. We set $d = \max_{t \geq t_\kappa, i \in \mathcal{N}_t} \left(\overline{s}_i^\perp - s_i^{acc}\right)$ and we let T_{stop} be the stopping horizon from proposition 2 for p vehicles; moreover, we define $T_{rec} = K_{rec}\tau \geq T_{stop} + \frac{v_{min}}{u_a} + \frac{d}{v_{min}} + \tau$. We assume that all vehicles of $\mathcal{N}_t$ for all $t > t_\kappa$ enter safely with a margin τ.*

Problem FH-SP$_{K_{rec}}$ is recursively feasible under the hypotheses of theorem 2, i.e. if there exists a solution to FH-SP$_{K_{rec}}$ at time t_κ for the vehicles of $\mathcal{N}_{t_\kappa}$, there exists a solution at $t_\kappa + \tau$ for the vehicles of $\mathcal{N}_{t_\kappa+\tau}$.

Sketch of proof. The idea between the choice of T_{rec} is to ensure that each vehicle can either stop safely before entering its acceleration region (without generating rear-end collisions), or has already planned to exit its no-stop region safely. Moreover, the safe entering hypothesis ensures that the entry of new vehicles does not invalidate previously safe solutions, which can therefore be extended. □

We obtain the equivalence between IH-SP and FH-SP$_K$:

Theorem 4. *Problems IH-SP and FH-SP$_K$ with $K\tau \geq T_{rec}$ are equivalent, i.e. an optimal solution to one is also an optimal solution to the other.*

Proof. Proposition 1 ensures that any optimal solution to IH-SP is a feasible solution of FH-SP$_K$. Proposition 3 shows that a solution to FH-SP$_K$ (with $K\tau \geq T_{rec}$) can be recursively extended to a solution of IH-SP; therefore, the optimal solution of FH-SP$_K$ is feasible for IH-SP. Using these two results, we deduce the stated theorem. □

An important corollary of theorems 1, 3 and 4 is that the control obtained by solving FH-SP$_K$ with K large enough is also a solution to the minimally deviating supervision problem, and ensures deadlock avoidance as well. Contrary to IH-SP, FH-SP$_K$ is relatively easy to solve with dedicated mixed-integer quadratic programming solvers, as will be demonstrated in the following section.

VI. Simulation Results

A. Simulation environment

The presented Supervisor framework has been validated using extensive computer simulations on various test scenarios. In the absence of standardized test situations and since no open-sourced implementation of comparable methods [7], [8] is available, this section does not aim at a quantitative comparison with existing algorithms. Since our Supervisor is by design guaranteed to output an optimal[1] safe control, the major evaluation criterion is rather its ability to handle a wider variety of traffic scenarios than existing techniques, which is demonstrated in the rest of this section.

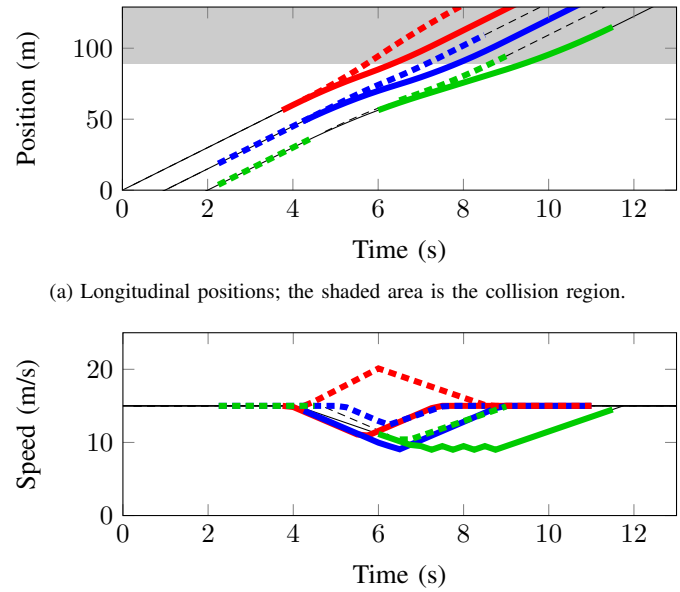
Due to implementation reasons, the resolution of the supervision problem is performed off-line and simulations are run in two successive phases. In the first phase, we define the geometry of the roads inside the supervision area and the corresponding possible paths, and compute the collision and acceleration regions information for each pair of paths. Since these sets only depend on the geometry of vehicles and paths, the corresponding parameters are computed off-line.

In the second phase, we run the simulation by coupling the high-fidelity vehicle physics simulator PreScan [39] with an external Python implementation of our supervisor. The actual resolution uses the commercial MIQP solver GUROBI [35]; the Python program runs a coarse simulation over a set time horizon with a fixed time step duration. Vehicles are generated using random Poisson arrivals, with a predefined arrival rate for each possible path, while respecting the safe entering condition; the initial velocity of each generated vehicle is chosen randomly according to a truncated Gaussian distribution. At each time step, the finite horizon supervision problem FH-SP is solved for the vehicles inside the supervision area, and yields the best safe control for the set of vehicles. The state of these vehicles at the next time step is then computed according to equations (4) and (5).

In parallel, we use PreScan to validate the consistency of this output: from the safe controls computed in the Python supervisor and knowing the reference paths of the vehicles, we compute a target state comprising a desired position, heading and longitudinal velocity for each vehicle. This target state is fed into a low-level controller which outputs a steering and an acceleration or braking control. The vehicle model used in the validation phase takes into account engine response as well as chassis and suspensions dynamics, but does not model road-tire friction. PreScan's collision detection and visualization capacities are then used to validate the absence of collision or dangerous situations. Note that vehicles controllers are designed to ensure a bounded positioning error for any vehicle, relative to their prescribed path and velocity profile. This error is taken into account in the computation of the collision regions, so that the system is robust to control imperfections.

B. Test scenarios

In the rest of this section, we consider three test scenarios – chosen to represent a wide variety of driving situations – consisting of merging on a highway, crossing an intersection or driving inside a roundabout. To showcase the performance of our framework in avoiding accidents and deadlocks, we assume that drivers are "oblivious" and focused on tracking a desired speed, regardless of the presence of other vehicles. A video of the presented simulations is available online[2].



