

# Roundabout Crossing With Interval Occupancy and Virtual Instances of Road Users

Stefano Masi<sup>1</sup>, Associate Member, IEEE, Philippe Xu<sup>1</sup>, Associate Member, IEEE,  
and Philippe Bonnifait<sup>1</sup>, Member, IEEE

**Abstract**—Although autonomous vehicle technology has evolved significantly in recent years, the navigation of self-driving vehicles in complex scenarios is still an open issue. One of the major challenges in these conditions is safe navigation on roads open to public traffic. The main issue is the interaction of the autonomous vehicle with regular traffic, as the behaviors and intentions of human-driven vehicles are hard to predict and understand. In this paper we propose a strategy to allow an autonomous vehicle to safely cross a multi-lane roundabout. Our approach uses a High-Definition (HD) map to predict at lane level the future situation, harnessing the concept of virtual instances of road users, which is a key concept in anticipating the situation in a roundabout that can be represented by a navigation graph with loops. This paper presents a methodology that uses intervals representing road occupancy by vehicles, with the road being widened to reflect uncertainties in localization. Our method safely avoids collisions and guarantees that no priority constraints are violated during the insertion maneuver. Moreover, the method does not provide an overly cautious insertion policy, *i.e.*, an autonomous vehicle does not wait for a long time before the insertion. The performance of our strategy was evaluated using the SUMO simulation framework. To better evaluate the complexity of the simulation scenario, a highly interactive vehicle flow was generated using real dynamic traffic data from the INTERACTION dataset. We report real tests carried out with an experimental self-driving vehicle on a test circuit. Our results show that this approach is easy to integrate into an embedded system and that it allows roundabouts to be crossed with a high level of safety.

**Index Terms**—Intelligent vehicles, autonomous driving.

## I. INTRODUCTION

NOWADAYS, Autonomous Driving (AD) vehicle technology is a very active field of research. In the near future AD vehicles are expected to co-exist with manually driven (MD) vehicles and other road users including bicycles and motorcycles. In this heterogeneous scenario, safe navigation has to be guaranteed in complex dynamic environments such as in intersections, lane merging and roundabouts, where the risk of accident is one of the highest on public roads.

Manuscript received April 29, 2020; revised September 21, 2020; accepted November 9, 2020. This work was supported in part by the Equipex ROBOTEX under Grant ANR-10-EQPX-44-01 and in part by the LabEx Maitrise des Systèmes de Systèmes Technologiques (MS2T) under Grant ANR-11-IDEX-0004-02. The Associate Editor for this article was L. Li. (Corresponding author: Stefano Masi.)

The authors are with the Heudiasyc UMR 7253, CNRS, Université de Technologie de Compiègne, 60200 Compiègne, France (e-mail: stefano.masi@hds.utc.fr).

Digital Object Identifier 10.1109/TITS.2020.3042870

1524-9050 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.  
See <https://www.ieee.org/publications/rights/index.html> for more information.

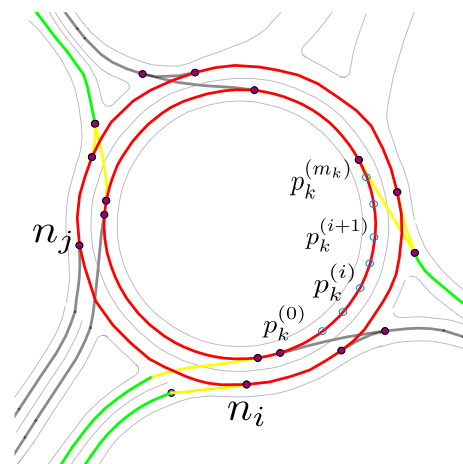


Fig. 1. A roundabout with its HD map representation. The decision zones are green, the transition zones are yellow, and the ring zone is red. The roundabout exits are blue. For a given link, the nodes and shape points are shown.

One way to improve this co-existence is to make use of V2V communications to share the intentions of the AD vehicle with other traffic participants. Other vehicles are thus able to build driving plans that explicitly consider the future behavior of the AD vehicle. However, this method assumes that all the road users are equipped with V2X devices, which is unrealistic in a short-term horizon.

The literature includes a number of works addressing intersection crossing with only AD vehicles (see [7], [22], [27], [32]) and with priority constraints [8]. Some approaches solve the intersection crossing problem with optimization [27] or with model predictive control [23], [36]. Another approach involves computing and avoiding situations that would necessarily lead to a collision even with emergency braking [33]. Finally, the Autonomous Intersection Management Protocol (AIM) [6], [30] and virtual platoon methods [9], [26] have shown interesting properties.

In a mixed traffic environment with AD and MD vehicles (in this paper an MD vehicle can be a car or any road user in the drivable space), avoidance sets can be computed with reachability theory [11], [21], [39]. The authors in [28], [29] implemented a collaborative motion planning algorithm that cooperates with MD vehicles. Other works such as [42] are based on worst-case MD vehicle behavior (as an example, accelerating when approaching an intersection is considered to

be the least desirable). In [14], the behavior of an MD vehicle is modeled with a mathematical representation of human driving. Some tentacles-based motion planning techniques can also be found in the literature [37]. Such methods are often based on occupancy grids to estimate the free space in the environment [2], sometimes in combination with HD maps [15].

In order to better understand vehicle behaviors in complex scenarios, we looked at the principal datasets in the literature containing traffic flow information for autonomous driving navigation tasks (collision avoidance, path planning, etc.). These include Common Road [1], ACFR [43] or INTERACTION [41]. For this study we chose to focus our attention on INTERACTION, because it provides highly dynamic traffic flow data in a number of road scenarios, including where there are several roundabouts.

The roundabout crossing problem has received little attention in the literature. In [3] the authors proposed an algorithm for solving the roundabout crossing problem using game theory, while Desraj *et al.* [10] considered partial order techniques to develop a strategy for collision avoidance between two vehicles on a roundabout. Other works, namely [34] and [35], have focused more narrowly on the control of self-driving vehicles. An alternative approach that involves an extra infrastructure layer to implement a time slot scheduling protocol can be found in [17]. Although the authors proposed interesting valid theoretical solutions, they obtained their experimental results only from test-bed scenarios with either simulated vehicles or simplified roundabout models. Moreover, their experiments involved a limited traffic flow (only one adversarial vehicle), which is far from being a full-scale experiment. If we look carefully at roundabouts, we see some similarities with classical road merging and intersections. In particular, during the insertion maneuver, a roundabout is quite similar to a road merging where vehicles on the main road have higher priority. There are, however, fundamental differences. First, in a roundabout the available space is limited w.r.t. highway merging, and consequently there is less time to plan a driving maneuver. Second, traffic behavior in the roundabout ring is heterogeneous and hard to predict. This aspect is more noticeable where the roundabout has more than one lane. In particular, if no lane markings are present, it is not easy to distinguish between inner and outer lanes. As a consequence, MD vehicles tend to have an irregular behavior, making the navigation maneuver in the roundabout challenging. Some examples of this behavior can be found in the roundabout recordings in the INTERACTION dataset [41]. This is different from an intersection (T-intersection or lane merging), where in general MD vehicles choose their lane before crossing the intersection or the merging. Third, on a roundabout it can be difficult to predict which exit a vehicle will take. To deal with this problem it is appropriate to consider all possibilities by manipulating several virtual instances of the vehicles.

In [24] we showed that using virtual vehicles along the lanes of an HD map is a good way of predicting the dynamic situation in a roundabout so as to control the longitudinal behavior of an AD vehicle. In [25] we improved and extended

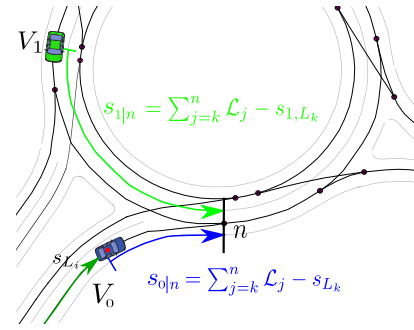


Fig. 2. The curvilinear abscissa w.r.t. a given link and the relative distance computation to the intersection point, as explained in section II. The blue arrow pointing towards the intersection point indicates the sign of the distance.

this method to a more complex case, i.e., a two-lane roundabout. The aim of the present paper is to provide an extended formalization and an experimental evaluation to show the feasibility of our approach in a real traffic scenario, and to address some issues that arise when virtual vehicle techniques are applied to roundabouts. The contributions of this work are as follows:

- A safe, priority-preserving and not overly cautious decision method for an AD vehicle crossing a two-lane roundabout among an MD vehicle flow;
- Extending the curvilinear coordinates virtual platooning to an interval-based curvilinear formalism to include both vehicle sizes and uncertainties in vehicle positions in the navigation algorithm;
- Using the virtual vehicle concept to handle multi-lane roundabout navigation and the prediction of vehicles' unknown intentions, and applying virtual vehicle methods to roundabout cycles;
- Generating an interactive, naturalistic and adversarial vehicle flow using real data from the INTERACTION dataset, to be used as a realistic benchmark for navigation algorithms in a SUMO-based simulation environment;
- A practical evaluation of our algorithm in terms of the safety and fluidity of the insertion, with realistic simulations and real experiments.

The paper is organized as follows. In the next section we introduce curvilinear formalism along with HD maps. Then, in section III we explain in detail the concept of virtual instances and how it can be applied to roundabout use. In section IV we present our roundabout crossing algorithm. Simulation results with a vehicle flow using real traffic data are presented in section V, followed by real experimental results in section VI.

