Cooperative intersection collision avoidance in a constrained communication environment

Alessandro Colombo* and Henk Wymeersch†
*Department of Electronics, Information, and Bioengineering, Politecnico di Milano, Italy
†Department of Signals and Systems, Chalmers University of Technology, Sweden
Email: alessandro.colombo@polimi.it, henkw@chalmers.se

***Abstract*—Intersections remain among the most accident-prone subsystems in modern traffic. With the introduction of vehicle-to-infrastructure communication, it is possible for the intersection to become aware of the incoming stream of vehicles and issue warnings when needed. We consider an approach where vehicles can act automatically on those warnings, leaving drivers maximal freedom of manoeuvre while guaranteeing safety with minimal intervention. We also quantify the impact of imperfect communication in the uplink (from vehicle to infrastructure) and the downlink (from infrastructure to vehicle).**

I. Introduction

With rapidly increasing urbanization and mobility, there is an imminent need for intelligent transportation systems (ITS), which can reduce congestion, traffic accidents, and pollution. Traffic accidents, which are an important source of personal and economic losses, are a particular application where ITS can benefit. A significant fraction of all traffic accidents take place near intersections, which are among the most complex and regulated traffic subsystems. Accidents near intersections are most often caused by human errors and poor judgement. A suitable collision avoidance system (CAS) can oversee the entire intersection, warn drivers about imminent dangers, or temporarily take over the wheel. An intersection CAS inherently requires awareness from multiple vehicles and thus must comprise both a control layer and a wireless communication layer. Both layers are traditionally considered separately.

In terms of control, several works have proposed a rule-based approach to coordinate the actions of multiple agents near an intersection, providing constraints within which individual agents can move safely [1]–[7]. Recently, [8]–[12] proposed MPC-based coordination strategies. The above results are all based on a control law which leaves partial or no control to the drivers of each single vehicle. A different route was followed in [13], [14], where algorithms were developed to verify the safety of human-decided manoeuvres. These results, initially limited to two-vehicle problems, were extended in [15]–[19] to handle many vehicles in more complex settings. The advantage of this approach is that the resulting architecture minimizes by design interferences with the normal drivers behaviour, and can be implemented as an additional safety layer over existing road infrastructure (traffic lights, road signs, single-vehicle driver-assist systems, etc.) remaining transparent as long as the system behaves correctly, and intervening only when needed to avoid a collision. In other words, this approach leaves the drivers the maximum possible freedom of manoeuvre.

All the above-mentioned works implicitly rely on a communication infrastructure to share information and decisions. Wireless communication forms an integral part of ITS [20]–[25], and can be enabled through the IEEE 802.11p standard. 802.11p has been defined to meet the communication demands of ITS applications, operating in 20 MHz and 10 MHz channels in the 5.9 GHz band, as part of the the higher layer standard, IEEE 1609. Different applications clearly have different requirements on the communication links, with the most stringent demands imposed by safety-related applications, with extremely low latencies (below 50 ms in pre-crash situations), high delivery ratios (for full situational awareness), and relatively long communication ranges (to increase the time to react in critical situations) [26], [27]. These requirements, in combination with a possible high density of vehicles, makes the design of vehicular ad hoc network communication challenging [28], [29]. This is further exacerbated by high mobility and passing vehicles, which leads to rapidly changing signal propagation conditions (including both severe multipath and shadowing) and constant topology changes.

From the discussion above, it becomes apparent that wireless communication in vehicular systems is challenging and affected by a number of impairments. These impairments will have an impact on the control algorithms running on a CAS. In order to fully understand the overall performance of the CAS, it is necessary to look at both the control and the communication layer simultaneously.

In this paper, we take a holistic approach to the problem, discussing an implementation of a CAS as described, e.g., in [19], in relation to the limits of the communication layer, and we explicitly include the properties of the communication layer in assessing the performance of the control layer. In Sec. II we formulate the problem, in Sec. III and IV we provide our model for the vehicles and the communication link, and in Sec. V we describe the CAS and provide a numerical validation of its performance.

II. Problem formulation

We consider the scenario in Fig. 1, where a set of vehicles, driven by human drivers, move on different paths crossing at a common intersection. Our objective is to design a control architecture to supervise the vehicles in order to avoid collisions caused by an erroneous manoeuvre by any of the drivers. The architecture which we envision is composed of two layers:

1) *Rear-end collision avoidance system (RCAS):* This first, non-cooperative layer is active at all times in each vehicle and is responsible for maintaining a safety distance Δ (specified in Sec. V) between a vehicle and the preceding vehicle on the same road. This element of the control architecture is equivalent to many already existing advanced

978-1-4673-6596-3/15 $31.00 © 2015 IEEE
DOI 10.1109/ITSC.2015.70

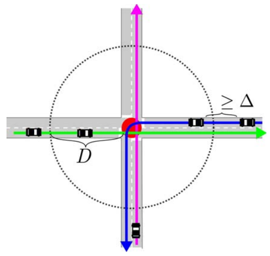

Fig. 1. Vehicles on 3 paths near an intersection. At all times a rear-end collision avoidance system forces vehicles to maintain a safety distance Δ, which depends on the relative velocity of the vehicles. Within a distance D of the intersection the vehicles establish a connection with the intersection collision avoidance system, which may override their input if necessary to avoid a collision at the intersection.

collision avoidance systems, and can be implemented simply as a radar-based active brake system on each vehicle.

