Adapting the Virtual Platooning Concept to Roundabout Crossing

Stefano Masi, Philippe Xu, Philippe Bonnifait

▶ **To cite this version:**

Stefano Masi, Philippe Xu, Philippe Bonnifait. Adapting the Virtual Platooning Concept to Roundabout Crossing. 29th IEEE Intelligent Vehicles Symposium (IV 2018), Jun 2018, Changshu, China. pp.1366-1372, 10.1109/IVS.2018.8500611 . hal-02080001

HAL Id: hal-02080001
https://hal.science/hal-02080001

Submitted on 26 Mar 2019

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

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

Adapting the Virtual Platooning Concept to Roundabout Crossing

Stefano Masi[1], Philippe Xu[1] and Philippe Bonnifait[1]

***Abstract*— Despite the tremendous growth of research regarding fully autonomous vehicles in the past few years, many safety critical scenarios, such as the crossing of roundabouts, are still open issues. One of the main challenge is to deal with the lack of visibility in complex environments. Vehicle-to-vehicle and vehicle-to-infrastructure communications offer an appealing solution to handle these situations in a cooperative way. To avoid computing an ad hoc control strategy for every possible scenario, we propose in this paper to adapt the concept of virtual platooning to roundabout crossing. This idea allows a single platooning control law to handle complex scenarios such as intersection and roundabout crossings. This work combines the use of high definition maps and a curvilinear coordinates framework to deal with any kind of roundabouts. The proposed approach is not limited to communicating autonomous vehicles but can also be used with manually driven communicating vehicles or non-communicating vehicles with the help on the infrastructure. A formal proof of the correctness of this approach is given and simulations have been carried out with a high definition map of a real roundabout. We also introduce a novel graphical representation called safety diagram to study de performances of our approach.**

I. Introduction

During the last decades, the research on autonomous driving systems has grown significantly. However, severe challenges for autonomous driving remain on some road sections that are complex for autonomous navigation, typically due to a lack of visibility. These are for instance tunnels, crossroads, roundabouts and railroad crossings which are known to be the cause of most of the road accidents. Roundabouts are increasingly used in urban areas because they allow a safer approach than traditional crossroads with traffic lights. In fact, compared to traffic signals, the research of the Insurance Institute for Highways Safety (IIHS) is indicating that roundabouts can lead to about a 90% reduction in fatalities and a 75% reduction in crashes with injuries [1]. Moreover, roundabouts guarantee also efficiency and are environmentally friendly, because a roundabout does not require a driver to stop unless there is traffic already in the circulating lanes. This can avoid to wait for a light to turn green or come to a complete stop if there is no traffic on the lanes.

While it is common to find papers in the literature that propose strategies to handle a crossroad intersection [2], [5], even focalized on cooperative driving control as [8], much fewer researches have been conducted regarding roundabouts. In [3], the authors proposed an algorithm that attempts to solve the roundabout crossing problem with the help of game theory, while some approaches to control vehicle inside a roundabout have been proposed in [10], [11]. Desaraju et al. [6] considered partial order techniques to develop a strategy in order to avoid a collision between two vehicles circulating into a roundabout. In these cases, the aspect that regards the control of the vehicles has mostly been considered, rather than the decision making part. Regarding the roundabout models, Rastelli et al. [10] used a very simple model based on a circle connected with some branches to model entries and exits, while the authors in [11] used Bézier curves to model entries and exits. In this paper, a decision making method for crossing a roundabout based on an high definition map and virtual platooning is proposed. One of the main advantages of virtual platooning [9] is to allow a single platooning control law to be adapted to several scenarios such as intersection crossing or lane merging. The aim of this paper is to generalize this concept to roundabout crossing. The map model that we use in this paper is the one exploited also in [7], and for the map matching we use the criterion explained in [7] and [4].