## II. INTERVAL-BASED CURVILINEAR COORDINATES ALONG HD MAPS

When multiple vehicles share the driving space, it is important to properly represent their spatial positions with respect to each other. Euclidean coordinates are often not ideal for representing spatial relationships between road users. Curvilinear coordinates, on the other hand, are able to effectively encapsulate lane-level interactions between vehicles, for example in terms of conflicting trajectories w.r.t. the AD vehicle navigation corridor [31], [42].

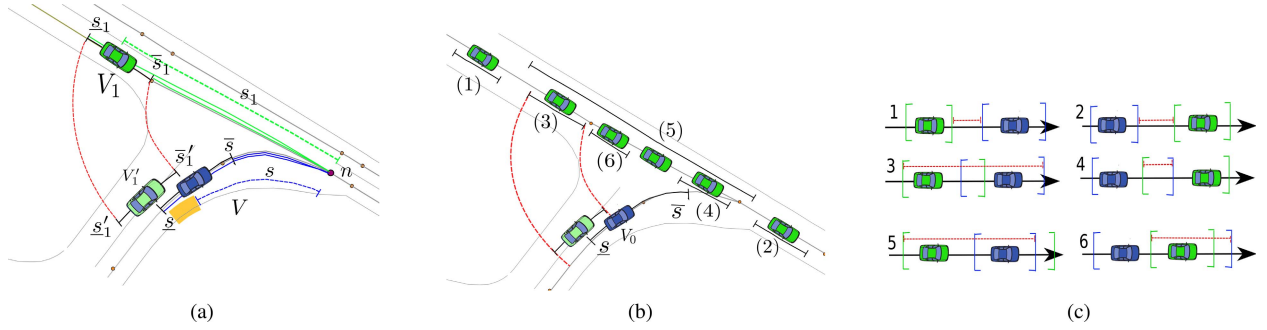


Fig. 3. (3a) Classical virtual platooning (dashed) and its extension to intervals (solid).  $V$  is the ego-vehicle and  $V_1$  is a virtual instance of  $V_1$ . (3b) Situation with 6 other vehicles (all the possible relative locations) Occupancy is projected onto the road map. (3c) Interval overlapping corresponding to Fig. 3(b).

The principle of curvilinear coordinates is to map a Euclidean pose (position and heading angle) onto a curvilinear pose with respect to a geometrical curve. These curves can be stored in a map in a variety of formats including clothoids, splines, or polylines, and typically represent vehicles' nominal paths. In many driving contexts, the center line of a lane is a sufficient approximation of the path that the vehicles follow. In the particular cases of overtaking and lane change, the AD vehicle can decide to replace its reference path by a path computed by a planner. Nowadays there are a number of open source navigation maps (*e.g.*, OpenStreetMap). However, for AD navigation tasks, an HD map, with a higher level of detail and better accuracy, is needed.

The present work adopts a polylines formalism to represent the roads. In particular, we define the road elements in a map as follows:

- Node: a set  $N$  of geo-referenced 2D points used to mark the start and the end points of part of a lane, in particular where two lanes split, merge or cross.
- Link: the portion of the lane between two nodes (the starting and ending nodes) that define the flow direction in the lane. The geometry of this part is represented as a polyline, which is a sequence of line segments. A link  $L_k$  composed of  $m_k$  line segments is described by  $m_k + 1$  points

$$L_k = (p_k^{(0)}, p_k^{(1)}, \dots, p_k^{(m_k-1)}, p_k^{(m_k)}), \quad (1)$$

in which the first and last points are nodes, *i.e.*,  $p_k^{(0)}, p_k^{(m_k)} \in N$ . It is important to note that no crossing can occur within a link, but only at the starting or ending nodes.

- Shape points: the 2D points used to model the geometrical shape of the lane (*i.e.*, the shape of the link) are called shape points. A node is also considered to be a shape point.
- Line segment: a line segment

$$l_k^{(i)} = (p_k^{(i)}, p_k^{(i+1)}), \quad (2)$$

attached to a link  $L_k$  is composed of two consecutive shape points, and its length is defined as

$$\ell_k^{(i)} = \|p_k^{(i+1)} - p_k^{(i)}\|. \quad (3)$$

A link  $L_k$  can be represented equivalently as an ordered sequence of  $m_k$  segments  $L_k = (l_k^{(0)}, l_k^{(1)}, \dots, l_k^{(m_k-1)})$ . The length of a link  $L_k$  is denoted

$$\mathcal{L}_k = \sum_{i=0}^{m_k-1} \ell_k^{(i)}. \quad (4)$$

To better illustrate this concept, Figure 1 shows the representation in terms of nodes and shape points for a given link. Another commonly used map representation is the *lanelet* representation [4], which uses the left and right bounds of the lane. In this case, a conversion into a center line representation is required beforehand, as in [18].

The first step when using a curvilinear representation is to identify the lane in which vehicles are traveling. For an AD vehicle, it is simply given by the path planning module in the form of a list  $P=(L_1, L_2, \dots)$  of links for the vehicle to follow (Figure 4). In contrast, the driving lanes for MD vehicles need to be estimated by a map-matching procedure. It is well known that map-matching ambiguities may arise when a road splits or when the vehicle changes lanes. Our preferred map-matching process is the method described in [24], where multiple candidate lanes can be simultaneously occupied by creating virtual instances of the MD vehicle, one for each lane that may potentially be occupied. This technique is detailed in section III-A.

The curvilinear abscissa  $s_{L_k}$  of a vehicle located on the link  $L_k$  and map-matched to its  $i$ -th segment  $l_k^{(i)}$  is computed w.r.t. the starting node  $p_k^{(0)}$  of  $L_k$  as follows:

$$s_{L_k} = \sum_{j=0}^{i-1} \ell_k^{(j)} + \lambda \ell_k^{(i)}, \quad (5)$$

where  $\lambda \in [0, 1]$  is a parameter to model the position along the map-matched segment  $l_k^{(i)}$ . Below,  $s_{L_k}$  will be denoted simply as  $s$  when there is no ambiguity. A practical computation of the curvilinear coordinates can be found in [16].

This representation is particularly useful in virtual platooning for safely crossing an intersection with a vehicle driving on another road [26]. Given the path of the vehicle  $P = (L_k, L_{k+1}, \dots)$ , the curvilinear abscissa from the path to any node, for example the ending node of the link  $L_n$ ,

is computed as

$$s_n = \sum_{j=k}^n \mathcal{L}_j - s_{L_k}. \quad (6)$$

Equation (5) computes the curvilinear abscissa of a vehicle measured w.r.t. the start node of the current link  $L_k$ , while in equation (6) the value  $s_n$  is measured from the ending node  $n$  of link  $L_n$  back to  $s_{L_k}$ . Note that  $s_n$  is negative and increases as  $s_{L_k}$  approaches the node  $n$ . Let us consider two vehicles  $V_i$  and  $V_j$  with paths  $P_i$  and  $P_j$ , respectively. If the two paths intersect, the notion of curvilinear abscissa can be used to compute the relative virtual gap between the two vehicles. Suppose that  $P_i$  and  $P_j$  intersect at a given node  $n$ , and let  $s_{i|n}$  and  $s_{j|n}$  be the curvilinear abscissa of  $V_i$  and  $V_j$  w.r.t. this node  $n$  as defined in equation (6). We define the virtual curvilinear signed inter-distance between  $V_i$  and  $V_j$  as follows:

$$d_{i,j|n} = -d_{j,i|n} = s_{j|n} - s_{i|n}. \quad (7)$$

Figure 2 illustrates this concept. Note that  $d_{i,j|n}$  is a signed inter-distance, where  $d_{i,j|n} > 0$  means that  $V_i$  is closer to  $n$  than  $V_j$ , i.e.,  $V_i$  is virtually ahead of  $V_j$ . Moreover, Figure 2 illustrates the computation of the curvilinear abscissa and distances.

However, in the real world a single curvilinear point is not sufficient to represent the physical occupancy by a vehicle. Instead of using curvilinear coordinates, we propose replacing them with curvilinear intervals to better represent occupancy by objects. Let us first generalize the virtual curvilinear inter-distance of equation (7) to the case of the interval curvilinear abscissa. For an interval  $[\underline{s}_i, \bar{s}_i]$  we define  $[\underline{s}_{i|n}, \bar{s}_{i|n}]$ , corresponding to the lower and upper curvilinear abscissa of the vehicle w.r.t. a given node  $n$  using equation (6). As stated previously, these values are negative and increase as the vehicle approaches the node  $n$ . We define the interval-based virtual curvilinear signed inter-distance  $d_{i,j|n}^*$  between  $V_i$  and  $V_j$ , whose paths intersect at a node  $n$ , as the signed distance between the front of  $V_j$  and the back of  $V_i$ :

$$d_{i,j|n}^* = \bar{s}_{j|n} - \underline{s}_{i|n}. \quad (8)$$

Therefore,  $d_{i,j|n}^* < 0$  means that w.r.t. node  $n$ ,  $V_i$  is virtually ahead of  $V_j$ , i.e., the vehicles are in the first configuration shown in Figure 3c. It is important to note that  $d_{i,j|n}^* > 0$  does not mean that  $V_j$  is ahead of  $V_i$ . Indeed, any of the other configurations in Figure 3c would lead to this case. Unlike where a single curvilinear abscissa is used, we have  $d_{i,j|n}^* \neq -d_{j,i|n}^*$ , i.e., this equation is not symmetric. To better illustrate the concept, Figures 3a, 3b and 3c show the case of a simple T intersection. For simplicity,  $d_{i,j|n}^*$  will from here on be denoted as  $d_{i,j}^*$  where there is no ambiguity.