2) *Intersection collision avoidance system (ICAS):* This second, cooperative layer exploits a communication link between vehicles and a central controller. Through cooperation among the vehicles, the ICAS can be less restrictive than a non-cooperative approach. The ICAS implements a least restrictive control, i.e., one that interferes with the drivers' decisions *only* if strictly necessary to avert a collision. This allows the architecture to work alongside the normal road infrastructure (traffic lights, stop signs, etc.), behaving transparently under safe conditions, and intervening only to avoid otherwise inevitable collisions.

The computational heart of the ICAS is a *supervisor*, an algorithm in charge of detecting erroneous manoeuvres, and of computing the necessary corrections. The supervisor is composed of two parts: a program running on centralized infrastructure (located at the intersection), and one running onboard each vehicle. The infrastructure and the vehicles communicate by means of a wireless network. The supervisor assumes perfect information (i.e., positions and velocities are measurable exactly), and is designed to cope with the inevitable limits of the communication link and to minimize the chances of misbehaviour despite these limits.

Due to the obvious geographical constraints of a centralized communication network, the ICAS will only supervise vehicles located within a distance D of the intersection (see Fig. 1). When a vehicle enters this so-called *controlled region*, defined as the interior of the dashed circle in Fig. 1, it communicates periodically (with period τ, typically around 0.1 s) its position, velocity, and the brake or accelerator input requested by the driver (and filtered by the RCAS) to the ICAS. The driver's input is assumed to be constant for the duration of the time interval τ. The information is used to project the state of all controlled vehicles τ seconds forward in time, and verify if their future state, reached using the drivers' desired inputs, is compatible with a safe (i.e., collision-free) evolution of the system. If this is not the case, the ICAS overrides the input of all controlled vehicles with a safe command, which is transmitted to all controlled vehicles using the communication link. Otherwise, the drivers' inputs are accepted, and can be used until the following time step. Since rear-end collisions are taken care by each vehicle, the ICAS is effectively in charge only of avoiding collisions at the intersection.

The main objective of this work is thus the following design problem:

Problem 1 (informal statement): Design a least restrictive supervisor for the ICAS.

The above problem will be formalized in Sec. V. The supervisor is designed keeping into account both the physics of the vehicles, and the inevitable limitations of the communication link. The next two sections discuss our model for these two aspects of the problem.

III. PHYSICAL MODEL

We model the longitudinal dynamics of each vehicle along its path as

$$\ddot{x}_i = f_i(\dot{x}_i, u_i), \tag{1}$$

with $i \in \{1, \dots, n\}$, where $x_i \in \mathbb{R}$ is the position and $u_i \in U_i \subset \mathbb{R}$ is the control input. We denote by $x_i(t, u_i)$ the position reached by vehicle i at time t, assuming initial conditions $(x_i, \dot{x}_i)$ (they will be explicitly specified when necessary), and using input u_i for a time t. We denote by $\mathbf{x}$, $\dot{\mathbf{x}}$, and $\mathbf{u}$ the vectors of positions, velocities, and inputs of all vehicles. At each time instant, we assume that vehicles $C \triangleq \{1, \dots, m\}$ are within the controlled region, while vehicles $N \triangleq \{m+1, \dots, n\}$ are not in the controlled region, and we use a subscript C or N to denote a state or input vector restricted to vehicles in C or N (e.g., $\mathbf{x}_C$ is the vector of the positions of all vehicles in the controlled region).

We assume that the set of inputs that each driver can choose at any given time is bounded below and above by the values $u_{i,\min}$ and $u_{i,\max}$, while we assume that the control systems (both the RCAS and the ICAS) can only force the use of a subset $[u_{i,\min}, u_{i,\text{com}}]$ of inputs, where $u_{i,\text{com}}$ is a non-emergency, or *comfortable* braking input, which satisfies $u_{i,\min} < u_{i,\text{com}} < 0$. We assume that $\dot{x}_i \in [0, \dot{x}_{i,\max}]$, and that (1) has unique solutions, depending continuously on initial conditions and parameters. We also assume that $f_i(\dot{x}_i, u_i)$ is non-decreasing in u_i, which implies that (1) is monotone [30], and that $\dot{x}_i = 0$ is reached in finite time using $u_{i,\min}$ (i.e., vehicles can stop in finite time). Finally, we assume that the dynamics of agents on the same path are identical, or equivalently that the dynamics of vehicles on the same path (described by $[f_i(\cdot), \dot{x}_{i,\max}, u_{i,\min}, u_{i,\max}]$) can be restricted to a common subset. This technical assumption is used to determine exactly the membership in (9).

A collision occurs if one of the two following conditions are met: (i) two vehicles on the same path hit each other; or (ii) two vehicles on different paths simultaneously occupy the intersection. These conditions can be represented mathematically as follows. We introduce an open interval (a_i, b_i), for the span of the intersection along the vehicle's path, and a minimum distance d between two vehicles on the same path. A collision occurs at time t when (i) $x_i(t) - x_j(t) < d$, or (ii) $x_i(t) \in (a_i, b_i)$ and $x_j(t) \in (a_j, b_j)$. We introduce the set $B \subset \mathbb{R}^n$ of collision points (the *Bad Set*) as $B \triangleq \big\{\mathbf{x} \in \mathbb{R}^n : \exists (i,j), i \neq j, (x_i \in (a_i, b_i) \text{ and } x_j \in (a_j, b_j)) \text{ or } (x_i - x_j < d)\big\}$. This is the set of all position vectors where at least two vehicles are in a collision configuration. We denote by $B_{\{\cdot\}}$ the bad set, restricted to the set of agents $\{\cdot\}$, and we say that $\mathbf{u}$ is a *safe input* for agents i and j if it is such that $\big(x_i(t, u_i), x_j(t, u_j)\big) \notin B_{\{i,j\}}$ for all $t \geq 0$.

