Autonomous cooperative driving: a velocity-based negotiation approach for intersection crossing

Gabriel Rodrigues de Campos, Paolo Falcone and Jonas Sjöberg

Abstract— **In this article, a scenario where several vehicles have to coordinate among them in order to cross a traffic intersection is considered. In this case, the control problem relies on the optimization of a cost function while guaranteeing collision avoidance and the satisfaction of local constraints.**

A decentralized solution is proposed where vehicles sequentially solve local optimization problems allowing them to cross, in a safe way, the intersection. This approach pays a special attention to how the degrees of freedom that each vehicle disposes to avoid a potential collision can be quantified and led to an adequate formalism to the considered problem. In the proposed strategy, collision avoidance is enforced through local state constraints at given time instants and agents are assumed to only communicate the available time to react and the time stamps at which they expect to be within the intersection.

Simulations results on the efficiency and performance of the proposed approach are also presented.

I. INTRODUCTION

Self-organized swarming behaviors in biological groups with distributed individual-to-individual interactions [3], [4], [14] have become the scientific motivation for studying multi-agent systems and the inherent coordination mechanisms. Recent surveys on distributed coordination can be found in [13], [19]. The cooperative strategies observed in the nature might have different form, structure or scale, but they aim for the same thing: optimize a task by using all tools available, *i.e.*, all the individuals/agents. The fundamental property of the cooperation among several agents is that the group behavior is not dictated by one of the individuals. On the contrary, the behavior results implicitly from the local interactions between the individuals and their neighbors.

This article focus on cooperative behaviors for autonomous cars at road intersections. For such scenarios, the advantages of distributed approaches are apparent: i) Vehicles can coordinate in order to trade-off their own objectives and a global goal, while avoiding conflicts; ii) Robustness to failures of single agents or communication can be guaranteed; iii) The dynamical features of the communication graph such as low data rates, dropouts or proximity-based communication can be adequately handled. A recent review of the vast literature on multi -agent systems can be found in [9], [15], [17], [18].

More precisely, this paper deals with cooperative driving strategies such that vehicles, equipped with communication devices, have to exchange information in order to coordinate and agree on how to cross the intersection without collisions. Ideally, by exploiting their communication capabilities, the vehicles should be able to coordinate in order to, *e.g.*, guarantee Quality of Service (QoS) requirements, minimize the aggregate fuel-consumption (by, *e.g.*, slowing down a light vehicle instead of a bus or a heavy truck). We abstract from the (many) implementation issues and focus on the fundamental aspects of the underlying decision making problems. The objective is to provide to each agent/vehicle mechanisms enabling distributed cooperative decision making leading to a solution that is guaranteed to be collision-free.

The collision avoidance problem in formations of autonomous mobile systems is a well studied topic in robotics and in air traffic management, see *e.g.*, [1], [6] or [16]. The reader can refer to [12] for an elaborate survey of conflict resolution approaches. In other works, a command governor approach is presented in [22] and a navigation functions's based methodology studied in [5]. A cooperative collision warning system, specially useful for aircraft collision avoidance techniques, is introduced in [10] and mixed-integer linear programming is proposed for multi-vehicle formations with uncoupled dynamics subject to coupled collision avoidance constraints in [20] or [21]. However, due to their nature, the computation effort of the methods presented in the last two contributions scales exponentially with the size of the problem and number of vehicles.

In this article, an autonomous control strategy for intersections where conventional traffic control devices (stop signs or traffic lights) have been removed is presented. An illustration of studied scenario is shown in Figure 1. Recent work on this topic can be found in [7], [8], where authors present experimental results for an active control system helping drivers to avoid collisions, or in [11] where provably safe scheduling algorithms for intelligent intersections are introduced. In the last work, the proposed solution is based on the interaction between cars and the intersection infrastructure, utilizing time-slot assignment established by the intersection infrastructure itself. They propose a hybrid architecture with an appropriate interplay between centralized calculations and distributed coordination, assuming that each car has an infinite horizon contingency plan which is updated at each sampling instant and distributed by the cars. Other works in this topic consider dynamic programming or even game-theoretic approaches applied to hybrid systems, as presented in [23], [24].