### III. CYBER-PHYSICAL ROUNDABOUT MODEL

#### A. Unknown Intentions of MD Vehicles and Graph Cycles

In order to encompass all the possible behaviors of an MD vehicle, we consider that a single MD vehicle can be represented by a number of different virtual instances.

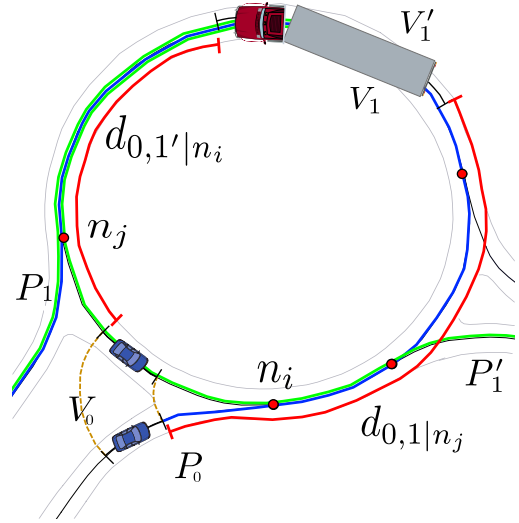


Fig. 4. The truck trajectories are estimated considering several virtual instances of the same vehicle in accordance with [24], assigning a possible trajectory for each instance. In this case, the truck can be both ahead and behind the AD vehicle, as explained in section III-A.

This approach has several advantages. First, it is helpful in cases where estimating the MD vehicle's behavior is problematic. If we consider the truck in Figure 4, it is not clear *a priori* which road the truck will exit onto. To cope with this, we propose representing the truck with two instances  $V_1$  and  $V_1'$  as we previously did in [24]. Each instance represents a possible path that the truck may follow. In cases where there is no ambiguity in truck positioning, the two instances overlap. In addition, once the position ambiguity is resolved, e.g., when it is clear that an MD vehicle has chosen one of the two branches of a bifurcation, the instance on the wrong candidate path is eliminated, and new instances are added if new ambiguities appear.

This method can also help to solve some situations of relative positioning inside the roundabout ring that arise in cyclic graphs. If we look carefully at Figure 4 and apply the virtual platooning algorithm as in [24], we see that the paths of  $V_0$  and the instance  $V_1$  of the truck have  $n_i$  as the first intersecting node (from the point of view of the AD vehicle  $V_0$ ). This means that  $V_1$  is virtually behind  $V_0$ , even though this is clearly not really the case, because the truck can be seen both behind and ahead  $V_0$ , depending on the point of view. Moreover, when considering long objects, such as a truck with a trailer, this ambiguity is more flagrant. This particular behavior is due to the circular shape of the roundabout ring, and in some cases it may produce an erroneous representation of the scenario. In fact, during the insertion maneuver  $V_0$  needs to consider the presence of the truck as a vehicle to follow, and it also needs to consider it as a possible oncoming vehicle on the left side of the roundabout. To overcome this issue, if we again consider the instance  $V_1'$  that represents the same vehicle but with a different path, we can see that it has  $n_j$  as the first intersecting node with the path of  $V_0$ . This means that, in this case,  $V_0$  is behind the instance  $V_1'$ . This representation is the inverse of the previous representation in which the instance

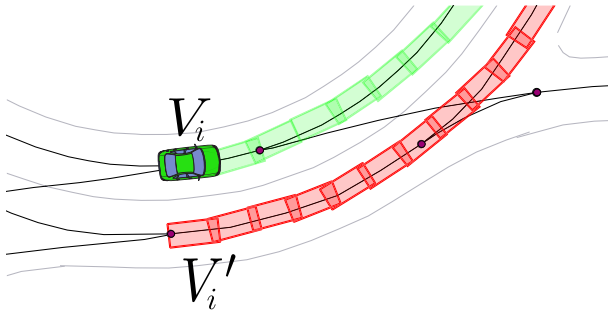


Fig. 5. Strategy for handling lane change maneuvers in a two-lane roundabout. The vehicle trajectory is green and the corresponding lane occupation is red.

$V_1$  of the truck was virtually behind  $V_0$ . It is therefore clear that in a roundabout there may be several instances of the same object with a different relative positioning w.r.t. the AD vehicle. In other words, an MD vehicle can be both virtually ahead of and behind the AD vehicle. This method enables the AD vehicle to overcome the problem of the roundabout loop and allows it to consider all the possible MD vehicle configurations in the scenario.

### B. Lane Change and Nudging Behaviors

In this section we look at modeling the behavior of other vehicles in a multi-lane roundabout. We consider again the roundabout shown in Figure 1. One of the most difficult issues in this scenario is handling vehicles that maneuver from the inner ring of the roundabout to the outer ring, which is the ring used by the AD vehicle in our method. In practice, it is very challenging to predict a lane change maneuver, especially when vehicles attempt to make a lane change with nudging [21]. In [25] we proposed three different strategies for handling the navigation of multiple vehicles inside a two-lane roundabout.

In order to predict these lane change maneuvers we again use the concept of virtual vehicles, just as we did to predict other vehicles' intentions (see Fig. 4). The main idea is to generate an extra virtual instance of a given vehicle located in the inner lane of the roundabout that can occupy the outer lane according to a chosen occupancy strategy.

The three strategies proposed in [25] generate an extra virtual instance as follows:

- 1) Systematically occupying both lanes of the roundabout ring if at least one lane is occupied.
- 2) Occupying both lanes where there is significant physical occupancy by a vehicle, but not otherwise.
- 3) Occupying the outer lane where the intention of a vehicle to change lane has been detected, but not otherwise.

In [25] we showed that strategy 3 above behaves as a compromise in terms of safety and cautiousness. However, in order to use this strategy effectively, a precise lane change intention detector is required. In this work, given that our system architecture does not possess this kind of lane change intention predictor, and given that our roundabout test-bed comprises a one-lane roundabout only, we limit ourselves to strategy 1 (Fig. 5).

## IV. ROUNDABOUT CROSSING METHOD

### A. Insertion Strategy

In this work we assume that the state of an MD vehicle  $V_i$  is represented as

$$V_i = [\underline{s}_i, \bar{s}_i, v_i, P_i], \quad (9)$$

where  $[\underline{s}_i, \bar{s}_i]$  are the lower and upper bounds over the curvilinear occupancy of  $V_i$  encompassing both the size and the uncertainty bounds over its position estimate,  $v_i$  is its estimated longitudinal speed, and  $P_i$  its predicted path.

The values of  $\underline{s}_i$  and  $\bar{s}_i$  are computed as explained in section II. From the perspective of the AD ego vehicle, the state of a nearby vehicle is typically provided by a perception system able to detect, track and map-match. In this work, in order to be compatible as far as possible with most of the state-of-the-art road user detection algorithms, no other assumptions have been made regarding the information about surrounding vehicles. In particular, the reader may refer to [5], which describes how the values in equation (9) are provided to our system.

We constrain the AD vehicle to navigate only in the outer lane of the roundabout. In other words, we do not allow the AD vehicle to overtake and change lanes as it crosses the roundabout. With this simplification, the navigation algorithm only needs to control the longitudinal motion of the AD vehicle to perform the task, the lateral control being done by path following.

Moreover, to cross a roundabout successfully, we need to take into account not only the safety inter-distance w.r.t. the vehicle ahead, but also the priority relationships in the roundabout scenario. In a roundabout, the priority lanes are situated inside the roundabout ring, while the non-priority lanes are in the entry branches. From these considerations we derive three rules to describe the ideal behavior of an AD vehicle. An AD vehicle:

- R1) must maintain a safe inter-distance w.r.t. the vehicle ahead;
- R2) must respect the traffic rules (vehicles inside the roundabout have the right of way);
- R3) must, as far as possible, avoid stopping on the carriageway.

This means that a vehicle in a non-priority lane is allowed to enter the roundabout only if the insertion maneuver does not influence the behavior of another vehicle with a higher priority. In other words, the entering vehicle is not allowed to force a priority vehicle to reduce its speed. A priority vehicle follows its reference speed profile and regulates its inter-distance only with respect to vehicles that have the same priority level. It is also desirable for the AD vehicle to execute an insertion that is as smooth as possible without stopping at the point of entry to the roundabout, which requires it to anticipate the behavior of the other vehicles.

### B. Classification of Roundabout Lanes and Priorities

In accordance with the problem statement and the three rules listed in section IV, we propose to decompose a roundabout into three zones as illustrated in Figure 1. Each zone describes

sub-steps of the insertion maneuver and a different priority rank, as follows:

a) The *Decision Zone* (in green in Fig. 1) is before the merging into the roundabout ring. In this zone, the AD vehicle does not have priority w.r.t. vehicles in the roundabout. It has to evaluate the possibility of a safe insertion in the roundabout without violating priority constraints.

b) The *Transition Zone* (in yellow in Fig. 1) is the final part of the entry lane where it merges with the roundabout ring. In this part, the AD vehicle performs a transition to enter into the roundabout. When the AD vehicle is in that zone, a safety inter-distance w.r.t. a potential oncoming MD vehicle on the roundabout ring must be maintained in order to allow a safe insertion.

c) The *Ring Zone* (in red in Fig. 1) corresponds to the roundabout ring. In this zone, the insertion maneuver is completed and the AD vehicle follows the nearest MD vehicle in the roundabout or drives at its nominal speed if it is alone.

d) The *Exit Zone* (in dark gray in Fig. 1) is the zone where the AD vehicle leaves the roundabout and continues its navigation following its path.