(a) Longitudinal positions; the shaded area is the collision region.

(b) Longitudinal velocities

Fig. 5. Vehicles positions and velocities in the merging scenario; solid lines correspond to vehicles on the entry lane, dashed lines to vehicles starting on the highway. The thick colored portions show overriding intervals.

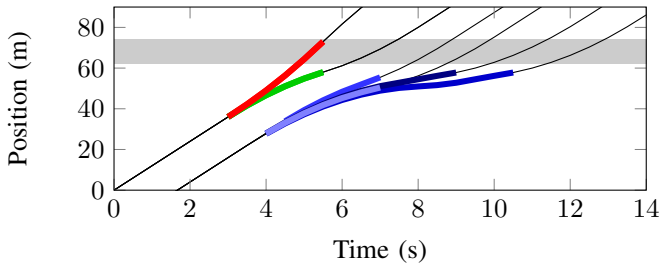
1) Highway merging: We first consider a very simple highway merging scenario, where an entry lane merges into a single-lane road; the possible paths for the vehicles are the same as in fig. 2c. The collision region between a vehicle i in the entry lane and a vehicle j on the highway have a single connected component given as $\underline{s}_{ij}^{\perp} = \underline{s}_{ji}^{\perp} = 89\,\mathrm{m}$ and $s_{ij}^{\parallel} = s_{ji}^{\parallel} = 94\,\mathrm{m}$, taking control errors into account.

To illustrate the action of the supervisor, we consider a set of six vehicles, three of which are on the highway and three on the entry lane. All vehicles are assumed to have "oblivious" drivers maintaining a constant speed, thus resulting in potential collisions. This admittedly unrealistic behavior has been chosen to generate a higher probability of collisions in absence of supervision. Figure 5 shows the longitudinal trajectories of the supervised vehicles; colored (thick) portions of the lines represent intervals of time during which overriding occurs. The area in gray corresponds to the collision region between entering vehicles and vehicles on the highway; thanks to the action of the supervisor, all collisions are successfully avoided.

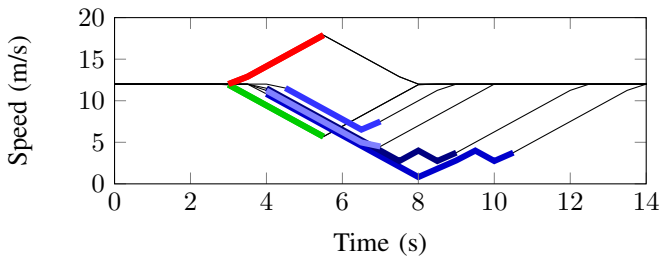
2) Intersection crossing: The second scenario is the crossing of a + shaped intersection by a total of eight vehicles, with two vehicles per branch. In each branch, the front vehicle goes straight, and the rear vehicle turns left; moreover, all vehicles in front start at the same distance from the center of the intersection, and the same is true for the vehicles in the

[1] Among the set of piecewise-constant controls with a given time step duration and in the sense of Definition 4

[2] Available at https://youtu.be/JJZKfHMUeCI



(a) Longitudinal positions; the shaded area is the collision region.



(b) Longitudinal velocities

Fig. 6. Vehicles positions and velocities in the intersection crossing scenario; solid lines correspond to vehicles on the entry lane, dashed lines to vehicles starting on the highway. The thick colored portions show overriding intervals.

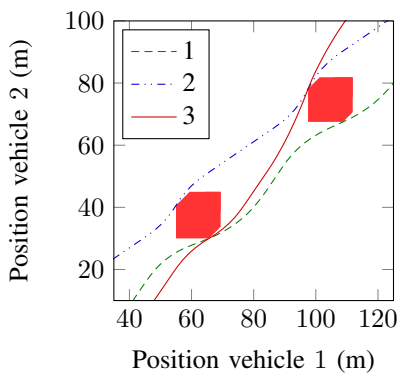


Fig. 7. Illustration of three possible classes of trajectories found by the solver, depending on the initial states of the vehicles. Trajectories 1 and 2 correspond to one vehicle passing the two collision points before the other. Trajectory 3 corresponds to the case where the vehicle on the inner lane enters after the other, and overtakes it inside the roundabout.

rear. This scenario illustrates the symmetry-breaking capacities of our framework, which handles this perfectly symmetrical scenario well, as shown in fig. 6. The area in gray corresponds to the collision region between vehicles on different branches. A video of a longer, one hour simulation is available also online[3].

3) Roundabout driving: Finally, the third scenario consists of vehicles driving inside a two-lanes roundabout. The particularity of this situation is that collision regions can have multiple connected components, for instance for the paths shown in fig. 2b. Since our formulation explicitly distinguishes each of these connected components, the supervisor is able to choose an ordering for each point of conflict, as illustrated in fig. 7: depending on the initial states and control targets of the vehicles, a different class of solution is chosen. A video of a longer, one hour simulation is also available online[4].

4) Computation time: Due to the relatively short time horizon needed to ascertain infinite horizon safety, computation time remains reasonable despite the NP-hardness of the MIQP formulation. Figure 8 shows the evolution of the computation time in the intersection crossing and roundabout scenarios; the limited available space in the merging scenario does not allow enough vehicles for a similar diagram. These measurements have been obtained on a computer equipped with an Intel Core i7-6700K CPU clocked at $4\,\mathrm{GHz}$ with $16\,\mathrm{GB}$ of RAM, using the GUROBI solver in version 7.0. It can be seen that computation time remains below the duration of a time step in 90% of cases for up to approximately ten simultaneous vehicles, thus allowing real-time computation at $2\,\mathrm{Hz}$.

Note that the MIQP problem only loosely depends on the paths geometry, but rather on the average number of conflicts per vehicle which is higher in the case of roundabout driving, thus explaining the longer times reported in fig. 8b. Moreover, the implemented algorithm has been devised for readability over efficiency, and can be optimized by removing redundant variables to further reduce computation time. In practice, this refresh rate means that vehicles could apply a new acceleration every $0.5\,\mathrm{s}$, which is faster than the typical reaction time of one second for a human driver, and should therefore be barely perceived. Note that for practical implementation purposes, the input of the supervisor should be predicted states at the end of the computation period instead of current states; since the acceleration of each vehicle is assumed to be known to the supervisor, these predictions can be easily performed by forward integration.