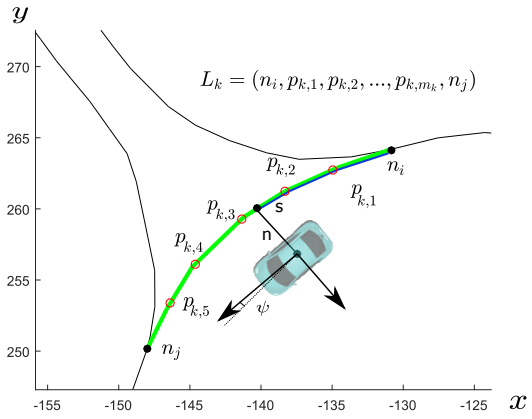


Fig. 1: The green part of the road is represented as a link L_k defined by the nodes n_i and n_j while its shape is described by the set of shape points $\{p_{k,h}\}$.

The paper is structured as follow: In section II, we describe the map model, the vehicle localization and curvilinear pose computing w.r.t. the map. In section III, the virtual platooning algorithm in the case of a crossroad intersection is presented. Section IV introduces the roundabout crossing algorithm based on virtual platooning for both autonomous and manually driven vehicles; while in section V the correctness of the approach is proved. Finally, in section VI, we present experiments carried out through simulation and introduce a novel diagram to analyze the performance of the algorithm.

[1]The authors are with Sorbonne universités, Université de technologie de Compiègne, CNRS, Heudiasyc UMR 7253, 60203 Compiègne Cedex, France firstname.surname@hds.utc.fr

II. Map Model and Curvilinear Pose

In this section, we introduce a model of the driving environment and of the vehicle pose in a map based context. Starting from the Lanelet map definition of [4], [7], we consider a links and nodes representation of the map. In particular, we use the following map components:

- Node: it is a set $N = \{n_1, \ldots, n_m\}$ of 2D points with coordinates $[x; y]$ and it is used to mark the start and the end points of a part of the lane, in particular where two lanes split, merge or cross.
- Link: it is the portion of the lane between two nodes, starting and ending nodes that define the flow direction of the lane. The geometry of this part is represented as a polyline, which is a sequence of shape points. It is important to note that no crossing can occur within a link, but only at the starting or ending nodes. A link is typically defined as $L_k = (n_i, p_{k,1}, \ldots, p_{k,m_k}, n_j)$.
- Shape points: 2D points used to model the geometrical shape of the lane, i.e., the shape of the line inside a link. A node is also considered as a shape point.
- Lane segment: a segment $Ls_{k,i} = (p_{k,i}, p_{k,i+1}, l_{k,i})$ attached to a link L_k composed by two consecutive shape points which length is defined as $l_{k,i} = ||p_{k,i+1} - p_{k,i}||$. Every link L_k is an ordered sequence of segments with the first segment $Ls_{k,0}$ having its starting shape point $p_{k,0}$ equal to the starting node of the link L_k and the last segment $Ls_{k,m_{k+1}}$ having its ending shape point $p_{k,m_{k+1}}$ equal to the ending node of the link L_k.

In this paper, we assume that both autonomous and manual vehicles define their paths as a sequence of links $T{=}(L_1, L_2, ...)$, to follow w.r.t. the map. We also suppose that all the vehicles share the same map. Moreover, in our representation, the links typically represent the center of the lanes, i.e. the nominal position of a vehicle inside the lanes. Therefore, it is much more convenient to represent a vehicle pose with curvilinear coordinates instead of Cartesian ones. For a given Cartesian pose $[x,\, y,\, \theta]^T$ we compute the corresponding Lanelets curvilinear pose $[s,\, n,\, \psi]^T$ w.r.t. the associated lane segment. The choice of the Lanelets approach allows us to have a continuous map matching on the curvilinear abscissa. This avoids discontinuous projections which occurs with Euclidean map matching [7].

We compute the curvilinear abscissa s w.r.t. the matched lane segment $Ls_{k,i}$, belonging to the link L_k, according to the following formula:

$$s = \sum_{j=0}^{i-1} l_{k,j} + \lambda l_{k,i}, \tag{1}$$

where $l_{k,j}$ is the length of the j-th segment of the link L_k, λ is the convex combination parameter computed with the Lanelets map matching criterion, under the constraint $p_\lambda = \lambda p_{k,i} + (1-\lambda) p_{k,i+1}$ [4], and the cumulative sum is computed from the length of the lane segment $Ls_{k,0}$, up to the segment $Ls_{k,i-1}$ which precedes the map matched one.

