

Decentralized model predictive control for smooth coordination of automated vehicles at intersection

Xiangjun Qian¹, Jean Gregoire¹, Arnaud de La Fortelle^{1,2}, and Fabien Moutarde¹

Abstract—We consider the problem of coordinating a set of automated vehicles at an intersection with no traffic light. The priority-based coordination framework is adopted to separate the problem into a priority assignment problem and a vehicle control problem under fixed priorities. This framework ensures good properties like safety (collision-free trajectories, brake-safe control) and liveness (no gridlock). We propose a decentralized Model Predictive Control (MPC) approach where vehicles solve local optimization problems in parallel, ensuring them to cross the intersection smoothly. The proposed decentralized MPC scheme considers the requirements of efficiency, comfort and fuel economy and ensures the smooth behaviors of vehicles. Moreover, it maintains the system-wide safety property of the priority-based framework. Simulations are performed to illustrate the benefits of our approach.

I. INTRODUCTION

Currently, traffic lights are installed in many intersections to coordinate conflicting traffic flows. However, there is a rising concern on the efficiency and safety of these systems. Recent advances in embedded sensors, V2X communication and on-board computing have enabled the emergence of automated and cooperative vehicles. Taking advantage of automated driving, studies have been conducted to explore ideas of autonomous intersections without traffic lights, as briefly presented below.

Planning-based approaches [1]–[3] compute in the first phase collision-free trajectories for all vehicles (often in a centralized and sequential way); then in the second phase vehicles are controlled to follow the trajectories to cross the intersection (this control phase is decentralized). In [1], the optimal speed profiles for a two-vehicle intersection are analytically studied assuming simple vehicle behaviors, while the extension to a multi-vehicle intersection is subject to future work. In [2], constrained nonlinear optimization techniques are employed to plan trajectories for all vehicles entering the intersection. The control goal is to minimize the total length of overlapped trajectories. However, the complexity of the optimization problem renders the solution hard to obtain.

Though planning-based approaches may have good properties since trajectories can be optimized in advance, a major weakness lies in the difficulty to execute the planned

trajectories, when facing changing environments or control uncertainties. Unfortunately, the collision-free property of planning-based approaches essentially relies on the perfect control assumption. Failing to respect planned trajectories means to — if properly handled — trigger an emergency action such as a general stop (that could even be unsafe, there is usually no proof that a general sudden brake is collision-free, but practically with some safety margin it can be). And worse, nothing proves the state reached after the emergency action is not a gridlock.

To enable a quick response to changes and uncertainties, reactive approaches [4], [5] have been proposed. Instead of programming complete trajectories, vehicles calculate their current control decisions with respect to other vehicles' states and environmental information. In [4], every vehicle uses a navigation function to decide the current control input. The navigation function includes a collision avoidance term which enables a vehicle to respond to maneuvers of other vehicles. A major difficulty of reactive approaches lies in the deadlock avoidance: without global coordination, it is difficult to get a proof that deadlocks are avoided.

In our previous work [6]–[8], a priority-based coordination scheme was proposed that gives a balance between planning-based approaches and reactive approaches. The problem of coordinating multiple vehicles at intersection is separated into two parts: high-level planning of priorities and low-level reactive control of vehicles. The planning of priorities decides the relative orders of vehicles to cross the intersection (and so ensures collision-free maneuvers). It has been proven, under mild assumptions, that it is possible to ensure the intersection to be deadlock-free by selecting acyclic priority graphs [6]. Since a priority graph corresponds to a non-empty homotopy class of trajectories in the system configuration space, it is possible to build a control preserving this priority. Moreover it has been shown that the intersection satisfying *priority preserving* condition is brake-safe: an emergency braking maneuver is collision-free. To demonstrate the feasibility of the framework, a "bang-bang" priority-preserving control law has been proposed [6].

The "bang-bang" control law is reactive and simple to compute. However, it leads to non-smooth vehicle behaviors, and to unnecessary energy consumption. Moreover, the control law requires vehicles to maximally brake if priority violations are imminent; however we could anticipate the future violations and brake earlier. These drawbacks make the "bang-bang" control law inappropriate in the real world.

Some recent works [5], [9]–[11] adopt the Model Predictive Control (MPC) approach to coordinate vehicles at

¹Xiangjun Qian, Jean Gregoire, Arnaud de La Fortelle and Fabien Moutarde are with MINES ParisTech, PSL - Research University, Centre for Robotics, 60 Bd St Michel, 75006 Paris, France {xiangjun.qian, jean.gregoire, fabien.moutarde, arnaud.de.la.fortelle}@mines-paristech.fr

² Arnaud de La Fortelle is also with Inria Paris - Rocquencourt, RITS team, Domaine de Voluceau - Rocquencourt, B.P. 105 - 78153 Le Chesnay, France

the intersection. MPC optimizes a predefined cost function (usually considering efficiency, ecological and safety objectives) over a finite time horizon to produce a sequence of control inputs, of which the first step is implemented. To take into account environmental changes, the algorithm is repeated every timeslot in a receding horizon fashion. The application of MPC on the priority-based framework seems promising to overcome the drawbacks of the "bang-bang" law.