The rest of this paper is organized as follows. Section II presents the considered problem while Section III includes the proposed control strategy. Simulation results are provided in Section IV and final conclusions and perspectives for future research are given in Section V.

The authors are with the Department of Signals and Systems, Chalmers University of Technology, 412 96 Göteborg, Sweden. Email: gabriel.campos@chalmers.se

This work is supported by Chalmers' Area of Advance in Transportation, SAFER-Vehicle and Traffic Safety Center and the Department of Signals and Systems (S2) .

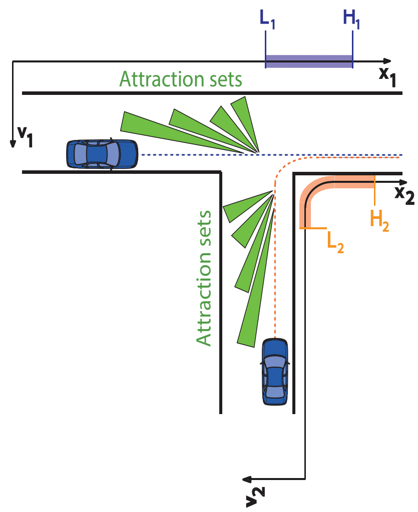


Fig. 1. Illustration of a traffic intersection scenario.

II. PROBLEM DESCRIPTION

Collision Avoidance (CA) in intersection crossing scenarios is a complex problem. A major challenge in the design of CA strategies is that safety involves ensuring collision avoidance not only prior to entering the intersection but within the intersection itself and, consequently, also upon exiting the intersection.

We assume a scenario including autonomous vehicles approaching a traffic junction, as shown in Figure 1. Note that the vehicles and all their variables are indexed to distinguish them. Moreover, each agent is assumed to be assigned to a driving task such that the following assumption holds.

Assumption 1: For each agent i, it is assumed that:

- the initial and final destinations corresponding to the driving tasks of all vehicles are known;
- the path leading a vehicle to its destination from its current position is assigned by its driving task and it is known;
- the vehicles do not change the assigned path once they enter the intersection.

It follows from the assumption that the generation of the driving paths is out of the scope of this paper. In particular, the vehicle steering is assumed to be controlled by a low level controller in order to follow a desired path. Similarly, the vehicles longitudinal velocities along the path are assumed to be regulated to a desired value by a low level controller. On the other hand, the desired longitudinal velocity for each vehicle is the result of the distributed agreement procedure, which is the topic of this article. Furthermore, we assume that at time zero all agents are before the intersection and that the intersection crossing problem has a feasible solution.

Under Assumption 1, the original problem can be considerably simplified. However, it is far from being trivial and, to the best of authors' knowledge, they are not too many efficient solutions to this problem.

Some preliminary results on the design of distributed agreement strategies are presented in this work. The formal stability analysis and robustness of the proposed approach with respect to uncertainties in sensing and communication, *e.g.*, noisy information and lost packets, will be considered in future research. The formalism used in the sequel is similar to the one presented in [8], which led to an useful and elegant collision model as a set of the vehicles configurations on their respective paths.

A. Vehicle modeling

Consider a set of N agents. Let $x_i = [p_i \; v_i]^T \in X_i$ denote the state of each vehicle $i \in N$, where $p_i \in P_i$ and $v_i \in V_i$ are its position and velocity, respectively. Note that $X_i = P_i \times V_i$ denote the state space, where P_i represents the set of all possible longitudinal positions along the path and V_i the set of all possible longitudinal velocities. Denote also k as the sampling instants and M the prediction horizon such that $k = \{1, \ldots, M\}$. Each agent is modeled as a discrete time double integrator. In a state-space form, the agent's dynamics can be described as follows:

$$x_i\,(k+1) = A\,x_i(k) + B\,u_i(k), \qquad (1a)$$
$$y_i(k) = C\,x_i(k), \qquad (1b)$$

where $u_i(k)$ denotes the control input, $x_i(k) = [p_i(k) \; v_i(k)]^T$ and

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Note that, as a part of the assigned driving task, each agent i has a given reference (desired) velocity denoted by $v_{di} \in V_i$, which is assumed to be constant over the all horizon. Define now:

$$\begin{cases} x_i^M = [x_i(0)^T, \cdots, x_i(M)^T] \\ y_i^M = [y_i(0)^T, \ldots, y_i(M)^T] \\ u_i^M = [u_i(0)^T, \cdots, u_i(M)^T] \\ v_{di}^M = [v_{di}(0), \cdots, v_{di}(M)] \end{cases} \qquad (2)$$

as the trajectory, the output, the input and the desired velocity sequence over a prediction horizon M, respectively. Furthermore, denote by:

$$\begin{cases} x = [x_i^T, \ldots, x_N^T]^T \\ y = [y_i^T, \ldots, y_N^T]^T \\ u = [u_i^T, \cdots, u_N^T]^T \\ v_d = [v_{di}, \ldots, v_{dN}]^T \end{cases} \qquad (3)$$

the state, the input, the output and the desired velocity vector of the global system composed of N agents, respectively.

B. Requirements and constraints

In realistic scenarios, several constraints such as bounded velocities, actuator limitations or time conditions such as minimum traveling time need to be handled. This section presents the considered constraints.

- ***Actuators limitations:*** To ensure that the control input is within the admissible actuator range (trust and brake), the following constraints are taken into account:

$$u_i^{min} \leq u_i(k) \leq u_i^{max}, \qquad \forall k \geq 0. \qquad (4)$$

For the sake of thoroughness of the notation, all inputs satisfying the previous condition are considered to belong to the set U_i.

- ***Position and velocity ranges:*** To ensure that vehicles operate within the feasible position range (set by the user or constrained by the environment) and within the desired velocity range (corresponding to, *e.g.*, the traffic rules), the vehicles' state variables are constrained such that:

$$\begin{bmatrix} 0 \\ v_i^{min} \end{bmatrix} \leq x_i(k) \leq \begin{bmatrix} p_i^{max} \\ v_i^{max} \end{bmatrix}, \qquad \forall k \geq 0, \quad (5)$$

where p_i^{max} is the maximum allowed position and v_i^{min} and v_i^{max} are constants defining the allowed velocity range.

- ***Safety:*** In this work, the proposed collision avoidance solution relies on the design a controller that prevents the system of reaching a given configuration. Let us define, for each agent i, the *critical set* $\mathbf{Cr}_i$ as the set of all displacements along the path leading to a potential collision. Thus, it follows:

$$\mathbf{Cr}_i = \{\ x_i \in X_i |\ p_i \in [L_i, H_i],\ \ \forall\ v_i \in V_i\ \}, \quad (6)$$

where $L_i < H_i$ are bounds on the position along the path of vehicle i defining, in a general way, the intersection. Note that these parameters depend on the physical system and its geometry. If at time k a point of the trajectory of agent i satisfies $p_i(k) < L_i$, we will generally say that agent i is "before" the critical set, while if a point satisfies $p_i(k) > H_i$ we will say that the agent is "after" the critical set. See Figure 2 for an illustration. Define now:

$$t_i^c = \{k |\ x_i(k) \in \mathbf{Cr_i},\ i \in \mathcal{N}\}, \quad (7)$$

as the set including all times instants for which the i-th agent's state lies within its *critical set*. Thus, it follows that the safety requirements with respect to collision avoidance can be written as:

$$t_i^c \cap t_j^c = \emptyset, \forall\, i, j \in N, j \neq i. \quad (8)$$

The previous condition can also be rewritten in the form of state constraints such as:

$$x_i(k) \notin \mathbf{Cr}_i, \forall k \in t_j^c, \forall i \in N, \forall j \neq i. \quad (9)$$

Note that the previous expression ensures that two agents are not within the intersection at the same times instants, therefore satisfying (8).

Remark 1: The used formalism and previous definitions are similar to the ones presented in [8] or more recently in [7]. However, it is important to point out that the nature of the set $\mathbf{Cr}_i$ is considerably different from the *bad set* defined in [8]. More precisely, [8] defines the *bad set* for the global state space of the system, therefore gathering information of all agents. Under such formalism, it follows that this set corresponds in fact to an effective collision. However, such formulation is not, in the authors' opinion, the more convenient to formulate the distributed collision avoidance problem considered in this paper. Here, the *critical set* $\mathbf{Cr}_i$ is defined by each agent's local information and can be represented in its individual state space. Thus, the *critical set* $\mathbf{Cr}_i$ only corresponds to a potential collision (a dangerous configuration) and a collision will only occur if at least the states of two agents lie within their respective critical set at a same time instant. Note, though, that an equivalent of [8]'s *bad set* can be easily found if one merges the information of the several $\mathbf{Cr}_i, \forall i \in N$.