As a consequence, the crossing of the whole transition zone must be taken into account in the decision-making procedure. Once a vehicle enters that zone, it can no longer change its decision. If it needs to stop, this means that the decision to enter the roundabout was erroneous, and the approaching vehicle is obliged to decelerate in order to avoid a collision. Finally, once the AD vehicle has crossed the transition zone it has the same priority as all the vehicles in the roundabout ring.

### C. Intervals-Based Virtual Platooning

One well-known technique for intersection crossing is to establish a crossing order between oncoming vehicles [26]. However, when using intervals to represent the curvilinear occupancy by vehicles, there may be cases where there is no total order between vehicles, if we apply virtual vehicle methods such as in [26] and [24]. This is due to a possible overlapping of intervals. Let us consider the case illustrated in Figure 3a, where the trajectory of the AD vehicle (in blue) crosses that of an MD vehicle (in green) at a point  $n$ . Let us define  $[\underline{s}, \bar{s}]$  as the curvilinear occupancy by the AD vehicle w.r.t. the origin  $n$ , and similarly  $[\underline{s}_i, \bar{s}_i]$  for a MD vehicle  $V_i$ . Figure 3b illustrates the six possible relative positions between the AD vehicle and the MD vehicles with their corresponding virtual projections (Fig. 3c).

Note that if there is no priority constraint between the vehicles, using intervals does not lead to a unique order among them. It is therefore necessary to choose an insertion policy [24]. This issue is beyond the scope of this paper, since we consider that vehicles inside the roundabout have the highest priority. In Figure 3c it will be remarked that in all the cases (2), (3), ..., (6), the AD vehicle cannot be guaranteed to be in front of the oncoming MD vehicle. These cases should not occur when the AD vehicle goes through the transition zone, since they represent infringements of rule R2. If the AD vehicle cannot guarantee that such cases will not occur, the

AD vehicle decreases its speed to let the oncoming vehicle go ahead. This maneuver can lead either to a safe stop at the give-way line, or to an adaptation of the AD vehicle's speed, depending on the relative overlapping of intervals over time. In other words, the AD vehicle tries to adapt its behavior as far as is possible to let the other car go first, and if it cannot do so, it performs a safe stop. Note that the only case where the AD vehicle may expect to be in front of the MD vehicle is the case (1).

Let us consider the scenario depicted in Figure 4, where the AD vehicle  $V_0$  is in the decision zone of the roundabout, and an oncoming priority MD vehicle  $V_1$  is in the ring zone. This scenario clearly corresponds to case (1), with  $d_{0,1}^* > 0$ .

Let us define  $t_0$  as the time when the front part of the AD vehicle enters the transition zone. At a given time  $t < t_0$  (i.e., when the AD vehicle is still in the decision zone), it can be seen that having

$$d_{0,1}^*(t_0) > d_{\text{safe}}, \quad (10)$$

is not sufficient to ensure an insertion maneuver that is safe and that respects rights of way in accordance with rule R2. If the speed  $v_1$  of vehicle  $V_1$  is greater than the speed  $v_0$  of  $V_0$ , the inter-distance between the two vehicles will diminish over time. This shows that vehicle kinematics must be taken into account at the decision-making level.

Considering again rule R2, the AD vehicle needs to guarantee that equation (10) will be satisfied throughout the insertion maneuver, i.e., from the moment that the upper bound  $\bar{s}$  enters the transition zone until the lower bound  $\underline{s}$  leaves it. Let  $\Delta t$  be the time needed by the AD vehicle to traverse the transition zone completely, i.e., the back of the AD vehicle has exited it. The decision to enter the roundabout is taken if:

$$\forall t \in [t_0, t_0 + \Delta t], \quad d_{0,1}^*(t) > d_{\text{safe}}. \quad (11)$$

In order to guarantee the inequality of equation (11), we need to know how both  $\underline{s}_{0|n}$  and  $\bar{s}_{1|n}$  evolve over time. In this work, we assume that both vehicles drive at a constant speed. This assumption may seem simplistic, but it is representative of real driver behavior as indicated in the INTERACTION dataset. Indeed, within the roundabout ring in the INTERACTION dataset, the speed profiles of the MD vehicles have a standard deviation of less than 1 m/s in average. Under this assumption, we have  $\Delta t = l/v_0$ , where  $l$  is the length of the transition zone, and the kinematics of each interval can be expressed as follows (bearing in mind the aforementioned considerations):

$$\underline{s}_{0|n}(t) = \underline{s}_{0|n}(t_0) - v_0 \cdot (t - t_0), \quad (12)$$

$$\bar{s}_{1|n}(t) = \bar{s}_{1|n}(t_0) - v_1 \cdot (t - t_0). \quad (13)$$

Substituting equations (12) and (13) in (11), we obtain

$$\underbrace{\bar{s}_{1|n}(t_0) - \underline{s}_{0|n}(t_0)}_{=d_{0,1}^*(t_0)} + (v_0 - v_1) \cdot (t - t_0) \geq d_{\text{safe}}. \quad (14)$$

The inequality (14) needs to hold  $\forall t \in [t_0, t_0 + \Delta t]$ .

If  $v_0 > v_1$ , it leads to

$$d_{0,1}^*(t_0) \geq d_{\text{safe}}. \quad (15)$$

This means that if the AD vehicle is traveling faster than  $V_1$ , it can insert if it is sufficiently ahead of  $V_1$  at  $t_0$ .

Otherwise, if  $v_0 < v_1$ , we have

$$d_{0,1}^*(t_0) \geq d_{\text{safe}} + \left(\frac{v_1}{v_0} - 1\right)l. \quad (16)$$

It can be seen that the relative speed  $v_1/v_0$  needs to be taken into account in the decision, and that if equation (16) holds at  $t_0$ , it also holds over the whole interval  $[t_0, t_0 + \Delta t]$ . This is particularly useful in the case where the AD vehicle accelerates from a low speed ( $v_0 = 0$  when stopping at a give-way marking) in order to enter the roundabout, because all the speed changes by the AD vehicle are already encompassed in the decision. In other words, the decision taken at  $t_0$  cannot change if the speed of the AD vehicle increases.

Inequality (16) allows the AD vehicle to decide if it has enough space to maintain a safety inter-distance w.r.t. a possible oncoming vehicle, knowing its velocity and its occupancy at time  $t_0$ . In the case where equation (16) is not satisfied, the ego vehicle slows down to perform a safe stop at the end of the decision zone (that coincides with the give-way marking).

Nevertheless, once the speed of the AD vehicle is close to zero, it is difficult for the AD vehicle to find a sufficiently large gap in order to perform the insertion. This is due to the singularity present in equation (16) when  $v_0 = 0$ . In fact, the function  $h(v_0) = \left(\frac{v_1}{v_0} - 1\right)$  is seen to have a peak towards  $+\infty$  for  $v_0 \rightarrow 0$ . This degrades the performance of the algorithm once the AD vehicle has stopped: it will only insert once no oncoming vehicle is present. To overcome this, we propose to replace  $h(v_0)$  with another function for the case  $v_0 < v_1$ . In particular, we look for a function that meets the following criteria:

- 1) As  $v_0 \rightarrow 0$ , the value of the function becomes less dependent on  $v_0$ .
- 2) For  $v_0 = 0$  the value of the function depends at least on  $v_1$ .

The main idea is to have a function allowing a safety gap to be set that depends only on the other vehicle's speed  $v_1$ . This solution is useful when the dynamic of the system is not well known and we need to perform a prediction without being too pessimist. In this work we have chosen to use a function with the following form:

$$\hat{h}(v_0) = A \left( \frac{1}{2} - \frac{1}{1 + e^{-\alpha(v_0 - v_1)}} \right) \quad (17)$$

where the two parameters  $A$  and  $\alpha$  need to be tuned experimentally to make a good insertion maneuver (see section V-C). Equation (16) now becomes

$$\begin{cases} d_{0,1}^*(t_0) \geq d_{\text{safe}} & \text{if } v_0 > v_1, \\ d_{0,1}^*(t_0) \geq d_{\text{safe}} + \hat{h}(v_0)l & \text{else.} \end{cases} \quad (18)$$

or equivalently

$$\begin{cases} d_{0,1}^*(t_0) - d_{\text{safe}} \geq 0 & \text{if } v_0 > v_1, \\ d_{0,1}^*(t_0) - d_{\text{safe}} - \hat{h}(v_0)l \geq 0 & \text{else.} \end{cases} \quad (19)$$

Figure 6 illustrates the function  $\hat{h}$  for several values of  $A$  and  $\alpha$ . It can be seen that the singularity present in  $h$  for

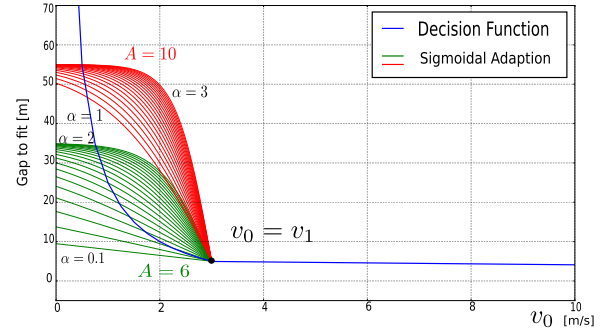


Fig. 6. The behavior of the decision function with  $h$  and  $\hat{h}$  (18) in terms of required inter-distance for several values of  $A$  and  $\alpha$  for a fixed value of  $v_1$ ,  $l$  and  $d_{\text{safe}}$ .