Contribution: In this paper, we propose a novel decentralized MPC control approach incorporating the priority-based coordination framework. Our innovation resides in two aspects

- Integration of MPC in a well designed framework with provable collision-free and deadlock-free properties.
- A "low-cost" cooperation strategy that allows vehicles to predict precisely the future states of conflicting vehicles.

Section II presents the system model and basic assumptions. Section III formulates the priority-based framework. Section IV presents the MPC based priority-preserving control law. Section V illustrate the benefits of our approach through simulations. Finally, section VI concludes the paper and discusses the perspectives .

II. SYSTEM MODEL

We consider the task of coordinating a collection N of vehicles to cross the intersection. Every vehicle $i \in N$ is constrained to forward-only motion along a fixed path (Figure 1a) $\gamma_i \subset \mathbb{R}^2$ and $x_i \in \mathbb{R}$ is its curvilinear coordinate along the path. Vehicle dynamics are simplified as a second order integrator. Given vehicle i , $v_i = \dot{x}_i$ is the speed of vehicle and $u_i = \ddot{x}_i$ the acceleration. We bound the speed and the acceleration: $v_i \in V_i = [0, \bar{v}_i]$ and $u_i \in U_i = [\underline{u}_i, \bar{u}_i]$. Let $s_i = (x_i, v_i)' \in S_i$ denote the state of a vehicle and $\mathbf{s} = \{s_i\}_{i \in N}$ denote the system state. We assume the control input is updated in discrete time τ :

$$\forall k \in \mathbb{N}, \forall t \in [k\tau, (k+1)\tau), u_i(t) \equiv u_i(k\tau) \quad (1)$$

With slight abuse of notation, we let $u_i(k)$, $v_i(k)$, $x_i(k)$ and $s_i(k)$ respectively represent the control, speed, position and state of vehicle i at time $k\tau$. The discrete-time state equation can then be given by

$$s_i(k+1) = As_i(k) + Bu_i(k) \quad (2)$$

where $A = \begin{pmatrix} 1 & \tau \\ 0 & 1 \end{pmatrix}$ and $B = \begin{pmatrix} \frac{1}{2}\tau^2 \\ \tau \end{pmatrix}$. We assume vehicles are equipped with sensors and communication devices that provides information of itself and other vehicles (absolute positions, velocities, accelerations, *etc.*). We assume two-way communication between vehicles can be established with small delay (much smaller than τ) through protocols like Dedicated Short-Range Communications [12]. We assume the initial vehicle configurations will not generate unavoidable collision.

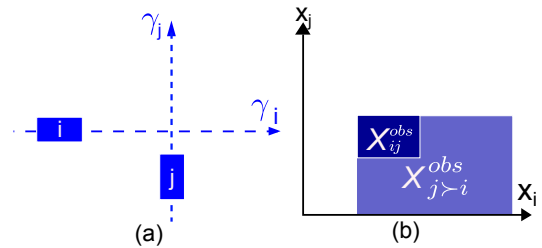


Fig. 1: The left drawing depicts an intersection with two automated vehicles to coordinate. The right drawing presents the obstacle region and the completed obstacle region of vehicles i and j .

III. PRIORITY-BASED COORDINATION FRAMEWORK

We adopt the priority-based framework [6] as the solution approach to the coordination problem. The general idea of the framework is to plan relative *priorities* between vehicles to cross the intersection and then control each vehicle to stay in the brake-safe sets with regards to prior vehicles. A vehicle i 's state is in the brake-safe set with regards to another vehicle j if an unpredictable event happens that makes vehicle j to maximally brake, vehicle i is still capable of braking without colliding with the first. The property of brake-safe set ensure the system to be robust to a wide class of uncertainties (emergency brake of other vehicles, *etc.*). In this section, we reformulate this framework, and use it as the foundation of our proposal.

A. Reformulation of the framework

We define the obstacle region $X_{ij}^{obs} \subset \mathbb{R}^2$ (Figure 1b) as the set of configurations (x_i, x_j) where i and j collide. We technically require X_{ij}^{obs} to be an open set. For every couple of vehicles with a non-empty obstacle region, one vehicle necessarily passes before or after the other one, which naturally leads to the notion of *priority*. We define $X_{j \succ i}^{obs} \equiv X_{ij}^{obs} + \mathbb{R}_+ \times \mathbb{R}_-$ (Figure 1b) as the set of bad configurations that may inevitably lead to the violation of the priority $j \succ i$, called completed obstacle region. Specially, Figure 2 illustrates the completed obstacle region of two vehicles on the same path. To describe the priorities between all vehicles, we can define an oriented priority graph G whose vertices are $V(G) := N$ and edges $(j, i) \in E(G)$. Each vertex represents a vehicle and the oriented edge between two nodes represents the priority relation between them.