C. Attraction sets and time to react

Crossing a traffic intersection can bee seen as a special case of a scheduling problem, where the access order to a shared resource needs to be computed. See Figure 1. In this type of scenarios, it is possible to find a set of initial conditions that will lead to an unavoidable collision. Obviously, such cases are out of the scope of collision avoidance strategies. However, in every other situation, one of the major challenges is to quantify the individual degree of freedom of each agent. For example, it is possible that agent i, once it detects a possible collision with another agent, is unable to control its future trajectory in such a way that it can influence the time instants $k \in t_i^c$ at which it will reach the intersection (represented here by its critical set). Based on such an argument, we are interested in determining if exists a trajectory leading a vehicle to the critical set in a finite number of steps, under any feasible control input. Thus, we introduce here the notion of *attraction set*, denoted by $\mathcal{A}_i, \forall i \in N$. Using reachability analysis tools, the set $\mathcal{A}_i$ is defined as the dual of one-step reachable set and is given as:

$$\begin{aligned} \mathcal{A}_i(\mathbf{Cr_i}) &= Pre(\mathbf{Cr_i}, U_i) \quad (10) \\ &= \{x_i(k)\ \in X_i : x_i(k+1) \in \mathbf{Cr_i}, \forall\, u_i \in U_i\}. \end{aligned}$$

In other words, the set $\mathcal{A}_i$ includes all possible state configurations that will lead the agent, unavoidably, to its *critical set* $\mathbf{Cr_i}$ in one step. The reader can refer to [2] for further details on reachability/controllability analysis. In a general way, define now:

$$\begin{aligned} \mathcal{A}_i(\mathcal{T}) &= Pre(\mathcal{T}, U_i) \\ &= \{x_i(k)\ \in X_i : x_i(k+1) \in \mathcal{T}, \forall\, u_i \in U_i\}. \end{aligned}$$

where $\mathcal{T}$ is usually referred to as the target set. Denote $\mathcal{A}_{i1} = \mathcal{A}_i(\mathbf{Cr_i})$. By performing backward sequential calculations, it is possible to compute the set $\mathcal{A}_i^z$ including all the *attraction sets* that will drive vehicle i to $\mathbf{Cr_i}$ in at most z steps such that:

$$\mathcal{A}_i^z = [\mathcal{A}_{i1}, \mathcal{A}_{i2}, \cdots, \mathcal{A}_{iz}], \quad (11)$$

where $\mathcal{A}_{ip} = \mathcal{A}_i(\ \mathcal{A}_{i(p-1)})$. Note that the scalar z is not a chosen parameter but it rather depends on the structure of the problem. In other words, z is the largest scalar before the backward reachability calculations result in an empty set.

Denote now $\tilde{k_j^c} = \min_{j \in N}\{t_j^c\}$. If no control input is applied ($u_i^M = 0$), the Time to React (Δ_i^{TR}) can be given as:

$$\Delta_i^{TR} = (\tilde{k_j^c} - z) - t_0, z \geq t, \forall i \in N.$$

In other words, the previous expression corresponds to the time interval between the time t_0 (when the coordination procedure starts) and the instant when agent i will enter the set $\mathcal{A}_i^z$. Thus,

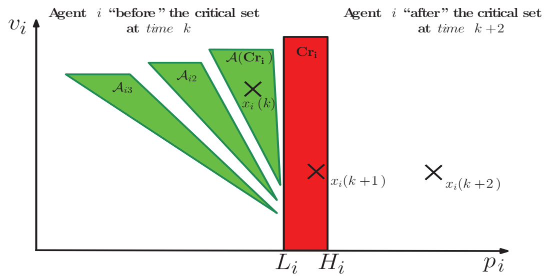


Fig. 2. Illustration of the attraction sets $\mathcal{A}_i^z$ and the critical set $\mathbf{Cr_i}$.

it follows that the agent's state will enter, unavoidably, into $\mathbf{Cr_i}$ in $\Delta_i^{attraction}$ steps such that:

$$\Delta_i^{attraction} = \Delta_i^{TR} + z.$$

For the sake of clearness, an illustrative schema is presented in Figure 2.