$v_0 = 0$  is avoided. For instance, for  $\alpha = 1$  and  $A = 10$ , if the AD vehicle is at  $v_0 = 0$ , then it decides to enter if the inter-distance is around 50 m. Applying the above to situations where there is traffic flow, algorithm 1 gives the insertion strategy in the general case. It will be remarked that where the AD vehicle has at least one vehicle behind and one ahead, the prediction (Eq. 19) is made with regard to the speed of the preceding vehicle rather than the AD vehicle itself. This takes into account the possible presence of a preceding vehicle during the insertion maneuver. On the other hand, where there is at least one vehicle behind the AD vehicle that does not satisfy Eq. 19, the AD vehicle performs a safe stop at the give-way marking, and then, in order to avoid unsafe configurations, when entering the roundabout it selects the farthest object that does not satisfy Eq. 19 as the vehicle to follow.

## V. CASE STUDY IN SIMULATION

### A. Traffic Simulation for Autonomous Vehicles

Simulation of the traffic environment is a technique widely used in relation to self-driving cars. It is an approach that has several advantages. Simulations are safer, more efficient, and cheaper than live testing with real vehicles. With simulations it is also possible to generate a variety of different driving scenarios that would be hard to create using real road agents, thus reducing the risk of damage to material and injury to people.

Of course, in order for simulators to be meaningful when testing AD applications, a realistic simulation level w.r.t. the real world is essential.

It is almost impossible to find a simulator that satisfies every single requirement for autonomous driving. However, there are a wide range of simulators that target individual aspects (traffic flow, sensor simulation, etc.).

In this paper we focus on traffic flow simulators. Simulators concerned with traffic flow can be separated into two main types, namely *macroscopic* and *microscopic* traffic simulators. Macroscopic simulators are mainly used to study large-scale flow problems, relating to road capacities and bottlenecks over a large complex road network. In these simulators the dynamics of individual traffic agents are quite simple, and sometimes there is an absence of information relating to single traffic agents. The level of detail is kept low to allow the simulator to

**Algorithm 1** Roundabout Insertion for an AD Vehicle  $V_0$ **Require:**  $V_0, t_0$ 

```

1:  $[V_1, V_2, \dots, V_c] \leftarrow \text{Perception()}$   $\triangleright$  Other road users
2:  $\Delta t \leftarrow l/v_0$ 
3:  $leader \leftarrow \emptyset, d_{leader} \leftarrow -\infty$ 
4:  $V_{RISK} \leftarrow []$   $\triangleright$  List of vehicles that represent a risk
5: for  $j = 1 : c$  do
6:    $n \leftarrow \text{FindFirstCommonNode}(P_0, P_j)$ 
7:   if  $n = \emptyset$  then
8:     Continue  $\triangleright V_j$  does not cross the path of  $V_0$ 
9:   else
10:     $d_{0,j|n}^*(t_0) \leftarrow \bar{s}_{j|n}(t_0) - \underline{s}_{0|n}(t_0)$ 
11:    if  $d_{0,j|n}^*(t_0) < 0$  then  $\triangleright$  Vehicle  $V_j$  in front of  $V_0$ 
12:      if  $d_{0,j|n}^*(t_0) > d_{leader}$  then
13:         $d_{leader} \leftarrow d_{0,j|n}^*(t_0)$ 
14:         $leader \leftarrow V_j$ 
15:      else
16:        Continue
17:      end if
18:    else  $\triangleright$  Vehicle  $V_j$  behind or intersecting  $V_0$ 
19:      if  $d_{0,j}^*(t_0) \geq d_{safe} + \left(\frac{v_j}{v_0} - 1\right)l$  then
20:        Continue  $\triangleright$  Constraint satisfied
21:      else
22:         $V_{RISK} \leftarrow V_{RISK} \cup [V_j, d_{0,j}^*(t_0)]$ 
23:      end if
24:    end if
25:  end if
26: end for
27: if  $leader = \emptyset$  then
28:    $v_d \leftarrow v_n$   $\triangleright$  Go with nominal speed
29: else
30:    $v_d \leftarrow leader.speed$   $\triangleright$  Speed of vehicle ahead
31: end if
32: if  $V_{RISK} \neq []$  then  $\triangleright$  Change leader vehicle
33:    $leader \leftarrow \max_{d_{0,j|n}^*} [V_{RISK}]$ 
34:    $v_d \leftarrow leader.speed$ 
35: end if
36:  $\text{SetSpeed}(v_d)$   $\triangleright$  Perform longitudinal control

```

perform simulations involving hundreds of vehicles and with a reasonable computation time. Macroscopic simulators are therefore not suitable for AD navigation. Interactions between the vehicles and driving maneuvers (e.g., lane changes) are overly simplified.

Microscopic simulators, on the other hand, focus on the characteristics of single road agents rather than global vehicle flow. The description level of every simulated road agent must be high and detailed. Longitudinal and lateral motion of vehicles can be obtained via a number of built-in driving models customizable with a large set of driving parameters. For some simulators, e.g., SUMO, it is also possible to parameterize some complex maneuvers such as lane changes.

For this work we decided to use the well-known SUMO simulator [20], widely used for microscopic traffic simulation in the area of collision avoidance [12], [19]. We refer interested readers to [12].

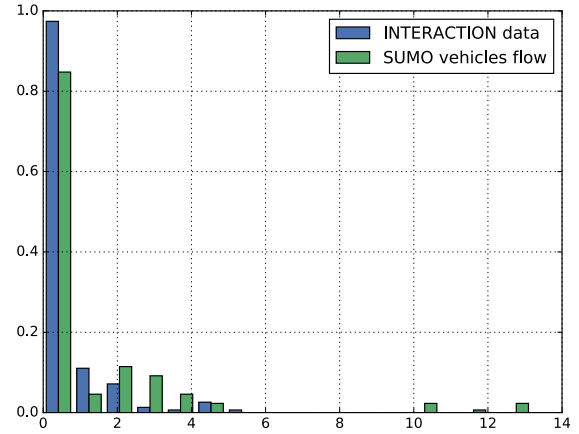


Fig. 7. Comparison between the  $\Delta TTIC_{min}$  distribution generated from the same sequence of the USA\_Roundabout\_FT scenario of the INTERACTION dataset (blue) and the one obtained from the traffic flow simulated by the resulting traffic flow simulation in SUMO (green). One can notice that an highly dynamic behavior ( $\Delta TTIC_{min} \leq 1$  s) is present in both scenarios.

Although simulation performance is becoming increasingly precise, it is important to be aware that a gap still exists between simulated maneuvers and those performed in the real world. However, state-of-the-art simulation technology is helping to narrow that gap, with regular new releases for existing simulators.

### B. Vehicle Flow Generation

This section proposes a method to reproduce a realistic vehicle flow in a simulated environment. We use data from the real traffic dataset INTERACTION. As stated previously, we chose this dataset because it provides data for highly interactive road users, including situations where road users have adversarial motion behaviors. Traffic flow data recordings at a microscopic level in a dense traffic flow situation are provided for every roundabout in the dataset.

In order to compare the performance of a generated flow w.r.t. the original flow, a metric is required. This metric must be able to capture the degree of interaction between road users, especially in complex driving situations (close navigation in parallel lanes, overtaking, intersection insertions without markings, etc.).

The authors in [41] propose using the  $\Delta TTIC_{min}$  (minimum time-to-conflict-point) difference metric for an interactive pair of vehicles [38]. Following [41], we define an interactive pair of vehicles as a pair of vehicles that have at least one common point in their respective reference paths.

For a given interactive pair of vehicles, the corresponding  $\Delta TTIC_{min}$  can be computed as

$$\Delta TTIC_{min} = \min_{t \in [T_{start}, T_{end}]} (TTIC_1^t - TTIC_2^t), \quad (20)$$

where  $TTIC_i^t = \Delta d_i^t / v_i^t$ , and  $i = 1, 2$ , is the traveling time to the conflict point of each vehicle in the interactive pair. Note that  $\Delta d_i^t$  is the distance from the vehicle position to the conflict point in a Frenet frame, which is equivalent to the absolute value of the curvilinear abscissa w.r.t. to a conflicting node, as defined in equation (6). Where  $\Delta TTIC_{min} \leq 3$  s,



there is deemed to be an interaction between the vehicles. Moreover, an interaction between two vehicles is termed *intense* if  $\Delta TTIC_{min} \leq 1$  s. In this study we focused on the data from the “USA\_Roundabout\_FT” scenario, because this is the most interactive two-lane roundabout, with the highest number of vehicles and recorded sequences.

We will now look at how SUMO can be used to generate a flow of vehicles based on the INTERACTION traffic data with the same degree of interaction in terms of  $\Delta TTIC_{min}$ . This has several advantages. It allows us to quantify the performance of our algorithm w.r.t. the interaction degree of the traffic scenario. It can give us a vehicle flow with the same degree of interaction as on a different roundabout test-bed (for example that shown in Figure 1). And a simulated vehicle flow in SUMO can react dynamically to the ego vehicle decisions (e.g., braking when the insertion maneuver is too aggressive), while this is not the case for a recorded flow dataset.