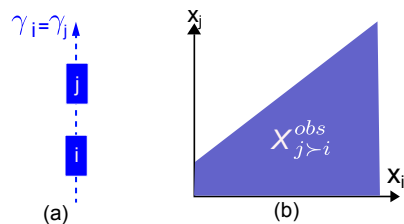


Fig. 2: Vehicle i and j are on the same path. Their completed obstacle region is illustrated in the right figure

The priority-based framework separates the coordination task into two parts:

- Design a feasible priority graph.
- Control the vehicles in a way that priorities are respected.

The assignment of priorities is itself a complex task, given potentially very high number of possible priority graphs ($2^{(N-1)N/2}$). Fortunately, simple heuristic algorithms are available [6] to compute a fairly efficient (in terms of delay) priority graph. In this paper, we assume the priority graph is given and we focus on the control of vehicles under fixed priorities.

Let $k \mapsto \Phi_i(k, s_i, u_i)$ denote the discrete-time state flow of the vehicle i starting from the initial state s_i and driven by the control signal u_i . We define a projection operator $\pi_x(s) = x$. We then define $\Phi_{x,i} = \pi_x(\Phi_i)$ that returns the evolution of the vehicle's position on the path. Let the current

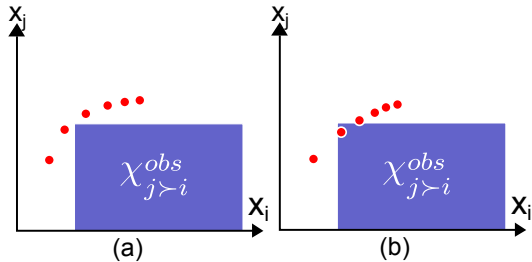


Fig. 3: The left drawing shows the brake trajectory ($\Phi_{x,i}(k, s_i, \underline{u}_i), \Phi_{x,j}(k, s_j, \underline{u}_j)$) of a brake-safe state. All configurations are outside the completed obstacle region. The right drawing shows a non brake-safe state where some configurations on the brake trajectory are inside the region.

time instant to be instant 0. For vehicle i and vehicle $j \succ i$, given the current state of vehicle j as s_j , the brake-safe set of vehicle i with respect to vehicle j can be given by a set-valued function $B_{j \succ i}(s_j)$ as

$$B_{j \succ i}(s_j) = \{s_i \in S_i \mid \forall k \geq 0, (\Phi_{x,i}(k, s_i, \underline{u}_i), \Phi_{x,j}(k, s_j, \underline{u}_j)) \notin X_{j \succ i}^{obs}\} \quad (3)$$

Figure 3 provides an illustration of the brake-safe state. Note that the brake-safe state is more strict than the *inevitable collision state* defined in [13] or the *escape set* defined in [14].

Remark 1: Strictly speaking, priority violations may occur between two consecutive time steps, as a side-effect of continuous system discretization. This problem can be avoided by adding proper margin on the completed obstacle region. The size of the margin is of the order of $O(\bar{v}_i \tau)$. The margin vanishes if $\tau \rightarrow 0$.

In order to respect priorities, a feasible control input of vehicle i at the current timeslot ($k = 0$) should guarantee the vehicle to stay in the brake-safe set at the next timeslot ($k = 1$). We refer to all feasible controls as the *priority-preserving* control inputs. Mathematically, the set of priority-

preserving inputs is given as a set-valued function

$$U_i^G(\mathbf{s}_{j \geq i}) = \{u_i \in U_i \mid \forall (j, i) \in E(G), \Phi_i(1, s_i, u_i) \in B_{j \succ i}(\Phi_j(1, s_j, \underline{u}_j))\} \quad (4)$$

where $\mathbf{s}_{j \geq i} = (s_j)_{j \succ i \mid j=i}$ includes the states of prior vehicles as well as the ego vehicle state.

We can prove that $U_i^G(\mathbf{s}_{j \geq i})$ is either an empty set, if there exists no control input allowing vehicle i to stay in the brake-safe set at the next time step; or it is a closed interval $[\underline{u}_i, \sup(U_i^G(\mathbf{s}_{j \geq i}))]$, where $\sup(U_i^G(\mathbf{s}_{j \geq i}))$ is the upper bound of the priority-preserving input. This property serves as an important preliminary for the proposed MPC approach.