III. CONTROL STRATEGY

This section presents the formal control problem of collision avoidance at traffic intersections and is divided in two parts. In the first part, we formalize the centralized problem while on a second part a sequential decentralized approach and the necessary decision order criteria are presented. It is important to recall here that due to the nature of the problem, most of the centralized solutions for this problem consider mixed-integer constraints as, *e.g.*, in [20], [21], for which the solution computation effort scales exponentially with the time horizon and number of vehicles. Therefore, the objective of the proposed approach is to provide a lower complexity solution, even if the result is assumed to be suboptimal.

A. Centralized strategy

Assume that the state can be measured at every sampling instant k. The formal open-loop optimization problem to be solved in a centralized way can be written as:

$$\min_{u^M, x^M} \sum_{k=0}^{M-1} \|v(k) - v_d\|_Q^2 + \|u(k)\|_R^2, \tag{12}$$

subject to

(1), (4), (5) and (9), $\forall i$,

where $R \succ 0$ and $Q \succeq 0$ are weighting matrices of appropriate dimensions penalizing the control signal and any deviation of the agent's speed from the desired value, respectively. Moreover, $u^M = [u_i^{M^T}, \cdots, u_N^{M^T}]^T$ and $x = [x_i^{M^T}, \ldots, x_N^{M^T}]^T$ are the input and state sequence over the finite-time horizon M and their definition follows directly from equations (2) and (3).

Though the major challenge is to avoid collision among vehicles, one can consider several metrics to evaluate the performance of the system rather than the cost presented in (12). For example, one could aim to minimize the aggregate fuel consumption or alternatively the traveling time till destination.

B. Decentralized strategy

We propose in the sequel a decentralized solution where vehicles sequentially solve local optimization problems allowing them to cross, in a safe way, the intersection. Therefore, there are two important points to be described. The first corresponds to the decision order to be defined with respect to a given criteria. The second corresponds to the control problem itself, where collision avoidance is enforced through local state constraints at given time instants.

1) Decision order: It follows from (6) and (11) that both $\mathbf{Cr}_i$ and $\mathcal{A}_i^z$ are time invariant, therefore offering precious information regarding the individual degree of freedom of each agent. Such information will be used in our protocol to establish cooperation among agents. More precisely, we propose a *control scheduling* instead of a *crossing scheduling*, where vehicles solve their optimization problem sequentially. Note that such strategy does not guarantee the order in which agents should cross the intersection, as it would happen in a *crossing scheduling* problem. Thus, our approach can be seen as a priority rule, giving to the first agent in the sequence the advantage of keeping its desired motion profile, *i.e.*, the optimal solution that minimizes its own cost function. Then, this vehicle will broadcast the time stamps at which it expects to be within the intersection and that will be used to enforce constraint (9) for the second agent's optimization problems. This will proceed until all N vehicles have computed their solution.

One can reasonably argue that such a sequential policy offers a high advantage to the vehicles with high priority (*i.e.* the first few to decide) over the remaining agents, who might have to do much larger maneuvers to avoid collisions. In order to compensate such behavior, and also to introduce some logical fairness in the protocol, the *control priority* in this work is defined in a proportionally inverse way with respect to the values of $\Delta_i^{TR}, \forall i \in N$. In other words, priority will be given to the agent which lies closer to its attraction set, *i.e.*, the agent with the lowest Δ_i^{TR} value, then to the agent with the second smallest Δ_i^{TR} and so on. Such policy is motivated by the fact that the agent with the lower Δ_i^{TR} has, among all vehicles, the lowest individual degree of freedom. Thus, it is the authors' opinion that in this case such policy compensates the drawbacks of sequential approaches and seems to incorporate what would happen in real traffic situations.