In the case of autonomous vehicles, where the paths of each vehicle are known a priori, the map matching is straightforward, because we reason on a single polyline. In contrast, for manual vehicles, ambiguities occur because several paths may be potential paths, e.g., when a road splits. In order to choose properly the correct road, we consider a map matching criterion that considers not only the lateral distance $|n|$, but also the relative heading $|\psi|$ between the vehicle heading and the abscissa axis of the local frame attached to $Ls_{k,i}$. Figure 1 illustrates both the map representation and the curvilinear pose computation.

III. Virtual Platooning

In this section, we focus our attention on the backbone of our system: the virtual platoon technique [9]. This method allows to compare the distances between two or more vehicles to a common point, even if they are not lying on the same lane. The aim is to establish a total order on the vehicles, based on the inter-distances comparison, in order to find which vehicle has to cross the roundabout first. The algorithm operates as explained in Algorithm 1.

Algorithm 1 Virtual platooning algorithm.

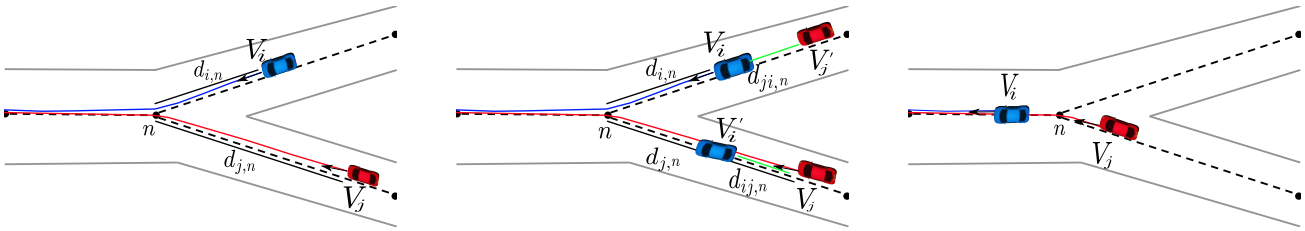
- Find the first common node n (if it exists) between two paths T_i, T_j belonging to two vehicles V_i and V_j, with $i \neq j$.
- Compute the curvilinear distances $d_{i,n}$ and $d_{j,n}$ from the vehicles positions to the common node n.
- Compute the virtual curvilinear inter-distance $d_{ij,n} = d_{i,n} - d_{j,n}$.
- If $d_{ij,n} \geq 0$, then V_i has to cross the intersection after V_j which is the leader, otherwise V_i crosses before V_j.
- Performing platooning control according to the order decided in the previous step.

Once the crossing order is decided, it is possible to use a platooning algorithm to keep each vehicle to a given inter-distance from its leader (virtual or real). In order to compute the curvilinear distance $d_{i,n}$ of a vehicle V_i to a node n with respect to its path $T = (L_1, L_2, ..., L_k, ..., L_n, ...)$, where L_k is the link on which the vehicle is and L_n is the link in which the ending node corresponds to the intersecting node n, we consider the map representation explained in Section II. Let $s_{V_i|L_k}$ be the curvilinear abscissa of the vehicle V_i lying on the link L_k, we can obtain $d_{i,n}$ as follows:

$$d_{i,n} = \sum_{j=k}^{n} \mathcal{L}_j - s_{V_i|L_k}, \tag{2}$$

where $\mathcal{L}_j$ is the length of the link L_j, i.e., the sum of the lengths of its lane segments.

Figure 2 illustrates a typical situation where two vehicles V_i and V_j have to cross an intersection. In this scenario, both V_i and V_j compute their curvilinear distances w.r.t. the common node n. After that, V_j is virtually projected onto the lane of V_i (Fig. 2b), in order to compute $d_{ij,n}$. The same thing happens for V_i. Finally, one can see that a total order between the vehicles has been established according to the sign of $d_{ij,n}$ for V_i and $d_{ji,n}$ for V_j. It is important to note that the order computed by each vehicle is the same for everyone, and the quantity $|d_{ij,n}|$ can be considered as



(a) (b) (c)

Fig. 2: Classical virtual Platooning for crossing a junction. (a) Initial scenario. The paths of vehicles V_i and V_j intersect at node n, with the distance $d_{i,n}$ and $d_{j,n}$ respectively. (b) Virtual platoon of V_j w.r.t. the virtual projection $V_i^{'}$ of V_i onto its own lane with an inter-distance $d_{ji,n}$. (c) Real platoon of V_j w.r.t. the vehicle V_i. Please note that this crossing order does not take into account traffic rules (e.g. stops or give way) as it is only computed w.r.t. the inter-distance between vehicles.

the virtual curvilinear inter-distance between the vehicles V_i and V_j. Note also that this approach works only under the assumption that vehicles can communicate their intended paths to the others.