B. A "bang-bang" priority-preserving control law

A simple priority-preserving control law is introduced in the previous work [6]. Let g_i^G denote the control law. For vehicle i , assuming that the system state at current timeslot is \mathbf{s} , we give g_i^G as :

$$g_i^G(\mathbf{s}) = \begin{cases} \bar{u}_i & \text{if } \bar{u}_i \in U_i^G(\mathbf{s}_{j \geq i}) \\ \underline{u}_i & \text{if } \bar{u}_i \notin U_i^G(\mathbf{s}_{j \geq i}) \end{cases} \quad (5)$$

At each time step, a vehicle i observes the system state, and maximally accelerates if maximal acceleration is priority-preserving. Otherwise it maximally brakes.

C. Problem under consideration and the proposed approach

The "bang-bang" control law proposed in section III-B demonstrates the reactivity of the priority-based framework. Vehicles base their next step control inputs on the current system state, allowing them to constantly react to the changing environment and avoid collisions. However, the "bang-bang" law produces non-smooth vehicle behaviors, which in turn leads to bad passenger experience and large fuel consumption.

The challenge is to find a control law that is not only priority-preserving and reactive, but also considers the practical needs of speed tracking efficiency, comfort and sustainability. Moreover, this control should still be decentralized, with each vehicle making its own control decision.

To deal with the problem, we propose a model predictive control approach that formulates our case to an optimal control cycle subject to system dynamics and constraints on system state and control input. In the rest of this paper, we describe our approach and present some preliminary simulation results.

IV. THE MODEL PREDICTIVE APPROACH

A. High-level view of the approach

We propose a decentralized solution where vehicles solve optimization problems locally, allowing them to safely cross the intersection. At the beginning of each timeslot, the automated vehicle performs the following three steps to decide the control input to be implemented:

- 1) The vehicle observes the current system state and predicts the evolution of the states of prior vehicles on a finite time horizon.

- 2) The vehicle calculates a sequence of control inputs by optimizing a predefined cost function under a series of constraints.
- 3) The vehicle only implements the first control input from the sequence calculated from step 2. The optimization process restarts at the next timeslot.

Step 1 and step 2 are two important points to be described in detail. For the sake of clarity, we firstly present the formulation of the optimization problem (step 2). In the second place, we come back to discuss the problem of predicting the future states of prior vehicles (step 1).

B. Optimization problem formulation

We consider a vehicle i , its initial state s at time $k = 0$, and the fixed priority graph G . Let K denote the length of prediction horizon. The local model predictive control problem can then be formulated as a constrained nonlinear optimization problem:

$$\min_{u_i} \mathcal{J}_i(s_i, u_i) = \min_{u_i} \sum_{k=0}^K \mathcal{L}_i(s_i(k), u_i(k)) \quad (6)$$

subject to

$$s_i(0) = s \quad (7)$$

$$s_i(k) \in S_i, u_i(k) \in U_i, k = 0, \dots, K \quad (8)$$

$$s_i(k+1) = As_i(k) + Bu_i(k), k = 0, \dots, K-1 \quad (9)$$

$$u_i(k) \in U_i^G(\mathbf{s}_{j \succ i}(k)), k = 0, \dots, K \quad (10)$$

where \mathcal{J}_i denotes the cost function to be minimized by selecting a proper sequence of control input $u_i(k), k = 0, \dots, K$. \mathcal{L}_i denotes the so-called running cost during the interval τ . We may select a quadratic running cost as

$$\mathcal{L}_i(k) = c_{i,1}(v_i^{target} - v_i(k))^2 + c_{i,2}u_i(k)^2 \quad (11)$$

where the first term is the efficiency cost (gap between the current speed and the target speed) and the second term penalizes the control signal. We may consider other metrics to evaluate the cost rather than the one proposed here. For example, one could directly aim to minimize the fuel consumption during the crossing. (7) and (8) respectively define the initial equality constraint and the boundary condition constraint. Equation (9) is the state transition constraint that describes the time-dependent evolution of the system. To ensure the control input to be always priority-preserving, we enforce the priority-preserving constraint at each time step $k = 0, \dots, K$ in (10).

Remark 2: In theory, it is not necessary to enforce the priority-preserving constraint on every data point in the prediction horizon, because only the constraint at the first time step is indispensable to guarantee the output to be priority-preserving. However, we opt to enforce constraints at the future time steps since it allows the vehicle to react earlier to possible priority violations in the future.

We are able to propose the following sufficient condition for the existence of solution of the optimization problem:

Theorem 1: The optimization problem (6)-(10) is feasible if the current vehicle state is brake-safe with regards to prior vehicles.

The proof is intuitive. A vehicle in brake-safe state ensures $U_i^G(\mathbf{s}_{j \succ i}(k))$ to be closed intervals, for all $k = 0 \dots K$, which in turn guarantees the solution space to be non-empty.

C. Predicting the future states of prior vehicles

In our proposed MPC approach, interactions among vehicles are fully encoded in the priority-preserving constraint (10): a vehicle should stay in the brake-safe state with regards to prior vehicles at any moment. This actually means that the vehicle should at least know the current states of prior vehicles (see Remark 2) to ensure its brake safety. In the second place, a vehicle may predict the future states of prior vehicles so that the ego vehicle control can be further optimized.