VII. Discussion on Implementation

In the previous sections, we presented an optimization-based algorithm for the supervision of semi-autonomous vehicles; we now briefly discuss obstacles and possible solutions for actual implementation. First and foremost, not all vehicles will be equipped with the required communication capacities at the same time; therefore, the ability to deal with unequipped vehicles and other traffic participants is key to envision actual applications. Second, this work assumes perfect communication and control, and in general ignores uncertainties arising from real-world constraints.

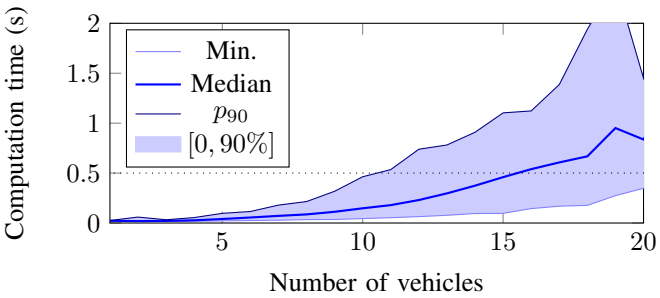
A. Dealing with unequipped vehicles

As with all innovations, the penetration rate of our system would gradually increase overtime, but remain below $100\,\%$ for years, yet the formulation proposed in Section IV requires all vehicles to be equipped with supervision capacities. Although a detailed study on the integration of unequipped vehicles in our framework is out of the scope of this paper, we present a possible technique to handle these vehicles provided that they can avoid longitudinal collisions with the leading vehicle, and have a bounded reaction time.

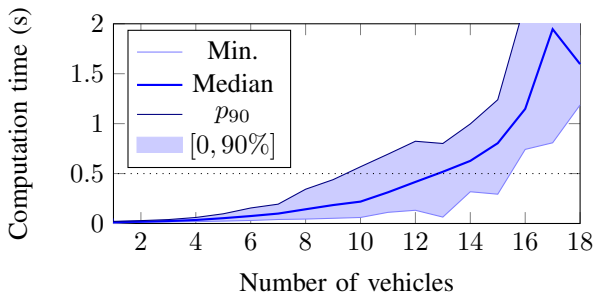
First, note that it is always possible to consider unequipped vehicles conservatively as proposed in [40]: at a given step k,

[3]https://youtu.be/cl32nbceZvw

[4]https://youtu.be/pLoG32wFnkE



(a) Intersection scenario



(b) Roundabout scenario

Fig. 8. Distribution of computation times depending on the number of vehicles, for $\tau = 0.5\,\mathrm{s}$. Shaded areas represent the $[0, 90\%]$ percentiles.

we compute the minimum and maximum curvilinear position that can be reached at time t_k by the unequipped vehicle i_u, denoted by $s^k_{i_u,min}$ and $s^k_{i_u,max}$ respectively. Using the same notations as in Section IV, we then define:

$$\varepsilon^{\parallel}_{i_u j,p}(k) = \chi_{[s^{\parallel}_{i_u j,p},+\infty[}(s^k_{i_u,max}), \tag{20}$$

$$\varepsilon^{\perp}_{i_u j,p}(k) = \chi_{[\overline{s}^{\perp}_{i_u j,p},+\infty[}(s^k_{i_u,min}). \tag{21}$$

Therefore, the unequipped vehicle is considered as occupying the conflict region at step k when there exists a control (maximum acceleration) for which it could be inside this region at step k. Similarly, the vehicle is only considered as liberating the conflict region when, even by applying a maximum braking, it would exit it. The collision avoidance constraints (9) and (10) are also modified to use $s^{k+1}_{i_u,min}$ and $v^{k+1}_{i_u,min}$, where $v^{k+1}_{i_u,min}$ is the minimum speed reachable by i_u at $k+1$. Other traffic participants such as cyclists (and, to a lesser extent, pedestrians) could also be taken into account in this fashion. Recently proposed "non-conservatively defensive strategies" [41] could also be applied.

A limitation of this simple approach is that it can lead equipped vehicles to often yield right-of-way to unequipped vehicles, which may be problematic and can slow the acceptance of the system. A possible method (introduced in [42]) to reduce this problem while improving the global level of safety is to use the existing equipped vehicles to force the unequipped ones to stop when required. Suppose that an unequipped vehicle (denoted by i_u) follows an equipped one (i_e), both crossing the path of another equipped vehicle j_e. By setting $\pi_{i_e j_e} = 0$ (thus requiring j_e to pass before i_e), we effectively force the unequipped vehicle i_u to also pass after j_e; the reaction time of the unequipped vehicle can be taken into account by adjusting the lower bound on the longitudinal acceleration of vehicle i_e.

Note that this approach still guarantees that no collision can happen between an unequipped and an equipped vehicle; moreover, as the penetration rate of equipped vehicles increases, additional rules may be enforced to reduce the number of occurrences in which conflicting unequipped vehicles are simultaneously allowed in the conflict region, thus increasing safety even for the unequipped vehicles. Future work will study the impact of penetration rate on safety and efficiency for both equipped and unequipped vehicles.

B. Practical implementation

We propose a centralized implementation, where a roadside computer (*supervisor*) with communication capacities is added to the infrastructure, and is tasked with repeatedly solving FH-SP$_K$. Note that resolution could also be performed using cloud computing, possibly providing much faster computations without necessitating fully dedicated hardware. The supervisor is also assumed to be equipped with a set of sensors (*e.g.*, cameras), so that the arrival of new vehicles in the supervision area can be monitored (in order to account for unequipped vehicles and other traffic participants). Equipped vehicles are supposed to regularly communicate their current state, including position, velocity and driver's control input, and receive instructions (the safe acceleration sequence (u^k_i) solution of FH-SP$_K$) from the roadside supervisor. The vehicle's on-board computer then uses these instructions to override the driver's control inputs when needed. We argue that the main sources of uncertainty, *i.e.* communication, sensing and control errors, can be taken into account by using safety margins when computing collision regions.