IV. Roundabout Crossing Algorithm

A. Autonomous Vehicles Case

In this section, we apply the concepts seen before in the case of a roundabout crossing for cooperative autonomous vehicles that exchange their pose and path. Suppose a vehicle V_i needs to cross a roundabout in the presence of other vehicles $V_1,\ V_2, \ldots,\ V_n$. The main idea is to find the leader vehicle V_j, if it exists, w.r.t. which the vehicle V_i needs to do platooning with. V_j is the vehicle with the smallest positive virtual inter-distance to V_i among all the vehicles whose paths crosses the one of V_i. If a vehicle has no leader, it means that there are no other vehicles on its path. Therefore, it can drive at its nominal velocity, without any risk of collision. Otherwise, it regulates its velocity in order to keep a safety inter-distance to its leader vehicle V_j. The same procedure is followed by all the other vehicles. This procedure, described in Algorithm 2, allows all the vehicles to cross the roundabout safely in a distributed manner.

Our approach has several assumptions. First, we suppose that the pose $[x,\ y,\ \theta]^T$ is well estimated, with low uncertainty for all the vehicles. We also assume that there is a perfect communication between all the vehicles without messages loss. Another assumption is that all the paths are known and shared, which is reasonable for cooperative autonomous vehicles, but not for manually driven communicating cars. We present hereafter a method to handle this issue.

Algorithm 2 Roundabout crossing procedure for vehicle V_i.

Require: $[x_i,\ y_i,\ \theta_i], T_i$
$d \leftarrow +\infty$
Send $(x_i,\ y_i,\ \theta_i,\ T_i)$
$[V_1,\ V_2,\ \ldots,\ V_c] \leftarrow$ Receive()
$s_{V_i|L_i} \leftarrow$ CurvAbscissa$([x_i,\ y_i,\ \theta_i],\ T_i)$
for $j = 1 : c$ **do**
  $n \leftarrow$ FindFirstCommonNode(T_i, T_j)
  if $n = \emptyset$ **then**
    Continue (V_j does not cross the path of V_i)
  else
    $s_{V_j|L_j} \leftarrow$ CurvAbscissa$([x_j,\ y_j,\ \theta_j],\ T_j)$
    $d_{ij} \leftarrow$ Interdistance$(s_{V_i|L_i},\ s_{V_j|L_j},\ n,\ T_i,\ T_j)$
    if $d_{ij} < 0$ **then**
      Continue (V_j goes after V_i)
    else
      $d \leftarrow \min(d, d_{ij})$
      $leader \leftarrow V_j$
    end if
  end if
end for
if $leader = \emptyset$ **then**
  Continue (go with nominal velocity)
else
  LongControl$(leader, d)$ (platooning with leader)
end if

B. Mixed traffic with Autonomous and Manually Driven Vehicles

The approach explained previously shows how to handle scenarios in which only autonomous vehicles are involved. In more general contexts, one has to deal with the possible concomitant presence of both autonomous and manually driven vehicles. While it is reasonable to assume that autonomous vehicles can share their intended paths, it is not the case for manually driven vehicles, as they are not known a priori, i.e., the path depends on the driver's intentions. Regarding the manually driven vehicles, we suppose only that their localization w.r.t. the map frame is known with low uncertainty by all the vehicles. This can be done supposing either to have manually driven communicating cars or to have an infrastructure able to localize the vehicles in the environment.