The knowledge of the current system state are assumed to be available (section II) through sensors and V2X communications. Thus the problem is reduced to predict the future states of prior vehicles.

In fact, assuming acyclic priorities, it is possible to obtain an "exact" prediction of prior vehicles' future states by sequentially solving local MPC problems. We may consider an example of three vehicles presented in Figure 4b. The priority relations are $1 \succ 2$, $1 \succ 3$ and $2 \succ 3$. At the beginning, we may easily solve the local MPC problem for vehicle 1 to obtain its states for the future K steps, since vehicle 1 has no prior vehicle. Assuming negligible calculation time and communication delay, the future states of vehicle 1 are then immediately transmitted to vehicle 2 and 3. In the sequel, vehicle 2 can then solve the optimization problem by enforcing the priority-preserving constraint (10) with respect to vehicle 1. The future states of vehicle 2 are then transmitted to vehicle 3. Vehicle 3 may finally calculate its trajectory for next K steps. In the end, the predictions are exact and all vehicles obtain their optimal trajectories.

We refer to this sequential scheme as MPC*. Such scheme is difficult to implement in real world, considering limited computation capacity and non-trivial communication delay.

To approximate MPC*, we may adopt a simple linear prediction scheme to estimate the future states. We assume that prior vehicles maintain constant velocities during the considered time horizon. Future states can then be calculated through system dynamics equation. This approach is of low complexity and requires no additional communication effort. We refer to this approach as MPC₀.

We argue that it is still possible to benefit from the calculations of other vehicles, even if the computation and communication delays are no longer negligible (while still smaller than τ). We observe that although vehicles re-calculate the control sequence at each time step, the difference of two sequences in two adjacent steps are not significant. Assuming the current timeslot is k_0 , we may then use the result of $k_0 - 1$ to predict the future states of prior vehicles. Considering

vehicle i , j and $j \succ i$, at timeslot k_0 , the observation of the state of vehicle j is s_j^{ob} . Vehicle j has shared the optimization outcome at time $k_0 - 1$ with vehicle i , denoted as $u_j^{pr}(k_0 + k)$, $k = -1, \dots, K - 1$. The estimation of vehicle j 's state \hat{s}_j over the prediction horizon $k = 0, \dots, K$ can be written as

$$\hat{s}_j(k_0 + k) = \begin{cases} s_j^{ob} & k = 0 \\ f(\hat{s}_j(k-1), u_j^{pr}(k_0 + k)) & k < K - 1 \\ f(\hat{s}_j(k-1), 0) & k = K \end{cases} \quad (12)$$

Three cases are considered in (12). Current state of vehicle j is simply estimated as the observation s_j^{ob} . The predicted control inputs are available for steps $k \leq K - 1$, thus the estimation of the future states until $K - 1$ is based on the information provided by j . Finally, for step K , the estimation is based on the assumption that vehicle j maintains constant speed.

We refer to the proposed approach as MPC₁. MPC₁ is a "low-cost" cooperation method since it only requires additional communication efforts.

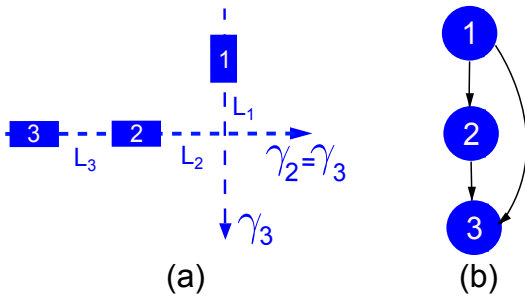


Fig. 4: Illustration of the intersection scenario (left) and the corresponding priority graph (right)

V. SIMULATION

A. Comparison of BB, MPC₀ and MPC₁

We compare BB, MPC₀ and MPC₁ through simulation. We first consider a system of three vehicles (Figure 4), labeled by numbers 1, 2 and 3. We determine the priorities as $1 \succ 2$, $1 \succ 3$ and $2 \succ 3$. The initial spatial localizations of vehicles are represented in Figure 4 ($L_1 = 40$ m $L_2 = 65$ m and $L_3 = 10$ m). Vehicles are supposed to be identical such that $V_i = [0, 8]$ m/s and $U_i = [-6, 3]$ (m/s²) for all $i \in \{1, 2, 3\}$. The initial velocities of three vehicles are set to 8 m/s. The parameters of the cost function are set to $v_i^{target} = 8$ m/s, $c_{i,1} = 1$ and $c_{i,2} = 6$ for all $i \in \{1, 2, 3\}$. The length of a timeslot is set to $\tau = 0.4$ s and the prediction horizon is set to 10.