Control of the vehicles is based on the detecting whether a given state $(\mathbf{x}, \dot{\mathbf{x}})$ admits at least one safe input for all agents. Using a common terminology from control theory [31], we define the *maximal controlled invariant set* as MCIS $\triangleq \{(\mathbf{x}, \dot{\mathbf{x}}) \in \mathbb{R}^{2n} : \exists \mathbf{u} \in [\mathbf{u}_{\min}, \mathbf{u}_{\mathrm{com}}], \mathbf{x}(t, \mathbf{u}) \notin B, \forall\, t \geq 0\}$. In simple terms, the role of a least restrictive supervisor, as required by Problem 1, is to maintain the state of the system within the MCIS, intervening if and only if the drivers' desired input would take the system out of this set. This is the essence of the formal definition of Problem 1 given in Sec. V. Note that, under the assumptions in Sec. III, exact and efficient approximate algorithms to determine if a state $(\mathbf{x}, \dot{\mathbf{x}})$ belongs to the MCIS are found in [15], [19].

IV. COMMUNICATION MODEL

The communication is broken down into an *uplink* (from vehicles to ICAS) and a *downlink* (from ICAS to vehicles).

A. Uplink model

In the uplink, we assume that users have been registered to a channel prior to arriving within a distance D from the intersection (hence the number of vehicles under ICAS control, m, is known), and that at least m non-interfering channels are available. With period τ, each vehicle i within a distance D from the intersection sends a signal $\mathbf{s}_i$. The observed signal by the ICAS infrastructure is $\mathbf{r}^{\mathrm{u}} = [\mathbf{r}^{\mathrm{u}}_1, \dots, \mathbf{r}^{\mathrm{u}}_n]$, with $\mathbf{r}^{\mathrm{u}}_i = h^{\mathrm{u}}_i \mathbf{s}_i + \mathbf{n}^{\mathrm{u}}_i, i \in C$, in which h^{u}_i represents the uplink channel between vehicle i and the infrastructure, and $\mathbf{n}^{\mathrm{u}}_i$ is white Gaussian noise with power σ^2. The entries in the vectors are sent successively, with independent and identically distributed noise, but with a constant channel. When each transmitted signal has power P, the quality of the transmission is characterized by the uplink signal-to-noise ratio (SNR) $\mathrm{SNR}^{\mathrm{u}}_i = \frac{P|h^{\mathrm{u}}_i|^2}{\sigma^2}$.

B. Downlink model

In the downlink, the infrastructure broadcasts a common control message $\mathbf{s}$ to all vehicles. This transmission arrives at vehicle i as $\mathbf{r}^{\mathrm{d}}_i = h^{\mathrm{d}}_i \mathbf{s} + \mathbf{n}^{\mathrm{d}}_i$, with a corresponding SNR $\mathrm{SNR}^{\mathrm{d}}_i = \frac{P|h^{\mathrm{d}}_i|^2}{\sigma^2}$.

C. Success probability

A transmission to/from vehicle i is considered successfully received if the SNR exceeds a threshold value γ. Due to the random nature of the channel, successes are random variables. The success probability $\mathbb{P}_i$ of vehicle i is given by

$$\mathbb{P}_i = \Pr\big(\mathrm{SNR}_i \geq \gamma\big), \tag{2}$$

in which SNR_i is the SNR associate with vehicle i, in either uplink or downlink, and $\Pr(\cdot)$ denotes the probability of the argument. The expression of the success probability (2) depends on the statistical model of the channel gain $|h_i|^2$. A common model is Rayleigh fading, in which the probability density function of $|h_i|^2$ is given by $p(|h_i|^2) = 1/\lambda_i \exp(-|h_i|^2/\lambda_i)$, where $\lambda_i = K|x_i - x_{\mathrm{s}}|^{-\eta}$ for a constant K, vehicle position x_i, supervisor position x_{s} and path-loss exponent η (generally $\eta \approx 2$). Under the Rayleigh fading model, the success probability becomes $\mathbb{P}_i = \exp\big(-\gamma\sigma^2/(PK|x_i - x_{\mathrm{s}}|^{-\eta})\big)$.

V. SUPERVISOR ARCHITECTURE

In this section, we detail the operation of the RCAS and ICAS, and provide an analysis of their behaviour.

A. RCAS

Given any two vehicles i and j on a path, with j preceding (driving in front of) i, the role of the RCAS is to ensure that the distance $x_j - x_i \geq \Delta(\dot{x}_i, \dot{x}_j)$ at all times, where $\Delta : (\dot{x}_i, \dot{x}_j) \to \mathbb{R}$ denotes the smallest distance which satisfies the implication

$$\begin{aligned} x_j - x_i \geq \Delta(\dot{x}_i, \dot{x}_j) \Rightarrow \\ x_i(t, u_{i,\mathrm{com}}) \leq x_j(t, u_{j,\min}) - d, \forall t \geq 0. \end{aligned} \tag{3}$$

Lemma 1: If Δ satisfies the implication (3) at all times, then $(u_{i,\mathrm{com}}, u_j)$ is a safe input for the agents i and j, for any possible u_j.