Using the curvilinear formalism introduced in section IV-C, we implemented an algorithm to compute common intersection points between vehicle trajectories. In our case, trajectories are encoded as a list of links that a vehicle follows on the road map, and consequently the intersection point is one node of the map. Unfortunately, neither vehicle trajectories nor a lane-level high-definition map are present in the INTERACTION dataset, and they therefore need to be computed offline.

To this end we compute an HD map representation of the “USA\_Roundabout\_FT” starting from its *lanelet2* [31] representation provided with the dataset interface. The main idea behind this step is to convert the *lanelet2* representation of street borders to a centerline representation using our curvilinear formalism. A Voronoi distances algorithm is used to compute the set of center lines, that is to say equidistant points from lane borders. However, this procedure is not completely automatic and can be error-prone where roads intersect and merge. Some manual corrections were therefore done offline at the end of this step.

Although the “USA\_Roundabout\_FT” has more than one lane, we represent it with only one lane in our HD map. This is to better capture the interaction between vehicles inside the roundabout ring, assigning vehicles navigating on multiple lanes to the same portion of the map, even where vehicle trajectories do not overlap. This is a reasonable strategy given that navigation in parallel lanes implies a possible implicit interaction that should be taken into account (we cannot know in advance the intentions of the vehicles).

To compute a reference trajectory for the vehicles w.r.t. the HD map, we implemented a map-matching procedure via the INTERACTION dataset animation tool. From the raw data on trajectories this procedure directly computes the corresponding curvilinear position on the HD map and lists all the map links that a vehicle traverses.

Once the intersection points between every interactive pair have been found, we use the HD map to compute the curvilinear distances  $\Delta d_i^l$  via the algorithm presented in section IV-C.

As for the input data to SUMO, we compute for every track of the “USA\_Roundabout\_FT” the arrival times, arrival speeds, inter-distances w.r.t. the vehicle ahead and behind.

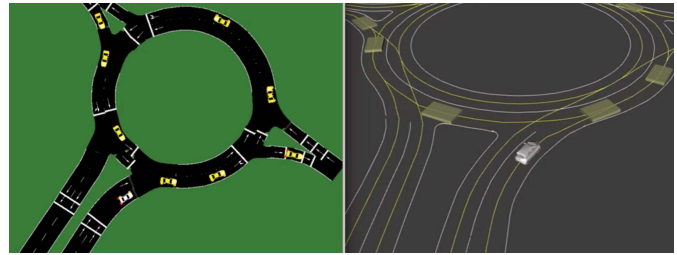


Fig. 8. The coupled simulator. The AD vehicle is in gray, while all the MD vehicles are in yellow. Note that the roundabout in SUMO was created using the HD map representation of the roundabout shown in Fig. 1.

The empirical distribution of these observed data was estimated [25] and used to randomly create vehicle objects with the same properties as their real-world counterparts. Finally, the speed of each vehicle was bounded by the speed limit provided by the road shape and the route attribute of each vehicle object was chosen randomly, with more weight given to routes intersecting the AD vehicle’s route.

The output of the system is a traffic flow with the  $\Delta TTIC_{min}$  distribution similar to that computed using the real traffic data. Figure 7 illustrates the output of the simulation for a recorded traffic sequence from INTERACTION. Highly dynamic behavior ( $\Delta TTIC_{min} \leq 1$  s) is present in both scenarios.

The high degree of interaction between interactive pairs comes mainly from the double lane of the roundabout ring. Where two or more vehicles are traveling very close to each other in parallel lanes in the roundabout ring, there is a high volume of interaction between them.

### C. Simulation Results

To validate our approach we used two simulators. The SUMO simulator [20] was used for microscopic traffic flow generation, while a ROS-based simulation framework was used to implement the navigation algorithm and the AD vehicle dynamics. The coupling and synchronization of the two simulators was done using the TraCi SUMO library and its Python API. A detailed explanation of the time synchronization and coupling of the two simulators can be found in [13].

We imported our HD map representation of the test-bed roundabout into the SUMO simulation environment using the Nedit and Netconvert tools included in the SUMO suite [18]. Figure 8 shows an overview of the roundabout scenario in both simulators.

For each simulation, a random high-density vehicle flow that meets the  $\Delta TTIC_{min}$  criterion was generated over a fixed time horizon  $T = 200$  s. Each simulation was between 200 and 400 seconds, giving more than 1 hour of simulation in total. The number of vehicles for each flow randomly varied between 50 and 175, giving more than 5,000 vehicles altogether.

The limits were chosen to capture a wide range of scenarios, ranging from a sparse traffic flow to a denser vehicle stream. To simulate localization uncertainty for the AD vehicle and perception uncertainty for MD vehicles, a  $\pm 1$  m bound is added to the curvilinear interval  $[\underline{s}_i, \bar{s}_i]$  and  $[\underline{s}, \bar{s}]$ , which represents the projection of the vehicle footprint on its lane.

TABLE I

THE AVERAGE INSERTION TIME, THE PERCENTAGE OF AVERAGE WAITING TIME RELATIVE TO THE NOMINAL CASE AND THE AVERAGE NUMBER OF INSERTIONS AS A FUNCTION OF VEHICLE FLOW FOR THE SINGLE-LANE ROUNDABOUT CASE

Flow size	50	75	100	125
Crossing Time (s)	5.60 (+1.3)	7.32 (+1.7)	10.05 (+2.39)	15.26 (+3.63)
Number of Insertions	24	21	18	16

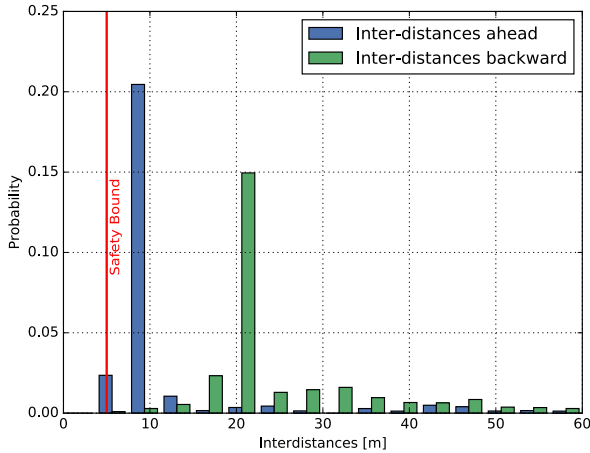


Fig. 9. Inter-distance distributions w.r.t. the vehicle ahead (blue) and behind (green) during an insertion maneuver. The red line represents the 5 m safety gap.

To experimentally validate our approach, we used the technique described above in relation to various scenarios in order to generate high density traffic flows. The simulations were done using the HD map representation of the roundabout shown in Figure 1.

To better quantify the performance of our algorithm, we computed the distributions of the inter-distances w.r.t. the vehicle ahead and behind during the crossing of the transition zone. Figure 9 shows the inter-distance distributions for both the ahead and behind gaps. It can be seen that the behind safety gap always meets the safety criterion, whereas there is a small violation of the safety bound w.r.t. vehicles ahead. This violation is due to deficiencies in the controller and can be ignored.

We are also interested in quantifying the insertion rate w.r.t. the size of vehicle flow. Table I illustrates the average insertion rate and waiting time as a function of the number of vehicles in the flow. The total length of the decision and transition zones is 33.4 m and the nominal speed in this zone is 30 km/h, which gives us a nominal waiting time of 4.2 seconds.

The number of insertions is seen to decrease w.r.t. the number of vehicles in the flow. Conversely, the waiting time increases.

## VI. REAL EXPERIMENTS

In order to validate our strategy in a real scenario, we implemented the whole system architecture on an autonomous Renault Zoé using ROS middleware. More details about the system architecture of our experimental cars are provided in [40]. We first tested roundabout insertion in a hybrid environment (*i.e.*, with simulated vehicles traveling on the



Fig. 10. The experimental “Seville” circuit and the experimental Renault Zoés for the real outdoor tests. In our configuration, the white car (fully autonomous) is the AD vehicle, while the others are the road users (manually driven).

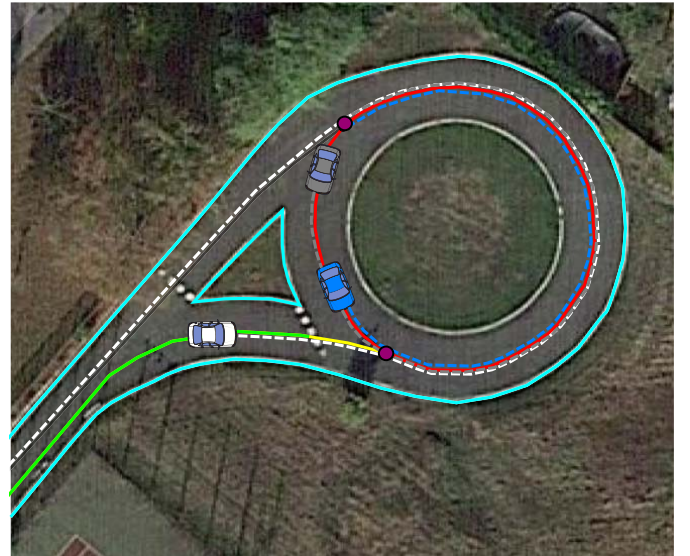


Fig. 11. A bird-eye view of the experimental circuit with the roundabout zones (solid lines) and vehicles trajectories (dashed lines). The AD vehicle is white and the MD vehicles are blue and gray. Notice that the situation case described in section III-A may occur.

test track) then with real road agents detected using a LiDAR-based perception system. In the autonomous experimental car, the system controls the throttle, the brake pedal and the steering wheel. In our case, the longitudinal motion of the vehicle (*i.e.*, acceleration and brake) is computed according to the traffic situation. In particular, based on the algorithm described in section IV-C, the vehicle can either perform an insertion maneuver into the roundabout or reduce its speed to let other cars go ahead, possibly stopping at the give-way sign. Regarding lateral motion, a simple lane keeping is performed and for the detection part we use a state-of-the-art LiDAR object detection algorithm [5] able to provide information about the detected objects in the form  $V_i = [\underline{s}_i, \bar{s}_i, v_i, P_i]$ . Note that  $P_i$  and the curvilinear conversion were computed using the methods described above.