Communications are assumed to have similar performance to current 802.11p specifications; we use the figures provided in [43], [44] as reference, with latency below $20\,\mathrm{ms}$, and packet loss probability of less than $30\,\%$ under $300\,\mathrm{m}$. To account for network congestion, we use more conservative values than those reported experimentally in [44]. Moreover, using the additional roadside sensors, we estimate that uncertainty in each vehicle's localization could be reduced to below $1\,\mathrm{m}$ longitudinally.

First, the $20\,\mathrm{ms}$ latency corresponds to less than $1\,\mathrm{m}$ at highway speed. Second, since they do not require exchanging a lot of data, such messages can be sent much more frequently than the refresh rate of the supervisor. Considering messages can be sent at $20\,\mathrm{Hz}$, the probability of a message not being received in $0.25\,\mathrm{s}$ is roughly $0.2\,\%$, and 6×10^{-6} after $0.5\,\mathrm{s}$. Since they receive a whole sequence of safe accelerations, individual vehicles can keep executing this sequence until a new one is successfully received. A worst-case scenario would be having one vehicle using acceleration u_a (maximum acceleration) where it should have used u_b (maximum braking): after a duration t, the corresponding positioning error is $\frac{1}{2}t^2(u_a + |u_b|)$, which is roughly $30\,\mathrm{cm}$ after $0.25\,\mathrm{s}$ and $1.3\,\mathrm{m}$ after $0.5\,\mathrm{s}$ for typical values of u_a and $|u_b|$ of $5\,\mathrm{m\,s^{-2}}$. More robust contingency protocols could likely be developed, and will be the subject of future work, but these values can be used as safety margins without compromising performance.

Similarly, positioning and control uncertainty can be accounted for as margins in the collision regions, provided they can be bounded. In this work, we assume that vehicle self-positioning can be improved using the roadside sensors from the supervisor (which can be precisely calibrated), which could provide relatively tight bounds on error.

VIII. Conclusion

In this article, we designed a framework allowing safe semi-autonomous driving of multiple cooperative vehicles in various traffic situations. We first introduce a set of linear constraints ensuring infinite horizon safety for a group of human-driven vehicles, traveling inside predefined corridors with the help of existing lane-keeping technologies. Based on this set of constraints, a discrete-time *Supervisor* is allowed to override the drivers' longitudinal control inputs if they would lead the vehicles into an inevitable collision state. In this case, the control used for overriding is chosen as close as possible to the one originally requested by the drivers. These two properties ensure that intervention only occurs when strictly necessary to maintain safety, thus facilitating the acceptation of the system by human drivers.

Theoretical considerations prove this supervisor guarantees both safety and deadlock avoidance, and can be applied without distinction to multiple situations such as traffic intersection, highway entry lanes or roundabouts. Using the realistic vehicle physics simulator PreScan, we demonstrated that our algorithm can handle complex situations over an arbitrary duration, with continuous arrivals of vehicles. Moreover, the proposed formulation can be solved in real-time on a standard desktop computer for up to ten vehicles, which makes it suitable for practical applications.

Additionally, this work opens up many perspectives for future research. First and foremost, the current framework does not deal with non-equipped vehicles or other traffic participants such as cyclists or pedestrians, nor does it take sensor and communication uncertainties into account. Before considering an actual implementation, the system should be more robust to these various sources of noise. Moreover, our formulation has been designed in a mostly centralized fashion; various approaches need to be explored to design a more realistic decentralized system, that could be implemented in actual cars.

Appendix A
Demonstrations

A. Proofs for Section IV

Before proving theorem 3, we introduce the following lemma stemming from graph theory:

Lemma 2. *Let $\mathcal{G} = (V, E)$ a directed graph with vertices set V and edges set E. All cycles in $\mathcal{G}$ can be removed by reversing a set of edges, each of them contributing to at least one cycle.*

Proof. The proof is based on the existence of minimum feedback arc sets [45], *i.e.* a minimum set $E_{feedback} \subset E$ such that $\mathcal{G}' = (V, E \setminus E_{feedback})$ is acyclic. By minimality of $E_{feedback}$, any $e \in E_{feedback}$ belongs to at least one cycle of $\mathcal{G}$. Moreover, it can be seen that reversing the edges of $E_{feedback}$ also leads to an acyclic graph, thus proving the lemma. □

Proof of Theorem 3. Note that the only constraints requiring a vehicle to stop are (8) to (10), forcing a vehicle j to wait for a vehicle i with $\pi^p_{ij} = 1$. From the hypotheses and theorem 2, there exists a solution $\mathbf{X}$ to IH-SP at time t_{κ_0}. We define a directed *priority graph* $\mathcal{G}_{\mathbf{X}} = (V, E)$ with $V = \mathcal{N}_{t_{\kappa_0}}$ and where an edge $i \to j$ belongs to E if there exists p such that $\pi^p_{ij} = 1$. Using this representation, a cycle in $\mathcal{G}_{\mathbf{X}}$ corresponds to a chain of q conflicting vehicles $i_1, i_2, \dots, i_q, i_{q+1} = i_1$ for which there exists a connected component $\mathcal{C}^{p_n}_{i_n i_{n+1}}$ such that $\pi^{p_n}_{i_n i_{n+1}} = 1$ for all $n = 1 \dots q$.

If $\mathcal{G}_{\mathbf{X}}$ is acyclic, it defines a (partial) topological order, and it is always possible to admit the vehicles one by one in that order. Therefore, there exists a feasible solution where all the vehicles of $\mathcal{N}_{t_{\kappa_0}}$ exit the supervision area in finite time.

We now assume that there exists at least one cycle in $\mathcal{G}_{\mathbf{X}}$. If all the vehicles involved in the cycle can exit in finite time, the result of the theorem is proven. Otherwise, we note $\mathcal{N}_{dead} \subset \mathcal{N}_{t_{\kappa_0}}$ a set of vehicles corresponding to a cycle in $\mathcal{G}_{\mathbf{X}}$: all of these vehicles are stopped at infinity, and are prevented to move further by a constraint of form (8), for a certain $j \in \mathcal{N}_{dead}$. Moreover, the no-stop condition (17) ensures that, for all $i \in \mathcal{N}_{dead}$ and all $k \geq \kappa$, $s^k_i \leq \underline{s}^{\perp}_i$.