Proof: Since (1) is monotone, (3) implies that $x_i(t, u_{i,\mathrm{com}}) \leq x_j(t, u_j) - d$ for all $t \geq 0$ and for all possible u_j. ∎

By the above result, whatever the input used by the predecessor j, vehicle i can maintain the safety distance Δ simply by applying $u_{i,\mathrm{com}}$. Therefore, the RCAS can be implemented simply as an active braking system, which applies $u_{i,\mathrm{com}}$ when it detects that the vehicle is getting too close to its predecessor. From now on, the input $u_{i,\mathrm{des}}$ requested by a vehicle is always assumed to have been already corrected, if necessary, by the RCAS, to avoid collisions with a preceding vehicle.

B. ICAS

The supervisor must satisfy Problem 1. This is formalized as follows:

Problem 2 (Formalization of Problem 1): Design a map $\mathbf{u}_{C,\mathrm{des}} \mapsto \mathbf{u}_C$ such that

$$\begin{bmatrix} \mathbf{x}_C(t, \mathbf{u}_C) \\ \mathbf{x}_N(t, \mathbf{u}_N) \end{bmatrix} \notin B, \forall\, t \in [0, \tau] \tag{4}$$

and

$$\begin{bmatrix} \mathbf{x}_C(\tau, \mathbf{u}_C) \\ \dot{\mathbf{x}}_C(\tau, \mathbf{u}_C) \\ \mathbf{x}_N(\tau, \mathbf{u}_N) \\ \dot{\mathbf{x}}_N(\tau, \mathbf{u}_N) \end{bmatrix} \in \mathrm{MCIS} \tag{5}$$

and

$$\mathbf{u}_C = \mathbf{u}_{C,\mathrm{des}} \iff \text{it satisfies (3), (4), and (5),} \tag{6}$$

for any possible $\mathbf{x}_N$ and $\mathbf{u}_N$.

In the above notation, we are calling $t = 0$ the beginning of the current control time interval, $t = \tau$ the end. Condition (4) requires that, if $(\mathbf{x}_C, \mathbf{x}_N)$ is the state of the system at the beginning of a control time interval, the input $(\mathbf{u}_C, \mathbf{u}_N)$ does not cause a collision for the length τ of the interval. Condition (5) instead requires that the state reached using $(\mathbf{u}_C, \mathbf{u}_N)$ *after* a time τ still admits a safe input for all agents. Failure to do so would result in an inevitable collision at some time $t > \tau$. Thus, the two conditions (4)–(5) together ensure that the supervisor avoid collisions. Then, by requiring (6) we also ensure least restrictiveness.

To design a supervisor to solve the above problem we must reject all inputs that would lead the state outside of the MCIS. The supervisor must also cope with the fact that only the subset C of agents is in communication with the ICAS, and therefore only the sub-vector $\mathbf{u}_C$ can be corrected. To ensure that such a supervisor can be designed, and that it correctly handles vehicles entering and leaving the controlled region, we require that the diameter D of the controlled region satisfy

$$D \geq \lim_{t\to\infty} x_i(t, u_{i,\text{com}}) - x_i(0) + \dot{x}_{i,\max}\tau,\ \forall\, \dot{x}_i(0),\ \forall\, i, \quad (7)$$

where $(x_i(0), \dot{x}_i(0))$ are the initial conditions of the trajectory $x_i(t, u_{i,\text{com}})$. In other words, D must be at least equal to the worst stopping distance of any vehicle, using $u_{i,\text{com}}$, plus the maximum distance $\dot{x}_{i,max}\tau$ covered by a vehicle during a single control time interval. Under this condition, the following lemma provides the key to the design of the above-specified supervisor.

Lemma 2: Assuming (7) and $x_j - x_i \geq \Delta(\dot{x}_i, \dot{x}_j)$ for all pairs i, j of agents on the same path, with j preceding i, conditions (4) and (5) hold if and only if the following conditions hold

$$\mathbf{x}_C(t, \mathbf{u}_C) \notin B_C,\ \forall\, t \in [0, \tau] \quad (8)$$

and

$$\begin{bmatrix} \mathbf{x}_C(\tau, \mathbf{u}_C) \\ \dot{\mathbf{x}}_C(\tau, \mathbf{u}_C) \end{bmatrix} \in \text{MCIS}_C \quad (9)$$

Proof: (4) and (5) $\Rightarrow$ (8) and (9) is obvious. The other implication follows using (3), (7), and the assumption $x_j - x_i \geq \Delta(\dot{x}_i, \dot{x}_j)$. Indeed, assuming (8) and (9) there exists an input $\mathbf{u}_C$ with $\mathbf{u}_C(t) \in [\mathbf{u}_{C,\min}, \mathbf{u}_{C,\text{com}}],\ \forall\, t > \tau$ such that $\mathbf{x}_C(t, \mathbf{u}_C) \notin B_C,\ \forall\, t \geq 0$. At the same time, by the right-hand side of (3), the input $\mathbf{u}_{N,\text{com}}$ applied to all agents in N satisfies $\mathbf{x}_N(t, \mathbf{u}_{N,\text{com}}) \notin B_N,\ \forall\, t \geq 0$, and by (7) using such an input, no agent $i \in N$ will intersect the interval $[a_i, b_i]$ (i.e., the intersection) for any $t \geq 0$. Therefore, by the right-hand side of (3) the input $\mathbf{u} = (\mathbf{u}_C, \mathbf{u}_{N,\text{com}})$ satisfies (4) and (5). ■