2) Convex optimization problems: This section shows how the centralized problem can be decentralized using state constraints to enforce collision avoidance not only at the sampling instants but also at instants between them. The decentralized problems can be presented as follows:

$$\min_{u_i^M, x_i^M} \sum_{k=0}^{M-1} \|v_i(k) - v_{di}\|_{Q_i}^2 + \|u_i(k)\|_{Ri}^2, \tag{13}$$

subject to

(1), (4) and (5)

and $p_i(\bar{k}_j^c + 1) < L_i, \ \forall j \neq i.$

and

$$\min_{u_i^M, x_i^M} \sum_{k=0}^{M-1} \|v_i(k) - v_{di}\|_Q^2 + \|u_i(k)\|_R^2, \quad (14)$$

subject to

(1), (4) and (5)

and $p_i(\tilde{k}_j^c - 1) > H_i, \forall j \neq i.$

where $\bar{k}_j^c = \max_{j \in N}\{t_j^c\}$ and $\tilde{k}_j^c = \min_{j \in N}\{t_j^c\}, \forall j \neq i$. The solutions and costs of problems (13) and (14) are denoted $u_{i1}^*, \mathcal{C}_1$, u_{i2}^* and $\mathcal{C}_2$, respectively. The initial strategy is now divided in two sub problems. For a given set of collision times t_j^c, problem (13) provides the control sequence u_{i1}^* such that agent i is "before" its critical set when $k = \bar{k}_j^c + 1$, and will therefore cross the intersection after agent j. On the other hand, the second problem guarantees that agent i is "after" its critical at time $k = \tilde{k}_j^c + 1$, therefore guaranteeing that agent i has already crossed the intersection when agent j will cross it. Therefore, each agent is expected to solve two low complexity problems and chose afterwards the solution associated with the lowest cost. The structure of the proposed solution is defined in Algorithm 1.

Algorithm 1 Convex Sequential Optimization

Calculate $\forall i \in N$:
  $x_i^M = [x_i(k), \cdots, x_i(M)]^T, k = \{1, \cdots, M\}$,
  with $u_i^M = 0$.

if NO COLLISION ($t_i^c \cap t_j^c = \emptyset, \forall\, i, j \in N, j \neq i.$)
  return
end if

if COLLISION

- ○ Broadcast $\Delta_i^{TR}, \forall i \in N$
- ○ Locally compare all Δ_i^{TR} and establish control priority.
- ○ Perform sequentially:
  - For the agent with the smallest Δ_i^{TR} :
    - Calculate future trajectory with $u_i^M = 0$.
    - Broadcast all new t_i^c.
  - For the agent with the 2^{nd} smallest Δ_i^{TR} :
    - Solve problem (13) and compute u_{i1}^*.
    - Solve problem (14) and compute u_{i2}^*.
    - Compare the costs $\mathcal{C}_1$ and $\mathcal{C}_2$ associated with u_{i1}^* and u_{i2}^*.
    - Calculate future trajectory with $u_i^M = u_{i1}^*$ if $\mathcal{C}_1 < \mathcal{C}_2$ or with $u_i^M = u_{i2}^*$ otherwise.
    - Broadcast all new t_i^c.

    ⋮
  - For the agent with N^{th} smallest Δ_i^{TR} :
    - Solve problem (13) and compute u_{i1}^*.
    - Solve problem (14) and compute u_{i2}^*.
    - Calculate future trajectory with $u_i^M = u_{i1}^*$ if $\mathcal{C}_1 < \mathcal{C}_2$ or with $u_i^M = u_{i2}^*$ otherwise.
- ○ Apply control sequences.

end if

Remark 2: In the proposed approach, and mainly due to its formulation, the complexity of the convex sub-problems does not increase with respect to the number of agents, since collision avoidance is enforced by local state constraints at given time stamps.

The proposed algorithm is defined as an open-loop constrained optimal control solution, where the problem is solved in one calculation instant. However, the authors are now working on a receding horizon procedure using similar priority assessment policies.