From lemma 2, we know that it is possible to change the values of the variables π^p_{ij} for $i, j \in \mathcal{N}_{dead}$ to render $\mathcal{G}_{\mathbf{X}}$ acyclic. Using the fact that $s^k_i \leq \underline{s}^{\perp}_i$ for all of these vehicles, we know that modifying these priorities does not violate constraints (8) to (10). Therefore, we can build a solution $\mathbf{X}'$ for which the corresponding priority graph is acyclic, which proves the theorem. □

B. Proofs for Section V

Proof of Lemma 1. The proof is trivial if we consider continuous-time dynamics, as a single vehicle can always apply a control lower or equal to u_b starting from time $t_k + \tau$, which ensures it is stopped for $t \geq t_k + \tau + \frac{v_{max}}{u_b}$. A slight additional complexity happens at the final braking time step when considering piecewise-constant controls, applying u_b for a duration τ might result in a negative velocity, which is not allowed in our framework. We now proceed to the formal proof, as below.

Let (u^k_i) be the control corresponding to trajectory s_i, and let us define a control (w^k_i) as: $w^\kappa_i = u^\kappa_i$, $w^k_i = \min(u_b, u^k_i)$ for $\kappa < k < \kappa + K$, and $w^{\kappa+K}_i = u_b$. We construct $(\tilde{u}^k_i)$ iteratively as $\tilde{u}^\kappa_i = u^k_i$ and, for $k \geq \kappa + 1$, $\tilde{u}^k_i = \begin{cases} w^k_i & \text{if } \tilde{v}^k_i + w^k_i \tau \geq 0 \\ -\frac{v^k_i}{\tau} & \text{otherwise} \end{cases}$, where $\tilde{v}^k_i$ is the speed of vehicle i at time t_k under control $(\tilde{u}^k_i)$.

As $\tilde{v}^{\kappa+1}_i = v^{\kappa+1}_i \leq v_{max} \leq (K-1)\tau|u_b|$ from the hypothesis, there exists a minimal value of $k_0 \geq \kappa$ such that $\tilde{v}^{k_0}_i \leq |u_b|\tau$; moreover, the condition on K ensures that $\tilde{v}^{\kappa+1}_i - (K-2)|u_b|\Delta_t \leq |u_b|\Delta_t$, and so $k_0 \leq (\kappa+1) + (K-2) = \kappa + K - 1$.

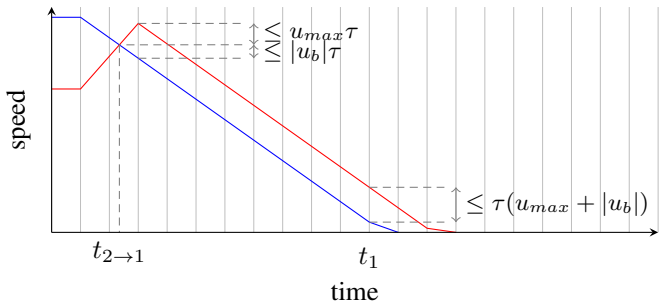


Fig. 9. Illustration of the "overshoot" phenomenon; the vertical grid correspond to integer multiples of the time step.

From the definition of $(\tilde{u}_i^k)$, we know that for all $k_0 + 1 \leq k \leq \kappa + K$, $\tilde{v}_i^k = 0$. Since $\tilde{u}_i^k \leq u_i^k$ for $\kappa \leq k \leq k_0 - 1$, we also know that $\tilde{s}_i^{k_0} \leq s_i^{k_0}$ and $\tilde{v}_i^{k_0} \leq v_i^{k_0}$. Finally, $\tilde{u}_i^{k_0}$ is the minimal admissible control starting from $\tilde{x}_i^{k_0}$; therefore, $\tilde{s}_i^{k_0+1} \leq s_i^{k_0+1} = s_i^{\kappa+K}$ which proves the above lemma. □

Proof of Proposition 2. We first consider the continuous-time case to give an intuition of the proof. We build upon the fact that the rearmost vehicle in a line can always brake with acceleration u_b until it fully stops. However, some the initial conditions may require vehicles in front to accelerate in order to avoid collisions, for instance if the rearmost vehicle is too fast. However, even in this case, we know that the second rearmost vehicle can brake with u_b as soon as it has matched the speed of the rearmost vehicle, and by induction this is true for all the vehicles in the line. In the continuous-time case, all of these vehicles can therefore stop in a time bounded by $\frac{v_{max}}{|u_b|}$.

When considering piecewise-continuous controls, an additional complexity arises from the fact that vehicles may match speed between two time steps, resulting in an "overshoot" in velocity. We use the hypotheses on the acceleration to bound this overshoot, as illustrated in fig. 9: we consider a fast vehicle (noted 1, blue curve) following a slower vehicle (noted 2, red curve). To avoid collisions, vehicle 2 is required to accelerate; vehicles match speed at time $t_{2\to1}$; however, due to the time discretization, the overshoot phenomenon can occur. Using the bounds on the acceleration, noting $k_{2\to1}$ the time step immediately following $t_{2\to1}$, we know that $v_2^k \leq v_1^k + \tau(u_{max} + |u_b|)$. Moreover, after step k, vehicle 2 can brake with a control u_b up to time t_1, corresponding to the first integer time step k_1 when $v_1^{k_1} \leq \tau|u_b|$. Since both vehicles 1 and 2 have the same acceleration of $[t_{2\to1}, t_1]$, we know that $v_2^{k_1} \leq \tau|u_b| + \tau(|u_b| + u_{max})$. Therefore, noting $k_2 = k_1 + 1 + \left\lceil \frac{u_{max}}{|u_b|} \right\rceil$, we know that $v_2^{k_2} \leq \tau|u_b|$. The same reasoning can then be repeated for the vehicles preceding vehicle 2. We will now formalize this recursion, as below.

We will prove by induction that, for $i \in \{1, \ldots, p\}$, there exists a dynamically feasible control $(\hat{u}_i^k)$ with $\hat{u}_i^\kappa = u_i^\kappa$ and $\hat{u}_i^k \leq u_i^k$ for $k \geq \kappa$ such that the corresponding vehicle speed $(\hat{v}_i^k)$ verifies $\hat{v}_i^{\kappa+K_i} \leq |u_b|\tau$ with $K_i\tau = \left\lceil \frac{v_{max}}{|u_b|} \right\rceil + (i-1)\left(1 + \left\lceil \frac{u_{max}}{|u_b|} \right\rceil\right)\tau$. First, for the rearmost vehicle $i = 1$, the proof of lemma 1 provides the result with $\hat{u}_i^k = \tilde{u}_i^k$.