The assumption $x_j - x_i \geq \Delta(\dot{x}_i, \dot{x}_j)$ is always true under the control of the RCAS. Therefore, from the above lemma we conclude that, in order to solve Problem 1, the ICAS does not need to collect information or to send commands to agents in the set N. In [19] is explained how to check (9) exactly, under the assumptions in Section III.

The flow charts in Figs. 2–3 define the supervisory architecture on the infrastructure side and on the vehicle side, respectively.

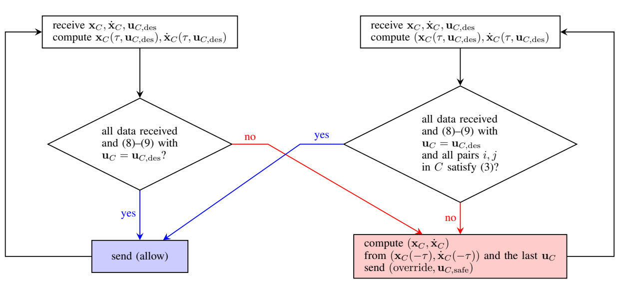

Fig. 2. ICAS operation: infrastructure side. At every control time interval, information is collected from all vehicles and an allow or override command is issued based on the predicted state of all vehicles.

On the infrastructure side, the supervisor can be in two modes: the *allow* mode (left side of the flow chart) or the *override* mode (right side of the chart). In allow mode, the supervisor is allowing the desired inputs. In override mode, it sends an override message to all vehicles in the controlled

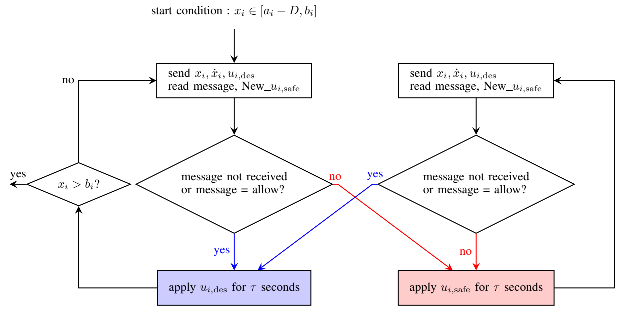

Fig. 3. ICAS operation: vehicle side. Vehicles periodically send their state and intention information once they are close to the intersection. They then receive the allow or override command from the ICAS infrastructure and act accordingly.

region, and a vector $\mathbf{u}_{\text{safe}}$ describing a safe input for all the vehicles for which it has state information. Note that, as proved in [19], if a safe input exists then it can be constructed as a finite sequence of inputs $u_{i,\min}$ and $u_{i,\text{com}}$, therefore $\mathbf{u}_{C,\text{safe}}$ can always be encoded as a finite message, despite being an infinite-horizon input signal. To compute such a safe input the supervisor must know the state of all agents in C. If the state of some agents was not received due to a broken communication link, the supervisor fills in the state vector by integrating their last known state and input. The only case where the state of a vehicle $i \in C$ cannot be reconstructed is if none of the messages sent from i have been received by the infrastructure since i entered the controlled region. In this case, we assume that the algorithm computes $\mathbf{u}_{\text{safe}}$ for the set of vehicles with known state, and completes it by setting $u_{i,\text{safe}} = u_{i,\min}$ for the missing ones (this procedure is not depicted in the flow chart). Switching from override to allow mode is conditional to receiving information from all vehicles in C, satisfying (8)–(9) for $\mathbf{u}_C = \mathbf{u}_{C,\text{des}}$, and having (3) satisfied by all pairs of agents in C which are on the same path.

On the vehicle side, a vehicle i starts executing the supervisory algorithm when its position enters the interval $[a_i - D, a_i]$ (this is when the vehicle joins the set C, by the definition in Sec. III). At this point the algorithm structure mirrors the one in the infrastructure, with an allow mode and an override mode. The vehicle uses $u_{i,\text{des}}$ in allow mode, and $u_{i,\text{safe}}$ in override mode.

We can now prove that the supervisor defined in Figs. 2, 3 indeed solves Problem 2.

Theorem 3: If all vehicles in the controlled region receive all messages from the infrastructure, the ICAS satisfies (4)–(5) in Problem 2.

Proof: Consider a single control time interval of length τ. Assume at first that all vehicles in C have correctly communicated their state to the supervisor at least once, so that their current state can be computed from past information if not received. By assumption, the inputs $\mathbf{u}_{C,\text{des}}$ requested by the vehicles do not cause rear-end collisions. If the message returned by the supervisor is *allow*, then $\mathbf{u}_{C,\text{des}}$ also does not cause collisions at the intersection. If the supervisor returns *override*, then as long as all vehicles in the controlled region receive the message from the supervisor, they coherently apply $\mathbf{u}_{C,\text{safe}}$. This prevents collisions among vehicles in C. Collisions among these vehicles and those in N are also avoided (thanks to the RCAS), by Lemma 2.