A simple way to extend the approach proposed previously to handle manually driven cars is to make the system aware of them by predicting their unknown paths. We propose a new approach based on replacing a manually driven vehicle V by two virtual copies $V^{'}$ and $V^{''}$. The idea is to assign to each virtual copy of V the two extreme paths w.r.t. the current position of V, that is the first exit branch for $V^{'}$ and the last exit branch (U-turn) for $V^{''}$, as it is shown in Fig. 3a. When a branch occurs, the paths of the two virtual

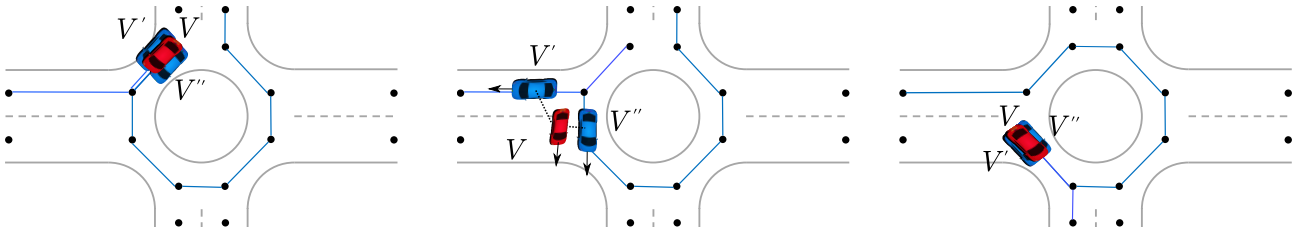


Fig. 3: Mixed scenario. (a) Generation of the virtual vehicles $V^{'}$ and $V^{''}$as copies of the vehicle V and their paths. (b) Branching scenario, the two vehicles $V^{'}$ and $V^{''}$follow their virtual paths until the most unlikely copy w.r.t. the real pose of V is deleted. (c) Recomputing of the virtual copies and their paths according to the new branch.

copies start to diverge from each other. Therefore, the map matching algorithm will assign two different projections for each of them (see Fig. 3b). Once the map matching criterion becomes unlikely regarding to the current pose of V, the virtual copy is deleted and there is no longer ambiguities on the path of the manually driven car. Then, two scenarios can occur:

1) The vehicle V is on an exit branch: in this case, the remaining copy keeps going on this branch until it disappears from the scenario. No new virtual copies need to be introduced.
2) The vehicle V is on an internal branch: in this case, the remaining copy is itself replaced by two new virtual copies updated w.r.t. the new configuration (Fig. 3c).

It is fairly simple to implement this approach by simply replacing the line 3 of Algorithm 2, by the code of Algorithm 3 where the idea is just to manipulate the virtual copies in place of the manual vehicles, and the line 10 of Algorithm 2, with the code of Algorithm 4 to delete the irrelevant copy after a branching.

The threshold gives the limit to which the vehicle can be far from the center of the lane, we suppose that a manually driven vehicle always remains in the neighborhood of the lane (proximity constraint [7]), this will guarantee that at least one of the virtual copies will remain after a branching.

Algorithm 3 Algorithms for virtual copies generation.

1: $[V_1,\ V_2,\ \ldots,\ V_c] \leftarrow$ receive()
2: **for** $j = 1 : c$ **do**
3: **if** isManual(V_j) **then**
4: $[T_j^{'},\ T_j^{''}] \leftarrow$ ExtremeTraj($[x_j,\ y_j,\ \theta_j]$)
5: $[V_j^{'},\ V_j^{''}] \leftarrow$ Copy(V)
6: $V_j^{'} \leftarrow$ UpdateTraj($T_j^{'}$)
7: $V_j^{''} \leftarrow$ UpdateTraj($T_j^{''}$)
8: Add($V_j^{'},\ V_j^{''}$)
9: Remove(V_j)
10: **end if**
11: **end for**

V. Correctness

In order to validate the algorithm, let us prove its correctness, i.e., the absence of unwanted behaviors. In particular, we are interested in proving both the absence of deadlock (all the vehicles are waiting to cross the roundabout) and starvation (one vehicle is waiting indefinitely). To do so, we decompose the problem in two possible situations. In the first one, we consider the case in which it exists a total ordered sequence among the vehicles, and in the second one, where it is not the case. We remind that every vehicle computes its own order sequence based on the current configuration of the other vehicles w.r.t. its own path.