We now let $i \geq 2$ and assume that every vehicle $j \in \{1, \ldots, i-1\}$ follows its corresponding control $(\hat{u}_j^k)$. We note $\hat{s}_j^k$ and $\hat{v}_j^k$ the position and speed of vehicle j at step k under this control. Since $\hat{u}_j^k \leq u_j^k$ for these vehicles, we deduce from the monotony of the system that the original control solution for vehicle i $(u_i^k)_{\kappa \leq k < \kappa+K}$ prevents rear-end collisions if vehicle $i-1$ applies $(\hat{u}_{i-1}^k)$. Therefore, any dynamically feasible extension of (u_i^k) is safe over $[\kappa\tau, (\kappa+K+1)\tau[$. As a result, the set $U_i^{safe}(u_i^\kappa, [\kappa, \kappa+K])$ of all admissible controls $(\hat{u}_i^k)_{\kappa \leq k \leq \kappa+K}$ for vehicle i such that $\hat{u}_i^\kappa = u_i^\kappa$ and $\underline{u}_i^k \leq \hat{u}_i^k \leq u_i^k$ for $\kappa \leq k < \kappa + K$ is not empty. We note $(\hat{u}_i^k)$ a minimum element of this set (and so $\hat{u}_i^k \leq u_i^k$); we will prove that $\hat{v}_i^{\kappa+K_i} \leq |u_b|\tau$.

If for all $k \geq \kappa$, $\hat{v}_i^k \leq \tilde{v}_{i-1}^k$, we conclude that vehicle i stops before vehicle $i-1$ which proves the result from the induction hypothesis. Otherwise, we let $k_0^i \geq \kappa$ be the minimum k such that $\hat{v}_i^k \geq \tilde{v}_{i-1}^k$, and we know that $\hat{v}_i^{k_0^i} \leq \hat{v}_{i-1}^{k_0^i - 1} + u_{max}\tau$. For all $k \geq k_0^i$, we know from the monotony of the system that the control $\min(\hat{u}_{i-1}^k, u_i^k, u_b)$ prevents rear-end collisions; we deduce that, for $k \geq k_0^i$, $\hat{v}_i^k \leq \hat{v}_{i-1}^{k-1} + u_{max}\tau$. Therefore, $\hat{v}_i^{\kappa+K_{i-1}+1} \leq \hat{v}_{i-1}^{K_{i-1}} + u_{max}\tau$ and we deduce from the induction hypothesis that $\hat{v}_i^{\kappa+K_{i-1}+1} \leq u_{max}\tau + |u_b|\tau$. Therefore, we obtain the recursion relation $K_i = K_{i-1} + 1 + \left\lceil \frac{u_{max}}{|u_b|} \right\rceil$ which yields the announced result.

Finally, we conclude that vehicle i can fully stop (without rear-end collisions) at step $\kappa + K_i + 1$; therefore the set of p vehicles can safely stop before the beginning of step $\kappa + K$ if $K \geq K_p = \frac{v_{max}}{|u_b|\tau} + (p-1)\left\lceil \frac{u_{max}}{|u_b|} \right\rceil + 1$. Since the recursion ensures that for all i and k, $\hat{u}_i^k \leq u_i^k$, we deduce that $s_i^{\kappa+K} \leq \hat{s}_i^{\kappa+K}$ which proves the proposition.

Note that the time needed for vehicles to match speeds can also be bounded by $\frac{v_{max}}{u_a}$ regardless of the value of p. Therefore, all vehicles can also fully stop within the time horizon if $T = K\tau \geq \frac{v_{max}}{|u_b|} + \frac{v_{max}}{u_a} + 2\tau$. Depending on the value of p, this bound may be better than the previously demonstrated one. □

Proof of Proposition 3. We consider a time $t_\kappa = \kappa\tau$, and we let $T_{rec} = K_{rec}\tau \geq T_{stop} + \frac{v_{min}}{u_a} + \frac{d}{v_{min}} + \tau$. Consider a solution $\mathbf{X}$ of FH-SP$_{K_{rec}}$ for the vehicles of $\mathcal{N}_{t_\kappa}$, defined for steps $\kappa \leq k \leq \kappa + K$. We will first show that this solution can be extended to a solution of FH-SP$_{K+1}$ for the vehicles in $\mathcal{N}_{t_\kappa}$. Note that the only constraints which can be unfeasible are the safety constraints (8)-(10) and the minimum velocity constraints (16) and (17). Consider a vehicle $i \in \mathcal{N}_{t_\kappa}$: using the control corresponding to this solution, two cases can arise:

- $s_i(T_{stop}) \leq s_i^{acc}$, in which case proposition 2 ensures that i and all the vehicles behind it can fully stop before reaching s_i^{acc}, and can remain stopped up to step $K_{rec}+1$. Since we also require that $s_j^{acc} \geq s_i^{acc}$ if j follows i, this ensures that keeping i and its followers stopped satisfies all the above constraints;
- otherwise, $s_i(T_{rec}) \geq \overline{s}_i^{\perp}$, in which case the crossing and minimum velocity constraints (8), (16) and (17) are satisfied for all conflicting vehicle j up to step K_{rec}. The requirement $T_{rec} \geq T_{stop}$ and proposition 2 ensure that the safe following constraints (9) and (10) involving

vehicle i remain satisfiable for the vehicles of $\mathcal{N}_{t_\kappa}$ up to step $K_{rec}+1$.

Indeed, if $\underline{s}_i^{\perp} \geq s_i(T_{stop}) > s_i^{acc}$, condition (16) ensures that vehicle i accelerates at least with acceleration u_a until reaching speed v_{min}, which takes at most a time $\frac{v_{min}}{u_a}$. The vehicle is then required to maintain speed v_{min} until reaching $\overline{s}_i^{\perp}$, which takes at most a time $\frac{d}{v_{min}}$. Therefore, vehicle i necessarily reaches $\overline{s}_i^{\perp}$ by time T_{rec}; the additional τ accounts for vehicles reaching doing so between two time steps.