Now consider the case where a vehicle $i \in C$ cannot send a message to the infrastructure upon entering the controlled region. Lacking this message, the supervisor switches to override mode and sets $u_i = u_{i,\min}$, so $\mathbf{u}_{C,\text{safe}}$ is defined for all vehicles in C. The only chance of a crash is if vehicle i reaches the intersection (and crashes with a vehicle from a different path) while applying $u_{i,\min}$, but by (7) $\lim_{t\to\infty} x_i(t, u_{i,\min}) \leq a_i$, therefore the agent will stop before reaching the intersection, while in override mode. ∎

Theorem 4: If all vehicles in the controlled region receive all messages from the infrastructure, and the infrastructure receives all messages from the vehicles, then the ICAS satisfies (4)–(5) *and* (6) in Problem 2.

Proof: We proved safety in Theorem 3. To prove least restrictiveness (6), note that, assuming messages from all vehicles in the control region are correctly received by the infrastructure, the override manoeuvre is initiated by an *override* message from the infrastructure if and only if $\big(\mathbf{x}_C(\tau, \mathbf{u}_{C,\text{des}}), \dot{\mathbf{x}}_C(\tau, \mathbf{u}_{C,\text{des}})\big) \notin \text{MCIS}_C$. By the definition of MCIS_C, this implies that a collision would certainly occur if the controlled agents used $\mathbf{u}_{C,\text{des}}$ instead of $\mathbf{u}_{C,\text{safe}}$. ∎

There is still one open issue: when the supervisor overrides the input requested by the vehicles, the corresponding trajectory of the vehicles in C may temporarily violate (3). To complete our control architecture, we must ensure that, for a suitable $\mathbf{u}_{C,\text{des}}$, control is always eventually returned to the vehicles, i.e., that the algorithms in Figs. 3 and 2 eventually switch back to the allow mode if $\mathbf{u}_{C,\text{des}}$ does not violate (8)–(9).

Theorem 5: If (8), (9), and all messages are received, the supervisor always eventually switches to the allow mode.

Proof: This requires satisfying the conditions in the right decision (diamond shaped) block of Fig. 3. Given the above assumptions, it suffices to check that all agents in C eventually satisfy (3). This is granted by the fact that, in override mode, all agents apply an input $u_i < 0$ and therefore eventually reach null velocity. Thus, eventually (3) reduces to $x_j - x_i \geq d$, which is ensured by (8). ∎

C. Expected performance of the supervisor

Theorem 6: Let $\mathbb{P}_{\text{crash}}$ be the probability of a crash for the unsupervised system. The supervisory architecture ensures safety with probability $\mathbb{P}_{\text{crash}}\,\big(1-\exp(-\alpha)\big)$ and least restrictiveness with probability $\exp(-\alpha)$ where

$$\alpha = \frac{\gamma\sigma^2}{PK} \sum_{i\in C} |x_i - x_{\text{s}}|^{\eta}. \tag{10}$$

Proof: Assuming that there is no error in positioning or in the actuators, the system becomes restrictive only if it overrides unnecessarily. The probability of being not least restrictive is simply that of losing a message from at least one vehicle, or, equivalently, the probability of being least restrictive is that of not losing any messages. Hence, $\Pr(\text{least restrictive}) = \prod_{i\in C} \mathbb{P}_i = \exp\Big(-\sum_{i\in C} \gamma\sigma^2/(PK|x_i - x_{\text{s}}|^{-\eta})\Big) = \exp(-\alpha)$. The system becomes unsafe only if, with an imminent crash, at least one vehicle does not follow the override message sent by the supervisor. This can only happen if the vehicle does not receive the override message. The probability of becoming unsafe is thus $\Pr(\text{unsafe}) = \mathbb{P}_{\text{crash}}(1 - \exp(-\alpha))$. ∎

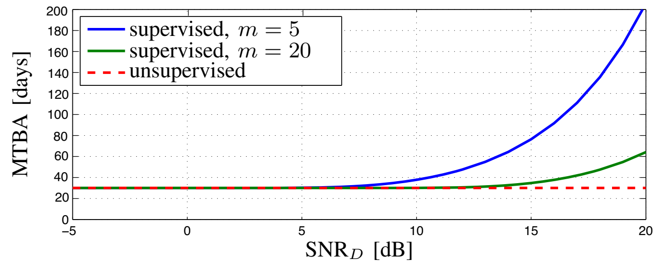

Fig. 4. Mean time between accidents for $\gamma = 8$ dB, $\eta = 2$, $m \in \{5, 20\}$ for the supervised and unsupervised system, as a function of the average SNR at a distance D from the intersection.

We can approximate α in (10) with its expected value, under the assumption that for vehicles in C, $|x_i - x_{\text{s}}|$ is uniformly distributed in the interval $[0, D]$: $\alpha \approx \frac{\gamma\sigma^2 m D^\eta}{PK\eta} = \frac{m}{\eta}\frac{\gamma}{\text{SNR}_D}$, in which SNR_D is the average SNR at a distance D from the intersection.