	Initial position	L_i	H_i	t_i^c	Δ_i^{TR}
Agent 1	(10, 6.25)	55	65	8	6
Agent 2	(45, 4)	75	80	8	7
Agent 3	(7, 11.5)	102	112	9	8

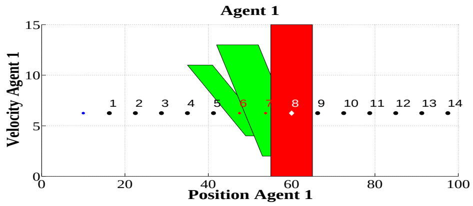
TABLE I

SIMULATION SETTINGS AND PARAMETERS

IV. SIMULATION RESULTS

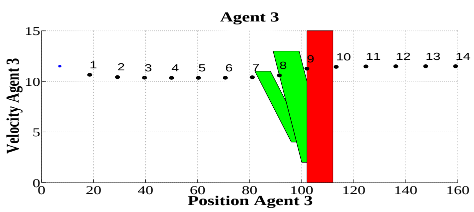
In this section, we present simulation results illustrating the performances of the proposed control strategy. We will consider a system of three vehicles ($N = 3$), mentioned throughout the sequel as vehicle/agent $1, 2$ and 3. The simulation settings are summarized in Table I and a graphical representation of the intersection scenario is presented in Figure 1. For each agent, the initial position is given by $x_i(0) = [p_i(0)\; v_{di}]^T$ and the finite time horizon has been set to $M = 14$. Moreover, agents are supposed to be identical with respect to the cost functions and control constraints such that $-2 < u_i < 2, \forall i \in N$. If no collision avoidance procedures are implemented, *i.e.*, if all agents respect their pre-defined trajectory, the collision between agent 1 and 2 will occur at $k = 8$, as stated in Table I. Moreover, accordingly to the scheduling criteria presented in previous sections and the different values of Δ_i^{TR}, the optimization order to this problem is: agent 1, 2 and 3.

Figure 3 presents the trajectories of the different vehicles accordingly to Algorithm 1. In all figures, the red set corresponds to the *critical set* $\mathbf{Cr}_i$, while the green sets represent $\mathcal{A}_i^z$. The black dots represent the state of each agent at different discrete time instants k, also presented in black. Note that $\mathbf{Cr}_i$ and $\mathcal{A}_i^z$ are time-invariant and represent a potential collision and the vehicle's ability to avoid it, respectively. Furthermore, recall that an effective collision occurs if at least the state of two agents lie within their respective critical set at a same time k. The reader can easily observe that collision avoidance constraints is now completely satisfied at all sampling times as well as between sampling instants. In particular, one can see that agent 1 keeps its desired speed, crossing the intersection at $7 < t < 9$. Based on these information, agent 2 solved problems (13) and (14) and implemented its optimal solution such that it crosses the intersection at $6 < t < 7$, *i.e.*, before agent 1. Finally, agent 3 crosses the intersection at $9 < t < 10$, *i.e*, after agent 1 and 2. Note that, as previously mentioned, the chosen *control scheduling* does not determine the real crossing order. In fact, the optimization problems are solved sequentially by agent 1, 2 and 3, while at the end the intersection is firstly crossed by agent 2, followed by agent 1 and 3.



(a)

(b)



(c)

F

Fig. 3. Simulation results for a three vehicle system accordingly to Algorithm 1. In all figures, the red set corresponds to the *critical set* $\mathbf{Cr}_i$, while the green sets represent $\mathcal{A}_i^z$. The black dots represent the state of each agent at different discrete time instants k, also presented in black.

These simulation results clearly show the efficiency and performance of the proposed approach. Furthermore, and in order to evaluate its practical efficiency, an experimental platform using real vehicles and real intersection scenarios is under preparation.

V. CONCLUSIONS AND PERSPECTIVES

In this article, we presented a cooperative driving strategy for intersection crossing, paying a special attention to how the degrees of freedom that each agent disposes to avoid a collision can be quantified. Firstly, we proposed a formalism allowing to represent, in the state space of each agent, not only a possible collision but also its individual ability to avoid possible conflicts. On a second step, we formally presented the centralized and a sequential decentralized problem. Using the new formalism, the order of the sequential approach is defined based on the attraction sets of each agent, which correspond to its degree of freedom in case of conflict resolution procedures. The proposed solution offers several advantages such as its low complexity and scalability. In fact, its complexity with respect to the number of agents remains constant since collision avoidance is enforced through local state constraints at given time instants. To complement these results that mainly consider feasibility conditions, future research should consider optimality arguments and formal analysis.

REFERENCES

[1] A. Bicchi and L. Pallottino. On optimal cooperative conflict resolution for air traffic management systems. *IEEE Transactions on Intelligent Transportation Systems*, 1(4):221–231, 2000.