Algorithm 4 Algorithm for virtual copies deletion

1: $[s_{V_j|L_j}, n_{V_j|L_j}, \psi_{V_j|L_j}] \leftarrow$ CurvCoords($[x_j, y_j, \theta_j], T_j$)
2: **if** isVirtual(V_j) AND $f(n_{V_j|L_j},\ \psi_{V_j|L_j}) < Threshold$ **then**
3: Remove(V_j)
4: **end if**

A. Totally ordered case

The first case considers the situation where there exists a totally ordered sequence for the vehicles. This will lead to a nominal virtual platooning scenario, where all the vehicles go through the roundabout one after the other, following the common order. We show that this occurs when, for any pair of paths, there is either no intersecting node or the first intersecting node is unique.

Let $V_1, V_2, \ldots, V_c$ be a set of c vehicles crossing a roundabout and $T_1, T_2, \ldots, T_c$ their respective paths. We suppose that for any pair of paths (T_i, T_j), either $T_i \cap T_j = \emptyset$ or $T_i \sqcap T_j = T_j \sqcap T_i = n$, where $T_i \sqcap T_j$ corresponds to the first common node between T_i and T_j relative to T_i, i.e., w.r.t. the ordered sequence of links of T_i.

We use a proof by contradiction to demonstrate that there exists a total order between the vehicles. Suppose there is no total order, that is there exists two vehicles V_i, V_j for which V_i decides that it goes before V_j:

$$d_{i,n_{ij}} < d_{j,n_{ij}} \quad \text{with} \quad n_{ij} = T_i \sqcap T_j, \tag{3}$$

and V_j also decides to go before V_i:

$$d_{i,n_{ji}} < d_{j,n_{ji}} \quad \text{with} \quad n_{ji} = T_j \sqcap T_i. \tag{4}$$

However, as we have supposed that $T_i \sqcap T_j = T_j \sqcap T_i$, which leads to $n_{ij} = n_{ji}$, the two relations (3) and (4) lead to a contradiction. This concludes the proof.

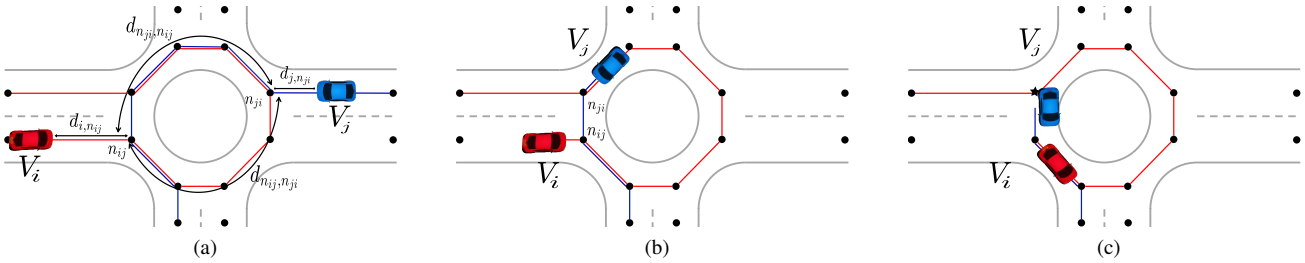


Fig. 4: Dangerous cases: (a) Potential deadlock case with the involved quantities. (b) Both vehicles are leaders at the same time, $T_i \sqcap T_j = n_{ij}$ for V_i and $T_j \sqcap T_i = n_{ji}$ for V_j. (c) V_i becomes leader of V_j as it overcomes n_{ji} and $T_i \sqcap T_j = n_{ij}$ for V_i and $T_j \sqcap T_i = n_{ij}$ for V_j.