Theorem 6 can more easily be understood in terms of the mean time between accidents (MTBA), which is $\tau/\mathbb{P}_{\text{crash}}$ for an unsupervised system and $\tau/(\mathbb{P}_{\text{crash}}(1-\exp(-\alpha)))$ for the supervised system. Fig. 4 shows the MTBA, when the unsupervised system has an MTBA of 30 days, for $\gamma = 8$ dB, $\eta = 2$, $m \in \{5, 20\}$, and a varying SNR_D ranging for -5 dB to 20 dB. We observe that the SNR at the distance D should be sufficiently large to see safety benefits from the supervisor. We also note that when more vehicles are supervised, the MTBA is reduced significantly.

In Fig. 5 we report the simulated trajectories of 5 vehicles at an intersection, initially positioned as in Fig. 1, under three communication scenarios. Trajectories of different colours are assigned to vehicles on different paths, with colours matching those of the paths in Fig. 1. All vehicle's initial velocities are set to $12 m/s$, and all drivers request $u_i = 2$ at all times if compatible with the safety distance. In the top panel the communication link works correctly at all times. The first two vehicles cross the intersection without triggering the ICAS, the following vehicles instead approach the intersection simultaneously, and would collide if not for the ICAS intervention at $t \in [4.3, 7.4]$ seconds. In the second panel, for $t \in [0.5, 1.5]$ the second vehicle on the green path enters the controlled region but cannot communicate its state. The supervisor issues an override command, and instructs this vehicle to apply $u_{\min}$. Again, collisions are averted. In the third panel the same vehicle also stops receiving all messages from the supervisor at $t = 4$. One second later the supervisor issues an override command, which is not received by the vehicle. This leads to a collision from $t \simeq 6.3$ (green and purple vehicle in the intersection).

VI. Conclusion

We have discussed a possible implementation of a least restrictive collision avoidance system for vehicles at an intersection, taking into account the tight interplay of the control and communication layers in determining the overall performance of the system. For the sake of simplicity, in our discussion we have omitted the added complicacy of sensors, actuators, and modelling uncertainties, which obviously play a big role in this kind of problems. Such nonidealities, in a control framework

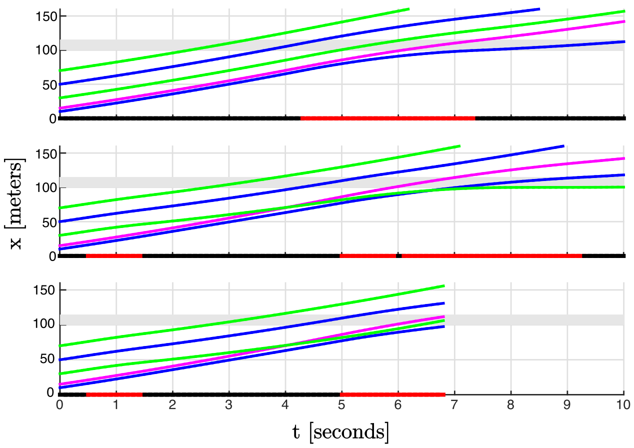

Fig. 5. Simulations of 5 agents on 3 paths. The horizontal axes are black when the supervisor is in allow mode, red when it is in override mode. The intersection (the gray band) is the interval $[100, 115]m$ along all paths. In all simulations vehicle dynamics are given by $\ddot{x}_i = u_i - 0.0005(\dot{x}_i)^2$ if $(\dot{x}_i > 0$ and $u_i - 0.0005(\dot{x}_i)^2 \leq 0)$ or if $(\dot{x}_i < \dot{x}_{i,\max}$ and $u_i - 0.0005(\dot{x}_i)^2 \geq 0)$; $\ddot{x}_i = 0$ otherwise. The quadratic term accounts for drag. Parameters are $\dot{\mathbf{x}}_{\max} = 16m/s$, $\mathbf{u}_{\min} = -4$, $\mathbf{u}_{\max} = 2$, $\mathbf{u}_{\text{com}} = -2$, $d = 5m$, $D = 65m$.

analogous to the one used here, have however been confronted in the literature (e.g., [14], [16]), and are known to be a solvable albeit challenging problem.

From the results above it is evident that the communication and control layer of a cooperative CAS are inextricably linked, and their design must proceed hand in hand to optimize the overall performance of the system.

VII. Acknowledgements

A.C. acknowledges grant AD14VARI02 - Progetto ERC BETTER CARS - Sottomisura B, H.W. was partially supported by the European Research Council under Grant No. 258418 (COOPNET).

References

[1] K. Dresner and P. Stone, "Multiagent traffic management: An improved intersection control mechanism," in *AAMAS*, 2005.

[2] ——, "Mitigating catastrophic failure at intersections of autonomous vehicles," in *AAMAS*, 2008.

[3] A. De La Fortelle, "Analysis of reservation algorithms for cooperative planning at intersections," in *ITSC*, 2010, pp. 445–449.

[4] H. Kowshik, D. Caveney, and P. R. Kumar, "Provable systemwide safety in intelligent intersections," *IEEE Trans. Veh. Technol.*, vol. 60, pp. 804–818, 2011.

[5] J. Gregoire, S. Bonnabel, and A. De La Fortelle, "Priority-based intersection management with kinodynamic constraints," in *ECC*, 2014, pp. 2901–2907.

[6] R. Azimi, G. Bhatia, R. Rajkumar, and P. Mudalige, "Vehicular networks for collision avoidance at intersections," SAE Technical Paper, Tech. Rep., 2011.