We run simulations with three different control strategies: the "bang-bang" (BB) law, MPC₀ and MPC₁. To solve the constrained nonlinear optimization problems in MPC₀ and MPC₁, we adopt the sequential quadratic programming algorithm provided by Matlab *fmincon* function. We record the velocities and accelerations of vehicles until $t = 21.6$ s, when all vehicles have crossed the intersection.

As vehicle 1 has no prior vehicle, it can cross the intersection under the target speed profile. Vehicle 2 is forced to decelerate and then re-accelerate to give the right-of-way to vehicle 1. Vehicle 3 should also decelerate and re-accelerate to respect the priority of vehicle 2. Figure 5 illustrates the evolution of position, velocity and acceleration of vehicle 2 under three control strategies. We observe that MPC₀ and MPC₁ allow vehicle 2 to have smooth velocity curves. The proposed MPC strategies also enable vehicle 2 to anticipate the risk of collision and decelerate earlier. The profiles of vehicle 2 under MPC₀ and MPC₁ are identical. In fact, for vehicle 2, MPC₀ and MPC₁ are equivalent to MPC*, as vehicle 1 maintains constant speed. From Figure 6, we see that vehicle 3 has less velocity drop under MPC₁ than MPC₀. As MPC₁ provides a better estimation of vehicle 2's state evolution than MPC₀, vehicle 3 can brake in a less aggressive way while still ensure the brake safety. From Table I, we note that delays introduced by MPC approaches are slightly larger than the bang-bang control as a trade-off of getting smoother behavior.

We may further compare the fuel consumptions of the three approaches. We adopt the simplified fuel consumption model proposed in [15]. The instant fuel consumption rate is formulated as

$$f(v) = b_0 + b_1 v + b_2 v^2 + b_3 v^3 + [a(e_0 + e_1 v + e_2 v^2)]_{a>0} \quad (13)$$

where v and a are respectively the instant velocity and acceleration of vehicle. The term $a(e_0 + e_1 v + e_2 v^2)$ only takes effects if $a > 0$, otherwise it equals to zero. The parameters are given as $b_0 = 0.160$, $b_1 = 2.45 \times 10^{-2}$, $b_2 = -7.42 \times 10^{-4}$, $b_3 = 5.98 \times 10^{-5}$, $e_0 = 0.072$, $e_1 = 9.68 \times 10^{-2}$ and $e_2 = 1.08 \times 10^{-3}$.

	BB	MPC ₀	MPC ₁
Vehicle 1	0	0	0
Vehicle 2	6.4	6.9	6.9
Vehicle 3	2.3	4.0	3.8

TABLE I: Comparison of the delays (s) induced by deceleration and re-acceleration. The delay is calculated as the difference of traversing time under maximal velocity and actual traversing time. Delays introduced by MPC approaches are slightly larger

	BB	MPC ₀	MPC ₁
Vehicle 1	7.3	7.3	7.3
Vehicle 2	10.9	9.7	9.7
Vehicle 3	10.8	9.7	9.3

TABLE II: Comparison of the fuel consumptions. MPC approaches are less consuming than BB control thanks to vehicles' smooth behaviors. For vehicle 3, MPC₁ outperforms MPC₀ thanks to more precise estimation of prior vehicles' future states.

We record the aggregated fuel consumptions (milliliters) at 21.6 s for the three approaches, as presented in Table II. We observe that the proposed MPC strategies are more energy-saving than the "bang-bang" law thanks to their smooth

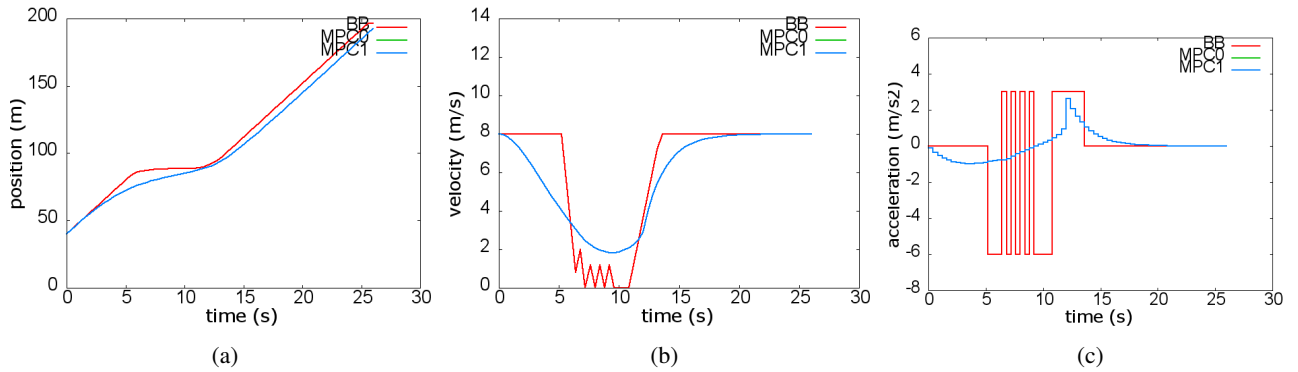


Fig. 5: Vehicle 2: position (left), velocity (middle) and acceleration (right) profiles under three different control strategies. MPC schemes anticipate possible priority violations and decelerate earlier. MPC schemes have smooth profiles. MPC₀ and MPC₁ are equivalent to MPC* as vehicle 1 maintains constant speed.