[2] F. Borrelli, A. Bemporad, and M. Morari. Predictive control. *In preparation*.

[3] L. Conradt and T. Roper. Group decision-making in animals. *Nature*, 421:155 – 158, 2003.

[4] I. D. Couzin, J. Krause, N. R. Franks, and S. A. Levin. Effective leadership and decision-making in animal groups on the move. *Nature*, 433:513–516, 2005.

[5] D. V. Dimarogonas, S. G. Loizou, K. J. Kyriakopoulos, and M. Zavlanos. A feedback stabilization and collision avoidance scheme for multiple independent non-point agents. *Automatica*, 42:229–243, 2006.

[6] E. Frazzoli, Z.-H. Mao, J.-H. Oh, and E. Feron. Resolution of conflicts involving many aircraft via semidefinite programming. *Journal of Guidance, Control, and Dynamics*, 24(1):79–86, 2004.

[7] M. Hafner, D. Cunningham, L. Caminiti, and D. D. Vecchio. Cooperative collision avoidance at intersections: Algorithms and experiments. *Accepted to IEEE Trans. Intelligent Transportation Systems*, 2013.

[8] M. R. Hafner, D. Cunningham, L. Caminiti, and D. D. Vecchio. Automated vehicle-to-vehicle collision avoidance at intersections. In *Proceedeings of the ITS World Congress*, 2011.

[9] J. Hespanha, P. Naghshtabrizi, and Y. Xu. A survey of recent results in networked control systems. *Proceedings of IEEE*, 95(1):138 – 162, 2007.

[10] J. Huang and H.-S. Tan. Design and implementation of a cooperative collision warning system. In *IEEE Intelligent Transportation Systems Conference*, pages 1017–1022, 2006.

[11] H. Kowshik, D. Caveney, and P. Kumar. Provable systemwide safety in intelligent intersections. *IEEE Transactions on Vehicular Technology*, 60(3):804–818, 2011.

[12] J. Kuchar and L. Yang. A review of conflict detection and resolution modeling methods. *IEEE Transactions on Intelligent Transportation Systems*, 1(4):179–189, 2000.

[13] S. Martínez, J. Cortés, and F. Bullo. Motion coordination with distributed information. *IEEE Control Systems Magazine*, 27:75–88, 2007.

[14] A. Okubo. Dynamical aspects of animal grouping: Swarms, schools, flocks, and herds. *Advances in Biophysics*, 22(0):1 – 94, 1986.

[15] R. Olfati-Saber, J. Fax, and R. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215 –233, 2007.

[16] A. U. Raghunathan, V. Gopal, D. Subramanian, L. T. Biegler, and T. Samad. Dynamic optimization strategies for three-dimensional conflict resolution of multiple aircraft. *Journal of Guidance, Control, and Dynamics*, 27(4):586–594, 2004.

[17] W. Ren and R. W. Beard. *Distributed Consensus in Multi-vehicle Cooperative Control*. Communication and Control Engineering. Springer, 2007.

[18] W. Ren, R. W. Beard, and E. M. Atkins. A survey of consensus problems in multi-agent coordination. In *American Control Conference*, pages 1859–1864, 2005.

[19] W. Ren and Y. Cao. Overview of recent research in distributed multi-agent coordination. In *Distributed Coordination of Multi-agent Networks*, Communications and Control Engineering, pages 23–41. Springer London, 2011.

[20] A. Richards and J. How. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In *Proceedings of the American Control Conference*, volume 3, 2002.

[21] T. Schouwenaars, J. How, and E. Feron. Decentralized cooperative trajectory planning of multiple aircraft with hard safety guarantees. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, pages 2004–5141, 2004.

[22] F. Tedesco, D. Raimondo, A. Casavola, and J. Lygeros. Distributed collision avoidance for interacting vehicles: a command governor approach. In 2^{nd} *IFAC Workshop on Distributed Estimation and Control in Networked Systems*, 2010.

[23] C. Tomlin, J. Lygeros, and S. Sastry. A game theoretic approach to controller design for hybrid systems. *Proceedings of the IEEE*, 88(7):949–970, 2000.

[24] C. Tomlin, G. Pappas, and S. Sastry. Conflict resolution for air traffic management: a study in multiagent hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):509–521, 1998.