[7] ——, "Intersection management using vehicular networks," SAE Technical Paper, Tech. Rep., 2012.

[8] R. Hult, G. R. Campos, P. Falcone, and H. Wymeersch, "An approximate solution to the optimal coordination problem for autonomous vehicles at intersections," in *ACC, Accepted*, 2015.

[9] G. R. de Campos, P. Falcone, and J. Sjöberg, "Autonomous cooperative driving: a velocity-based negotiation approach for intersections crossing," in *ITSC*, 2013.

[10] G. R. Campos, P. Falcone, H. Wymeersch, R. Hult, and J. Sjöberg, "A receding horizon control strategy for cooperative conflict resolution at traffic intersections," in *CDC*, 2014.

[11] Kyoung-Dae Kim, "Collision free autonomous ground traffic: A model predictive control approach," in *ICCPS*, 2013.

[12] Kyoung-Dae Kim and P. R. Kumar, "An MPC-based approach to provable system-wide safety and liveness of autonomous ground traffic," *IEEE Trans. Autom. Control*, vol. 59, pp. 3341–3356, 2014.

[13] R. Verma and D. Del Vecchio, "Semiautonomous multivehicle safety: A hybrid control approach," *IEEE Robot. Autom. Mag.*, vol. 18, pp. 44–54, 2011.

[14] M. Hafner, D. Cunningham, L. Caminiti, and D. Del Vecchio, "Cooperative collision avoidance at intersections: Algorithms and experiments," *IEEE Trans. Intell. Transp. Syst*, 2013.

[15] A. Colombo and D. Del Vecchio, "Efficient algorithms for collision avoidance at intersections," in *HSCC*, 2012.

[16] L. Bruni, A. Colombo, and D. Del Vecchio, "Robust multi-agent collision avoidance through scheduling," in *CDC*, 2013.

[17] H. Ahn, A. Colombo, and D. Del Vecchio, "Supervisory control for intersection collision avoidance in the presence of uncontrolled vehicles," in *ACC*, 2014.

[18] A. Colombo, "A mathematical framework for cooperative collision avoidance of human-driven vehicles at intersections," in *ISWCS*, 2014, pp. 449–453.

[19] A. Colombo and D. Del Vecchio, "Least restrictive supervisors for intersection collision avoidance: A scheduling approach," *IEEE Trans. Autom. Control*, 2015.

[20] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil, "Vehicular Networking: A Survey and Tutorial on Requirements, Architectures, Challenges, Standards and Solutions," *Commun. Surveys Tuts.*, vol. 13, no. 4, pp. 584–616, 2011.

[21] P. Papadimitratos, A. La Fortelle, K. Evenssen, R. Brignolo, and S. Cosenza, "Vehicular Communication Systems: Enabling Technologies, Applications, and Future Outlook on Intelligent Transportation," *IEEE Commun. Mag.*, vol. 47, no. 11, pp. 84–95, Nov. 2009.

[22] H. Hartenstein and K. Laberteaux, "A Tutorial Survey on Vehicular Ad Hoc Networks," *IEEE Commun. Mag.*, vol. 46, no. 6, pp. 164–171, Jun. 2008.

[23] K. Dar, M. Bakhouya, J. Gaber, M. Wack, and P. Lorenz, "Wireless Communication Technologies for ITS Applications [Topics in Automotive Networking]," *IEEE Commun. Mag.*, vol. 48, no. 5, pp. 156–162, May 2010.

[24] F. Anjum, S. Choi, V. Gligor, R. Herrtwich, J.-P. Hubaux, P. Kumar, R. Shorey, and C.-T. Lea, "Guest Editorial Vehicular Networks," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 8, pp. 1497–1500, Oct. 2007.

[25] M. Alsabaan, W. Alasmary, A. Albasir, and K. Naik, "Vehicular Networks for a Greener Environment: A Survey," *Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1372–1388, Jan. 2013.

[26] C. F. Mecklenbrauker, A. F. Molisch, J. Karedal, F. Tufvesson, A. Paier, L. Bernado, T. Zemen, O. Klemp, and N. Czink, "Vehicular Channel Characterization and Its Implications for Wireless System Design and Performance," *Proc. IEEE*, vol. 99, no. 7, pp. 1189–1212, Jul. 2011.

[27] J. Santa, R. Toledo-Moreo, M. A. Zamora-Izquierdo, B. Úbeda, and A. F. Gómez-Skarmeta, "An analysis of communication and navigation issues in collision avoidance support systems," *Transport. Res C-Emer.*, vol. 18, no. 3, pp. 351–366, Jun. 2010.

[28] E. Steinmetz, M. Wildemeersch, and H. Wymeersch, "WiP abstract: Reception probability model for vehicular ad-hoc networks in the vicinity of intersections," in *ICCPS*, April 2014, pp. 223–223.

[29] K. Sjöberg, "Medium Access Control for Vehicular Ad Hoc Networks," Ph.D. dissertation, Chalmers University of Technology, 2013.

[30] D. Angeli and E. D. Sontag, "Monotone control systems," *IEEE Trans. Autom. Control*, vol. 48, pp. 1684–1698, 2003.

[31] J. Lygeros, C. Tomlin, and S. Sastry, "Controllers for reachability specifications for hybrid systems," *Automatica*, vol. 35, pp. 349–370, 1999.