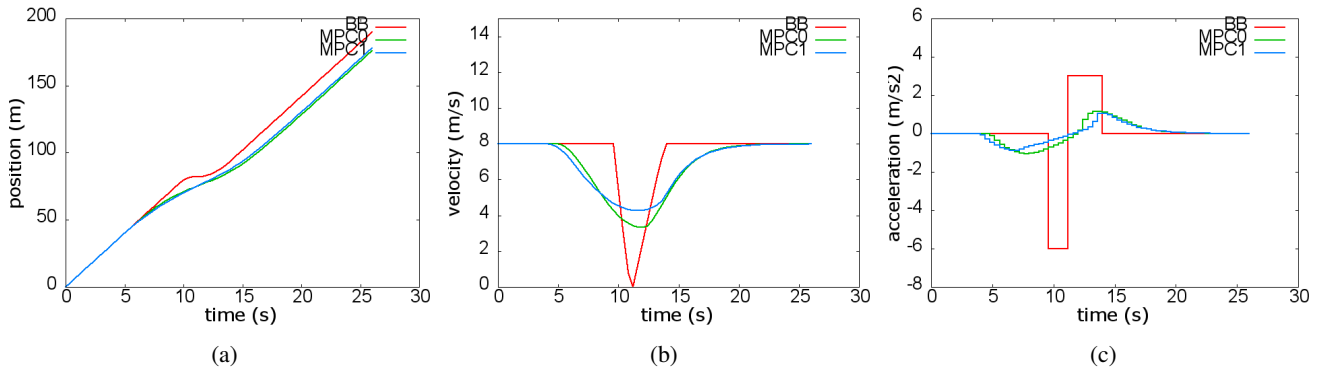


Fig. 6: Vehicle 3: position (left), velocity (middle) and acceleration (right) profiles under three different control strategies. MPC₁ leads to less velocity drop than MPC₀ thanks to more precise prediction of prior vehicles' future states.

behaviors (around 10 %). Moreover, with respect to vehicle 3, MPC₁ consumes 4% less fuel than MPC₀ during the entire horizon thanks to the more precise estimation of prior vehicles' future states.

With the analysis above, we conclude that the proposed decentralized MPC approach allows smooth coordination of multiple vehicles at intersection. Preliminary results show that MPC₁ exhibits better performance than MPC₀ in terms of fuel consumption.

B. Illustration of Emergency Braking

We use two scenarios of emergency braking to demonstrate the resilience property of MPC₀ (the result of MPC₁ is similar). We consider two vehicles i and j such that $j > i$. The first scenario is a car-following case with the initial gap being 10 meters. The second scenario is an intersection scenario where two vehicles starting from position zero approach an intersecting region located at $(50, 50)$. In both cases, the leading vehicle j suddenly brakes with maximal brake command at time $t = 4.0$ s and then re-accelerate to the target speed. Figure 7 and Figure 8 are respectively system trajectories in configuration space for two cases. The moment that vehicle j starts to brake is marked by a red arrow. We see that trajectories do not intersect with the completed obstacle regions.

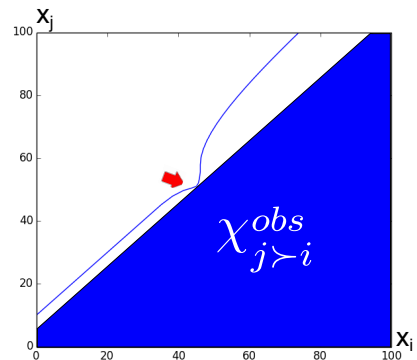


Fig. 7: System trajectory of the car-following scenario after an emergency braking of vehicle j . The moment that vehicle j starts to brake is marked by a red arrow.

VI. CONCLUSIONS AND PERSPECTIVES

We present a novel decentralized MPC approach that allows smooth coordination of automated vehicles at intersection. More precisely, we adopt the priority-based coordination framework and separate the coordination problem into a priority assignment problem and a vehicle control problem. Priorities are fixed and the decentralized MPC scheme

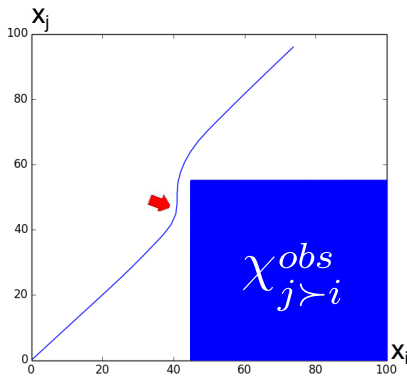


Fig. 8: System trajectory of the intersection scenario after an emergency braking of vehicle j . The moment that vehicle j starts to brake is marked by a red arrow.

permits vehicles to smoothly cross the intersection without violating priorities. The proposed scheme is efficient and reactive, ensuring the system to be resilient to unexpected events (emergency braking).