The above considerations ensure that the solution of FH-SP$_K$ at time t_κ can be prolongated to a solution of FH-SP$_{K+1}$ for the vehicles of $\mathcal{N}_{t_\kappa}$. Finally, noting that the safe entry hypothesis ensures that this solution remains safe even when taking the vehicles of $\mathcal{N}_{t_\kappa+\tau} \setminus \mathcal{N}_{t_\kappa}$ into consideration. By definition, there also exists a safe control (and therefore a solution to FH-SP$_K$) for these vehicles at time $t_\kappa+\tau$. As a result, there exists a solution to FH-SP$_K$ at time $t_\kappa+\tau$ for the vehicles of $\mathcal{N}_{t_\kappa+\tau}$ which proves the stated result. □

References

[1] A. Vahidi and A. Eskandarian, "Research advances in intelligent collision avoidance and adaptive cruise control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 4, no. 3, pp. 143–153, sep 2003.

[2] E. Coelingh, A. Eidehall, and M. Bengtsson, "Collision Warning with Full Auto Brake and Pedestrian Detection - a practical example of Automatic Emergency Braking," in *13th International IEEE Conference on Intelligent Transportation Systems*. IEEE, sep 2010, pp. 155–160.

[3] D. Geronimo, A. M. Lopez, A. D. Sappa, and T. Graf, "Survey of Pedestrian Detection for Advanced Driver Assistance Systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 7, pp. 1239–1258, jul.

[4] M. Diaz-Cabrera, P. Cerri, and P. Medici, "Robust real-time traffic light detection and distance estimation using a single camera," *Expert Systems with Applications*, vol. 42, no. 8, pp. 3911–3923, may 2015.

[5] D. Kim, J. Choi, H. Yoo, U. Yang, and K. Sohn, "Rear obstacle detection system with fisheye stereo camera using HCT," *Expert Systems with Applications*, vol. 42, no. 17-18, pp. 6295–6305, oct 2015.

[6] Z. Kim, "Robust Lane Detection and Tracking in Challenging Scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 1, pp. 16–26, mar 2008.

[7] A. Colombo and D. Del Vecchio, "Least Restrictive Supervisors for Intersection Collision Avoidance: A Scheduling Approach," *IEEE Transactions on Automatic Control*, vol. 60, no. 6, pp. 1515–1527, jun 2014.

[8] G. R. Campos, F. Della Rossa, and A. Colombo, "Optimal and least restrictive supervisory control: Safety verification methods for human-driven vehicles at traffic intersections," in *2015 54th IEEE Conference on Decision and Control*. IEEE, dec 2015, pp. 1707–1712.

[9] T. Fraichard and H. Asama, "Inevitable Collision States. A Step Towards Safer Robots?" *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, no. October, pp. 388–393, 2003.

[10] R. Naumann, R. Rasche, and J. Tacken, "Managing autonomous vehicles at intersections," *IEEE Intelligent Systems and their Applications*, vol. 13, no. 3, pp. 82–86, 1998.

[11] K. Dresner and P. Stone, "Multiagent traffic management: a reservation-based intersection control mechanism," in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2004*. IEEE Computer Society, July 2004, pp. 530–537.

[12] L. Makarem and D. Gillet, "Fluent coordination of autonomous vehicles at intersections," in *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, vol. 18, no. 1. IEEE, oct 2012, pp. 2557–2562.

[13] X. Qian, J. Gregoire, A. de La Fortelle, and F. Moutarde, "Decentralized model predictive control for smooth coordination of automated vehicles at intersection," in *2015 European Control Conference (ECC)*. IEEE, jul 2015, pp. 3452–3458.

[14] K. Mu, F. Hui, and X. Zhao, "Merging Driver Assistance Decision System Using Occupancy Grid-Based Traffic Situation Representation," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, sep 2015, pp. 231–237.

[15] I. A. Ntousakis, I. K. Nikolos, and M. Papageorgiou, "Optimal vehicle trajectory planning in the context of cooperative merging on highways," *Transportation Research Part C: Emerging Technologies*, vol. 71, pp. 464–488, oct 2016.

[16] A. Mosebach, S. Röchner, and J. Lunze, "Merging control of cooperative vehicles," *IFAC-PapersOnLine*, vol. 49, no. 11, pp. 168–174, 2016.

[17] V. Desaraju, H. Ro, M. Yang, E. Tay, S. Roth, and D. Del Vecchio, "Partial order techniques for vehicle collision avoidance: Application to an autonomous roundabout test-bed," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, may 2009, pp. 82–87.

[18] R. Vasudevan, V. Shia, Yiqi Gao, R. Cervera-Navarro, R. Bajcsy, and F. Borrelli, "Safe semi-autonomous control with enhanced driver modeling," in *2012 American Control Conference*. IEEE, jun 2012, pp. 2896–2903.

[19] A. Gray, Y. Gao, J. K. Hedrick, and F. Borrelli, "Robust Predictive Control for semi-autonomous vehicles with an uncertain driver model," in *2013 IEEE Intelligent Vehicles Symposium*, no. 1239323. IEEE, jun 2013, pp. 208–213.

[20] A. Gray, Y. Gao, T. Lin, J. K. Hedrick, and F. Borrelli, "Stochastic predictive control for semi-autonomous vehicles with an uncertain driver model," in *16th International IEEE Conference on Intelligent Transportation Systems*. IEEE, oct 2013, pp. 2329–2334.

[21] C. Liu, A. Gray, C. Lee, J. K. Hedrick, and J. Pan, "Nonlinear stochastic predictive control with unscented transformation for semi-autonomous vehicles," in *2014 American Control Conference*. IEEE, jun 2014, pp. 5574–5579.

[22] T.-C. Au, S. Zhang, and P. Stone, "Autonomous Intersection Management for Semi-Autonomous Vehicles," *Handbook of Transportation*, pp. 88–104, 2015.

[23] S. A. Reveliotis and E. Roszkowska, "On the Complexity of Maximally Permissive Deadlock Avoidance in Multi-Vehicle Traffic Systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 7, pp. 1646–1651, jul 2010.

[24] L. Bruni, A. Colombo, and D. Del Vecchio, "Robust multi-agent collision avoidance through scheduling," in *52nd IEEE Conference on Decision and Control*. IEEE, dec 2013, pp. 3944–3950.