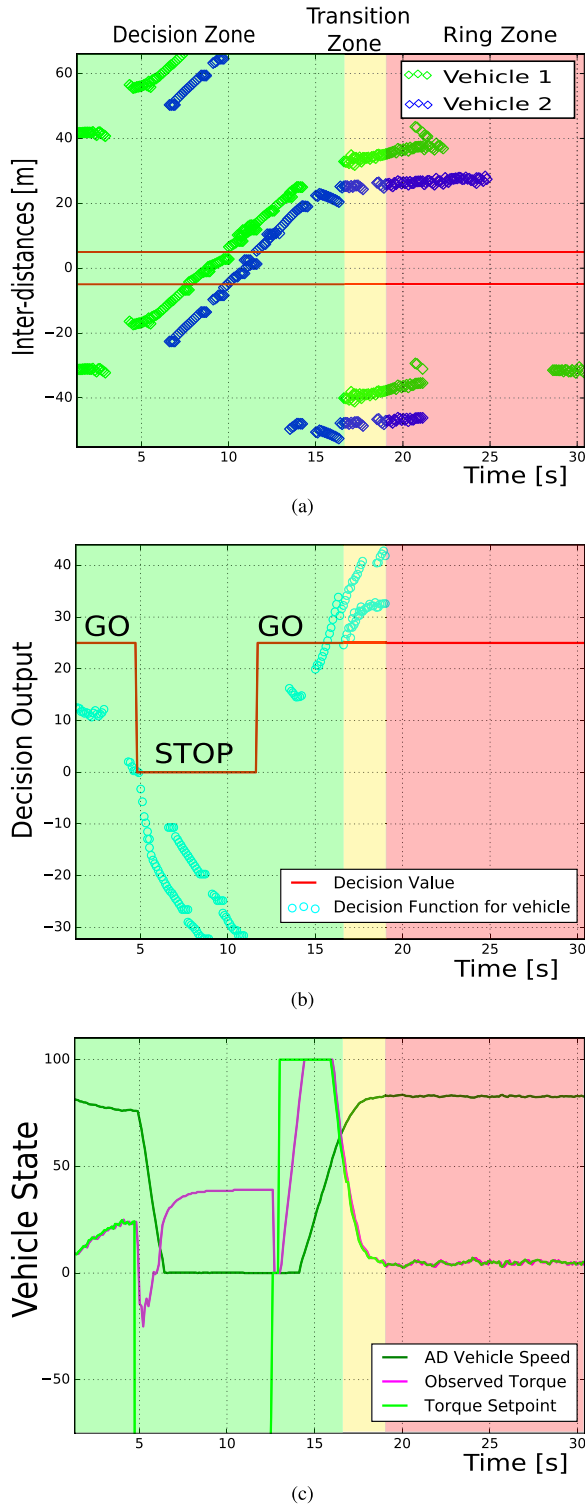


Fig. 12. Experimental results. Fig.12a illustrates the (virtual) inter-distances of the two cars w.r.t the AD vehicle as a function of time. Notice that the same vehicle may appear both behind and in front of the AD vehicle because of the vehicle instance concept. Figure 12(b) shows the value of the decision function (red) and the result of equation 19 (blue dots) for the different vehicles (virtually) behind the AD vehicle (points  $< 0$  in Fig. 12a). Figure 12(c) shows the torque setpoint and the corresponding vehicle torque and speed (amplified by a scale factor) as a function of the decision taken in Fig. 12b.

Let us consider a scenario where two cars are driving close to each other inside a roundabout doing infinite loops and the AD vehicle needs to enter the roundabout (Figure 11).

Figure 12a illustrates the (virtual) inter-distances of the AD vehicle w.r.t the other road agents (behind and ahead) in the three zones. As we can see, the sign of the inter-distance depends on the relative positioning between the MD vehicles and the AD vehicle

Notice that due to the concept of vehicle instances (section III) a vehicle inside the roundabout can be both ahead of and behind the AD vehicle, which is consistent with the circular shape of the roundabout. The colors of the background denote the zone of the roundabout where the AD vehicle is located. While crossing the transition and ring zones the safety gap is always maintained.

If we observe the points in the decision zone carefully, we see that the other road agents virtually overtake the AD vehicle. Figure 12b shows that as a result of the negative values of the constraint of equation (18), the decision changes from go to stop. As a consequence, the AD vehicle lets the other vehicles go ahead and enters the roundabout behind them.

To better illustrate the situation at the decision and control level, Figure 12c shows the system behavior during the whole maneuver. In particular, we can see that the set-point torque changes following the decision output in Figure 12b. Consequently, the vehicle speed and the torque applied to the engine change accordingly.

When the red function decreases to zero in Figure 12b, the controller performs a safe stop maneuver. Conversely, once the decision-making part decides to let the vehicle enter the roundabout, the controller accelerates accordingly. Note that where the vehicle is completely stationary (*i.e.*,  $v = 0$ ) it is still in the decision zone. This means that no safety violation occurs during the maneuver.

## VII. CONCLUSION

This work has presented a method based on virtual vehicles for performing a safe roundabout crossing. We first described how the virtual instances of vehicles can be used to handle the particular shape of roundabouts. We then proposed an approach based on this technique and occupancy intervals for computing the best gap to choose during a roundabout insertion maneuver. This approach was tested under a simulated traffic flow generated from real data. The degree of interaction of the generated flow was used to create a scenario close to real-world driving. We showed that the proposed insertion maneuver ensures safety. We also proposed some performance indexes to evaluate its efficiency in terms of traffic fluidification.

Our method was tested on an experimental test circuit with real road users and a real AD vehicle, in order to evaluate the performance of the proposed algorithm in a real-world scenario with a perception system that provides information about the surrounding road agents.

As a future prospect our objective is to extend the tests with a real vehicle to more complex scenarios (e.g., two-lane roundabouts with more MD vehicles) and to include in their architecture a lane change intention detector. In order to obtain a better performance when the traffic flow is dense, it would also be interesting to study how a negotiation layer involving

intentions and actions of other parties during the interaction could be implemented. This layer would potentially play an important role when AD vehicles are involved in a decision process where rules are not clear.

#### ACKNOWLEDGMENT

This work was carried out within SIVALab, a shared laboratory between Renault and Heudiasyc CNRS/UTC, through the TORNADO project.