B. *Differently ordered case*

We suppose now that the first intersecting node between two paths can be different, i.e., $\exists i, j$ such that $T_i \sqcap T_j = n_{ij}$ and $T_j \sqcap T_i = n_{ji}$ with $n_{ij} \neq n_{ji}$. We want to show that no deadlock can occur even in this case. Let consider a deadlock configuration in which both vehicles are waiting for the other to cross the roundabout. This means that

$$d_{i,n_{ij}} > d_{j,n_{ij}} \qquad \text{and} \qquad d_{j,n_{ji}} > d_{i,n_{ji}}. \tag{5}$$

We want to prove that this configuration cannot occur. Based on Fig. 4a, we can write the following expressions:

$$d_{j,n_{ij}} = d_{j,n_{ji}} + d_{n_{ji},n_{ij}} \text{ and } d_{i,n_{ji}} = d_{i,n_{ij}} + d_{n_{ij},n_{ji}}, \tag{6}$$

where $d_{n_{ji},n_{ij}} > 0$ and $d_{n_{ij},n_{ji}} > 0$ are the curvilinear distances from the node n_{ji} to the node n_{ij} and from the node n_{ij} and n_{ji} according to the flow direction of the roundabout.

If we take the relation $d_{i,n_{ij}} > d_{j,n_{ij}}$ and substitute the corresponding values, we obtain the following inequalities:

$$\begin{aligned} d_{i,n_{ij}} > d_{j,n_{ij}} \quad &\Leftrightarrow \quad d_{i,n_{ji}} - d_{n_{ij},n_{ji}} > d_{j,n_{ji}} + d_{n_{ji},n_{ij}} \\ &\Leftrightarrow \quad d_{i,n_{ji}} > d_{j,n_{ji}} + d_{n_{ji},n_{ij}} + d_{n_{ij},n_{ji}} \\ &\Rightarrow \quad d_{i,n_{ji}} > d_{j,n_{ji}} \end{aligned} \tag{7}$$

This contradicts the deadlock condition (5). The same reasoning can be generalized to any number of vehicles.

We have proved that no deadlock occurs, but there exist some cases in which potentially unsafe configurations can arise. Figure 4b shows a case in which, as it was told before, all the vehicles think to be the leaders, in fact we have $d_{i,n_{ij}} < d_{j,n_{ij}}$ and $d_{j,n_{ji}} < d_{i,n_{ji}}$. Nobody perform control w.r.t. the other, and both the vehicles proceed with their nominal velocities. If the distance between n_{ij} and n_{ji} is not sufficiently large, we can have not enough space to brake V_j when it overcomes the node n_{ji} and becomes the follower of V_i. (Fig. 4c).

VI. Experiments and Results

To show the behavior of the algorithm, we consider different experimental scenarios. Simulations have been carried out considering the map representation of large roundabout in the city of Compiègne (see Fig. 5). This roundabout presents two lanes into the central part and two-lanes roads

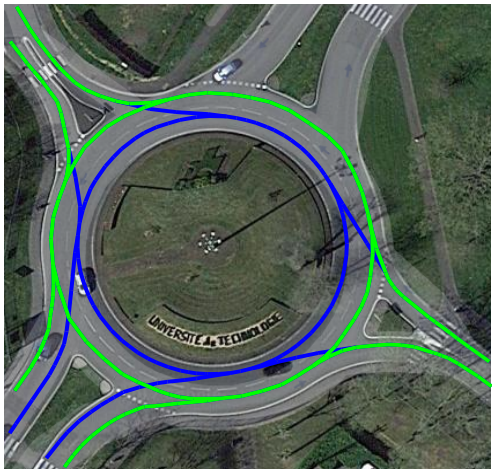


Fig. 5: The "Guy Deniélou" roundabout in the city of Compiègne, France. The green line represents the outer lane of the roundabout while the blue one is the inner one. The north-east exit of the roundabout was not mapped and was ignored in this study.

for some branches. In the simulations, we considered only the outer lane for the roundabout and the right side lane for the branches with multiple lanes which are represented as the green lines in Fig. (5). The generalization to multiple lanes is straightforward by adding intersections nodes where the green and blue lines cross. The original virtual platoon algorithm enables to handle such situations. We also suppose the communications between vehicles to be perfectly reliable and we compute analytically the evolution of the curvilinear abscissa along the road path as a function of the actual curvilinear abscissa and the vehicle velocity instead of using the map matching with real poses. This needs to be done because it is no possible to change dynamically an offline recorded data set, in order to make the data synchronized with the control proposed by the algorithm.