[25] H. Ahn and D. Del Vecchio, "Semi-autonomous intersection collision avoidance through job-shop scheduling," in *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*. ACM, 2016, pp. 185–194.

[26] J. Gregoire, "Priority-based coordination of mobile robots," Ph.D. dissertation, MINES ParisTech, 2014. [Online]. Available: http://arxiv.org/abs/1410.0879

[27] K. D. Kim and P. R. Kumar, "An MPC-based approach to provable system-wide safety and liveness of autonomous ground traffic," *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3341–3356, dec 2015.

[28] T. Siméon, S. Leroy, and J. P. Laumond, "Path coordination for multiple mobile robots: A resolution-complete algorithm," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 42–49, 2002.

[29] J. Peng, "Coordinating Multiple Robots with Kinodynamic Constraints Along Specified Paths," *The International Journal of Robotics Research*, vol. 24, no. 4, pp. 295–310, apr 2005.

[30] E. R. Müller, R. C. Carlson, and W. K. Junior, "Intersection control for automated vehicles with MILP," *IFAC-PapersOnLine*, vol. 49, no. 3, pp. 37–42, 2016.

[31] J. Park, S. Karumanchi, and K. Iagnemma, "Homotopy-Based Divide-and-Conquer Strategy for Optimal Trajectory Planning via Mixed-Integer Programming," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1101–1115, 2015.

[32] F. Borrelli, D. Subramanian, A. Raghunathan, and L. Biegler, "MILP and NLP Techniques for centralized trajectory planning of multiple unmanned air vehicles," in *2006 American Control Conference*. IEEE, 2006, pp. 5763–5768.

[33] S. Cafieri and N. Durand, "Aircraft deconfliction with speed regulation: New models from mixed-integer optimization," *Journal of Global Optimization*, vol. 58, no. 4, pp. 613–629, 2014.

[34] N. Murgovski, G. R. de Campos, and J. Sjoberg, "Convex modeling of conflict resolution at traffic intersections," in *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, dec 2015, pp. 4708–4713.

[35] Gurobi Optimization, Inc., "Gurobi optimizer reference manual," 2015. [Online]. Available: http://www.gurobi.com

[36] F. Altché, X. Qian, and A. De La Fortelle, "Least Restrictive and Minimally Deviating Supervisor for Safe Semi-Autonomous Driving at an Intersection: An MIQP Approach," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems*, Nov. 2016.

[37] P. Polack, F. Altché, B. d'Andréa-Novel, and A. de La Fortelle, "The Kinematic Bicycle Model : a Consistent Model for Planning Feasible Trajectories for Autonomous Vehicles ?" *2017 IEEE Intelligent Vehicles Symposium (IV)*, no. IV, 2017.

[38] F. Altché, X. Qian, and A. De La Fortelle, "Time-optimal Coordination of Mobile Robots along Specified Paths," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2016.

[39] TASS International. [Online]. Available: http://www.tassinternational.com/prescan

[40] G. Schildbach, M. Soppert, and F. Borrelli, "A collision avoidance system at intersections using Robust Model Predictive Control," in *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, jun 2016, pp. 233–238.

[41] W. Zhan, C. Liu, C.-y. Chan, and M. Tomizuka, "A non-conservatively defensive strategy for urban autonomous driving," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, nov, pp. 459–464.

[42] X. Qian, J. Gregoire, F. Moutarde, and A. De La Fortelle, "Priority-based coordination of autonomous and legacy vehicles at intersection," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 1166–1171.

[43] D. Jiang, V. Taliwal, A. Meier, W. Holfelder, and R. Herrtwich, "Design of 5.9 ghz dsrc-based vehicular safety communication," *IEEE Wireless Communications*, no. 5, pp. 36–43, oct.

[44] S. Demmel, A. Lambert, D. Gruyer, A. Rakotonirainy, and E. Monacelli, "Empirical IEEE 802.11p performance evaluation on test tracks," in *2012 IEEE Intelligent Vehicles Symposium*. IEEE, jun, pp. 837–842.

[45] D. Younger, "Minimum feedback arc sets for a directed graph," *IEEE Transactions on Circuit Theory*, vol. 10, no. 2, pp. 238–245, Jun 1963.

Florent Altché received the M.S. degree in engineering from French École Polytechnique in 2014 and a Specialized Master degree of public policies from École des Ponts ParisTech in 2015. He is currently working towards a Ph.D. degree at the Centre for Robotics at Mines ParisTech. His research interests include coordination and cooperation of autonomous vehicles or robots, as well as planning in an uncertain environment.

Xiangjun Qian received the B.S. degree in computer science from Shanghai Jiao Tong University, Shanghai, China in 2010, the Dip-Ing degree from MINES ParisTech, Paris, France, in 2012, and received his Ph.D. degree in robotics and automation at MINES ParisTech in 2016. His main research interest lies in the control and coordination of autonomous vehicles. He is also interested in the application of machine-learning techniques on the analysis of large-scale transportation networks.

Arnaud de La Fortelle (M10) received the M.S. degree in engineering from École Polytechnique and École des Ponts et Chausses, Paris, France, in 1994 and 1997 respectively and the Ph.D. degree from École des Ponts et Chausses in 2000, with a specialization in applied mathematics. He is professor at MINES ParisTech, Paris, France.

From 2003 to 2005, he investigated communications for cooperative systems and the architecture required in distributed systems at INRIA, participating in the CyberCars project. Since 2006, he has been the director of the Joint Research Unit LaRA (La Route Automatisée) of INRIA and MINES ParisTech and, since 2008, has also served as the Director of the Center of Robotics in MINES ParisTech. His research interests include cooperative systems (communication, data distribution, control, and mathematical certification) and their applications (Autonomous vehicles, collective taxis...). He coordinates the international research chair *Drive for All* (with partners UC Berkeley, Shanghai JiaoTong University and EPFL).

Dr. de La Fortelle has been elected to the Board of Governors of IEEE Intelligent Transportation System Society in 2009 and is member of the Board of the French Automotive Engineers Society. He is President of the French ANR evaluation committee for sustainable transport and mobility since 2015 and served as expert in European H2020 program.