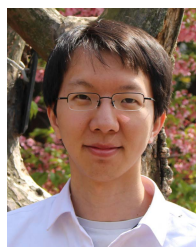
#### REFERENCES

- [1] M. Althoff, M. Koschi, and S. Manzing, "CommonRoad: Composable benchmarks for motion planning on roads," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 719–726.
- [2] B. Qadeer Baig, M. Perrollaz, and C. Laugier, "Advances in the Bayesian occupancy filter framework using robust motion detection technique for dynamic environment monitoring," *IEEE Robot. Autom. Mag.*, vol. 21, no. 1, pp. 40–48, Mar. 2014.
- [3] L. Banjanovic-Mehmedovic, E. Halilovic, I. Bosankic, M. Kantardzic, and S. Kasapovic, "Autonomous vehicle-to-vehicle (V2 V) decision making in roundabout using game theory," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 8, pp. 292–298, 2016.
- [4] P. Bender, J. Ziegler, and C. Stiller, "Lanelets: Efficient map representation for autonomous driving," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2014, pp. 420–425.
- [5] E. Bernardi, S. Masi, P. Xu, and P. Bonnifait, "High integrity efficient lane-level occupancy estimation of road obstacles through Lidar and HD map data fusion," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2020, pp. 1602–1607.
- [6] D. Carlino, S. D. Boyles, and P. Stone, "Auction-based autonomous intersection management," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 529–534.
- [7] L. Chen and C. Englund, "Cooperative intersection management: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 570–586, Feb. 2016.
- [8] A. de La Fortelle and X. Qian, "Autonomous driving at intersections: Combining theoretical analysis with practical considerations," in *Proc. ITS World Congr.*, Oct. 2015.
- [9] E. Debada, L. Makarem, and D. Gillet, "A virtual vehicle based coordination framework for autonomous vehicles in heterogeneous scenarios," in *Proc. IEEE Int. Conf. Veh. Electron. Saf. (ICVES)*, Jun. 2017, pp. 51–56.
- [10] V. Desaraju, H. C. Ro, M. Yang, E. Tay, S. Roth, and D. Del Vecchio, "Partial order techniques for vehicle collision avoidance: Application to an autonomous roundabout test-bed," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 82–87.
- [11] P. Falcone, M. Ali, and J. Sjöberg, "Predictive threat assessment via reachability analysis and set invariance theory," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1352–1361, Dec. 2011.
- [12] M. C. Figueiredo, R. J. F. Rossetti, R. A. M. Braga, and L. P. Reis, "An approach to simulate autonomous vehicles in urban traffic scenarios," in *Proc. 12th Int. IEEE Conf. Intell. Transp. Syst.*, Oct. 2009, pp. 1–6.
- [13] M. Garzon and A. Spalanzani, "An hybrid simulation tool for autonomous cars in very high traffic scenarios," in *Proc. 15th Int. Conf. Control, Autom., Robot. Vis. (ICARCV)*, Nov. 2018, pp. 803–808.
- [14] J. I. Ge, S. S. Avedisov, C. R. He, W. B. Qin, M. Sadeghpour, and G. Orosz, "Experimental validation of connected automated vehicle design among human-driven vehicles," *Transp. Res. C, Emerg. Technol.*, vol. 91, pp. 335–352, Jun. 2018.
- [15] T. Gindele, S. Brechtel, J. Schroder, and R. Dillmann, "Bayesian occupancy grid filter for dynamic environments using prior map knowledge," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2009, pp. 669–676.
- [16] E. Hery, S. Masi, P. Xu, and P. Bonnifait, "Map-based curvilinear coordinates for autonomous vehicles," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1699–1705.
- [17] G. Ibanez, T. Meuser, M. A. Lopez-Carmona, and D. Lopez-Pajares, "Synchronous roundabouts with rotating priority sectors (SYROPS): High capacity and safety for conventional and autonomous vehicles," *Electronics*, vol. 9, no. 10, p. 1726, 2020.
- [18] M. Klischat, O. Dragoi, M. Eissa, and M. Althoff, "Coupling SUMO with a motion planning framework for automated vehicles," in *Proc. SUMO User Conf.*, 2019, pp. 1–9.
- [19] Z. Kokkinogenis, M. Teixeira, P. M. d'Orey, and R. J. F. Rossetti, "Tactical level decision-making for platoons of autonomous vehicles using auction mechanisms," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Paris, France, Jun. 2019, pp. 1632–1638.
- [20] D. Krajzewicz, G. Hertkorn, C. Rössel, and P. Wagner, "SUMO (Simulation of Urban MObility)—an open-source traffic simulation," in *Proc. 4th Middle East Symp. Simulation Modeling (MESM)*, Jan. 2002, pp. 183–187.
- [21] K. Leung, E. Schmerling, M. Chen, J. Talbot, J. C. Gerdes, and M. Pavone, "On infusing reachability-based safety assurance within probabilistic planning frameworks for human-robot vehicle interactions," in *Proc. Int. Symp. Experim. Robot.*, 2018, pp. 561–574.
- [22] L. Li and F.-Y. Wang, "Cooperative driving at blind crossings using intervehicle communication," *IEEE Trans. Veh. Technol.*, vol. 55, no. 6, pp. 1712–1724, Nov. 2006.
- [23] S. Magdici and M. Althoff, "Adaptive cruise control with safety guarantees for autonomous vehicles," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5774–5781, Jul. 2017.
- [24] S. Masi, P. Xu, and P. Bonnifait, "Adapting the virtual platooning concept to roundabout crossing," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1366–1372.
- [25] S. Masi, P. Xu, and P. Bonnifait, "A curvilinear decision method for two-lane roundabout crossing and its validation under realistic traffic flow," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2020, pp. 1019–1025.
- [26] A. I. M. Medina, N. van de Wouw, and H. Nijmeijer, "Automation of a T-intersection using virtual platoons of cooperative autonomous vehicles," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, Sep. 2015, pp. 1696–1701.
- [27] E. R. Müller, R. C. Carlson, and W. Kraus, Jr., "Intersection control for automated vehicles with MILP," in *Proc. 14th IFAC Symp. Control Transp. Syst. (CTS)*, vol. 49, 2016, pp. 37–42.
- [28] M. Naumann, M. Lauer, and C. Stiller, "Generating comfortable, safe and comprehensible trajectories for automated vehicles in mixed traffic," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 575–582.
- [29] M. Naumann and C. Stiller, "Towards cooperative motion planning for automated vehicles in mixed traffic," *CoRR*, vol. abs/1708.06962, pp. 575–582, Aug. 2017.
- [30] T. Nguyen and T.-C. Au, "A constant-time algorithm for checking reachability of arrival times and arrival velocities of autonomous vehicles," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Paris, France, Jun. 2019, pp. 2039–2044.
- [31] F. Poggendorf, et al., "Lanelet2: A high-definition map framework for the future of automated driving," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 1672–1679.
- [32] X. Qian, J. Gregoire, A. de La Fortelle, and F. Moutarde, "Decentralized model predictive control for smooth coordination of automated vehicles at intersection," in *Proc. Eur. Control Conf. (ECC)*, Jul. 2015, pp. 3452–3458.
- [33] G. R. de Campos, P. Falcone, and J. Sjöberg, "Autonomous cooperative driving: A velocity-based negotiation approach for intersection crossing," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 1456–1461.
- [34] J. Pérez Rastelli, V. Milanés, T. De Pedro, and L. Vlacic, "Autonomous driving manoeuvres in urban road traffic environment: A study on roundabouts," *IFAC Proc. Vols.*, vol. 44, no. 1, pp. 13795–13800, 2012.
- [35] J. P. Rastelli and M. S. Peñas, "Fuzzy logic steering control of autonomous vehicles inside roundabouts," *Appl. Soft Comput.*, vol. 35, pp. 662–669, Oct. 2015.
- [36] A. T. Tran, M. Kawaguchi, H. Okuda, and T. Suzuki, "A model predictive control-based lane merging strategy for autonomous vehicles," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Paris, France, Jun. 2019, pp. 594–599.
- [37] F. von Hundelshausen, M. Himmelsbach, F. Hecker, A. Müller, and H. J. Wunsche, "Driving with tentacles: Integral structures for sensing and motion," *J. Field Robot.*, vol. 25, no. 9, pp. 640–673, Jun. 2008.
- [38] W. Zhan, L. Sun, D. Wang, Y. Jin, and M. Tomizuka, "Constructing a highly interactive vehicle motion dataset," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 1366–1372.
- [39] I. Xausa, R. Baier, O. Bokanowski, and M. Gerdts, "Computation of avoidance regions for driver assistance systems by using a Hamilton-Jacobi approach," *Optim. Control Appl. Methods*, vol. 41, no. 2, pp. 668–689, Mar. 2020.
- [40] P. Xu, G. Dherbomez, E. Hery, A. Abidli, and P. Bonnifait, "System architecture of a driverless electric car in the grand cooperative driving challenge," *IEEE Intell. Transp. Syst. Mag.*, vol. 10, no. 1, pp. 47–59, Jan. 2018.

- [41] W. Zhan *et al.*, “INTERACTION dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps,” 2019, *arXiv:1910.03088*. [Online]. Available: <https://arxiv.org/abs/1910.03088>
- [42] J. Ziegler *et al.*, “Making bertha drive—An autonomous journey on a historic route,” *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 2, pp. 8–20, Apr. 2014.
- [43] A. Zyner, S. Worrall, and E. M. Nebot, “ACFR five roundabouts dataset: Naturalistic driving at unsignalized intersections,” *IEEE Intell. Transp. Syst. Mag.*, vol. 11, no. 4, pp. 8–18, Sep. 2019.



**Stefano Masi** (Associate Member, IEEE) received the B.S. and M.S. degrees in computer engineering from the Università degli Studi di Genova (UNIGE), Italy, in 2015 and 2017, respectively, and the Master Diploma degree in engineering of complex and interacting systems from the Université de Technologie de Compiègne (UTC), France, in 2017, where he is currently pursuing the Ph.D. degree with the Heudiasyc UMR 7253, a joint research unit between UTC and CNRS. He is working as an Research and Development Engineer in the French national project Tornado. Since 2017, he has also been a member of SIVALab, a joint laboratory between Renault, UTC, and CNRS. His research interests include self-driving vehicles navigation, collision avoidance, information fusion, and environment perception applied to autonomous vehicles.



**Philippe Xu** (Associate Member, IEEE) received the B.S. and M.S. degrees in computer science from the École Normale Supérieure de Cachan, France, in 2009 and 2011, respectively, and the Ph.D. degree from the Université de Technologie de Compiègne (UTC), France, in 2014. Since 2015, he has been an Associate Professor with the Computer Science Department, UTC, where he is carrying out his research at Heudiasyc UMR 7253 (UTC/CNRS). Since 2017, he has also been a member of the SIVALab, a joint laboratory between Renault, UTC, and CNRS. His research interests include information fusion, computer vision, and machine learning applied to localization and perception for autonomous vehicles.



**Philippe Bonnifait** (Member, IEEE) received the Ph.D. degree in automatic control and computer science from the École Centrale de Nantes, France, in 1997. He is currently a Professor with the Computer Science and Engineering Department, Université de Technologie de Compiègne, France. Since 1998, he has been a part of the Heudiasyc UMR 7253, a joint UTC-CNRS research laboratory. His research interests include intelligent vehicles, and high-integrity positioning and map-matching for autonomous navigation in structured outdoor environments.