We discuss two different methods (MPC_0 and MPC_1) that allows a vehicle to predict the behaviors of prior vehicles. We demonstrate that MPC_1 performs better in terms of fuel consumption. MPC_1 requires only local information (from prior vehicles).

An important unsolved problem is the stability of the proposed MPC approaches. Even though simulations suggest the system to be stable if a long enough prediction horizon is used, it is necessary to obtain a formal stability guarantee before implementing in vehicles. In the future, approaches surveyed in [16] can be adapted to achieve this goal.

The vehicle dynamic model adopted in this paper is in its simplest form. In the future, we plan to consider more realistic vehicle dynamics (bicycle model, *etc.*). We should also carefully consider uncertainties in control and perception. We will further investigate the impact of communication delay and loss to our approach. Finally, the work will be implemented both in a realistic simulation platform and in real vehicles as a part of the European project AutoNET2030.

ACKNOWLEDGEMENT

This work is supported by the European Commission under AutoNet2030, a EU FP7 project (Grand Agreement no 610542). The authors would like to thank all partners within AutoNET2030 for their cooperation and valuable contribution. We also thank Silvere Bonnabel and Gabriel Rodrigues de Campos for their insightful comments. The authors thank reviewers for their valuable comments.

REFERENCES

- [1] A. Charalampidis and D. Gillet, "Speed profile optimization for vehicles crossing an intersection under a safety constraint," in *European Control Conference (ECC)*, 2014, pp. 2894–2901, June 2014.
- [2] J. Lee and B. Park, "Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, pp. 81–90, March 2012.

- [3] A. de La Fortelle, "Analysis of reservation algorithms for cooperative planning at intersections," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pp. 445–449, Sept 2010.
- [4] L. Makarem, M. H. Pham, A.-G. Dumont, and D. Gillet, "Micro-simulation Modeling of Coordination of Automated Guided Vehicles at Intersection," in *Transportation research board 91st annual meeting*, 2012.
- [5] L. Makarem and D. Gillet, "Model predictive coordination of autonomous vehicles crossing intersections," in *Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on*, pp. 1799–1804, Oct 2013.
- [6] J. Gregoire, S. Bonnabel, and A. de La Fortelle, "Priority-based coordination of robots," *CoRR*, vol. abs/1306.0785, 2013.
- [7] J. Gregoire, S. Bonnabel, and A. de La Fortelle, "Priority-based intersection management with kinodynamic constraints," in *European Control Conference (ECC)*, 2014, pp. 2902–2907, June 2014.
- [8] X. Qian, J. Gregoire, F. Moutarde, and A. de La Fortelle, "Priority based coordination of autonomous and legacy vehicles at intersection," in *Intelligent Transportation Systems (ITSC), 2014 17th International IEEE Conference on*, 2014.
- [9] M. Kamal, J. Imura, T. Hayakawa, A. Ohata, and K. Aihara, "A vehicle-intersection coordination scheme for smooth flows of traffic without using traffic lights," *Intelligent Transportation Systems, IEEE Transactions on*, vol. PP, no. 99, pp. 1–12, 2014.
- [10] K.-D. Kim and P. Kumar, "An MPC-based approach to provable system-wide safety and liveness of autonomous ground traffic," *Automatic Control, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2014.
- [11] G. R. Campos, P. Falcone, H. Wymeersch, R. Hult, and J. Sjoberg, "Cooperative receding horizon conflict resolution at traffic intersections," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pp. 2932–2937, Dec 2014.
- [12] P. Papadimitratos, A. de La Fortelle, K. Evenssen, R. Brignolo, and S. Cosenza, "Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation," *Communications Magazine, IEEE*, vol. 47, pp. 84–95, November 2009.
- [13] T. Fraichard and H. Asama, "Inevitable collision states - a step towards safer robots?," *Advanced Robotics*, vol. 18, pp. 1001–1024, 2004.
- [14] D. D. Vecchio, M. Malisoff, and R. Verma, "A separation principle for a class of hybrid automata on a partial order," in *American Control Conference*, pp. 3638–3643, 2009.
- [15] M. Kamal, M. Mukai, J. Murata, and T. Kawabe, "Ecological vehicle control on roads with up-down slopes," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, pp. 783–794, Sept 2011.
- [16] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.