In order to properly control the vehicles during platooning, we use a longitudinal control law based on the inter-distance between the leader vehicle (virtual or real) and the followers which implements the law. In our case, the wanted curvilinear inter-distance is a linear function of the velocity v of a vehicle in the following form:

$$d = d_0 + h \cdot v, \tag{8}$$

where d_0 is the standstill distance and h is the time headway. To do the feedback, we use a proportional-derivative control law computing the acceleration of the follower as follows

$$a(t) = \alpha_1(s_l(t) - d - s_f(t)) + \alpha_2(\dot{s}_l(t) - \dot{s}_f(t)), \quad (9)$$

where α_1 and α_2 are two gains, $s_l(t)$ and $s_f(t)$ the curvilinear abscissa of the leader and the follower vehicles, and $\dot{s}_l(t)$, $\dot{s}_f(t)$ their corresponding velocities. Saturation functions in the form $v(t) = \max(v_{max}, v(t))$ and $v(t) = \min(v_{min}, v(t))$ are considered to bound the velocity into fixed limits. Acceleration saturations have also implemented.

A. Case study: Safety diagram

Let introduce a new concept called "Safety diagram" in order to illustrate the behaviors of the vehicles in the scenario. The aim of this diagram is to show when safety margins are violated, i.e., when the curvilinear inter-distance w.r.t. the first non-virtual obstacle along a vehicle path is too short given its current velocity. Consider a diagram as the one in Fig. 6, where the inter-distance is plotted as a function of the velocity of the vehicle. We can see that Eq. (8) draws a boundary between the safe and unsafe zones. If a point (v, d) is above that line, it means that the inter-distance d is sufficiently large for the velocity v. Inversely, if the point (v, d) is found under the boundary, it means that d is too short for the vehicle velocity v, i.e., an unsafe case occurs. Finally, if (v, d) lies on the boundary, it means that the vehicle is perfectly at the desired curvilinear inter-distance in the platoon w.r.t. its velocity v.

To illustrates a typical scenario that occurs in our algorithm with the corresponding safety diagram interpretation, let us consider a leader vehicle driving at a constant velocity v_l and a follower in virtual platoon w.r.t. the leader. Figure 6 shows the behavior of the follower that is entering the roundabout at time $t = t_0$ and finds a leader vehicle ahead of him. From t_0 to t_1 the follower accelerates to reach the desired inter-distance from its leader, until the speed saturation occurs at t_1. Then from t_1 to t_2, it continues to move at the maximum allowed speed v_{max}, decreasing the inter-distance until t_2 when it starts braking as it catches the leader. Finally, at t_3, the follower reaches the optimal inter-distance w.r.t. the leader and continues driving at the same velocity as the leader.

B. Monte Carlo simulation

To test our approach, we developed a simulator able to generate random vehicles over time, with stochastic trajectories. The simulator implements the algorithm described in section IV, allowing us to choose the frequency for generating both autonomous and manually driven vehicles, in order to cover every possible scenarios. Figure 7 shows the resulting safety diagram for a given simulation with a dozen vehicles mixing both autonomous and manually driven ones. One can see that all the trajectories tends to converge towards the safety boundary which is the optimal inter-distance. The

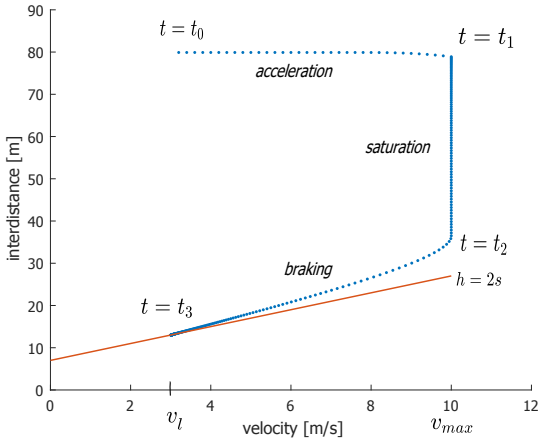


Fig. 6: Safety diagram illustration. The blue dots represent the (v, d) values of the following vehicles w.r.t. the leader with velocity $v_l = 3m/s$ and $v_{max} = 10m/s$. The orange line represents the safety boundary with $d_0 = 7m$ and $h = 2s$.

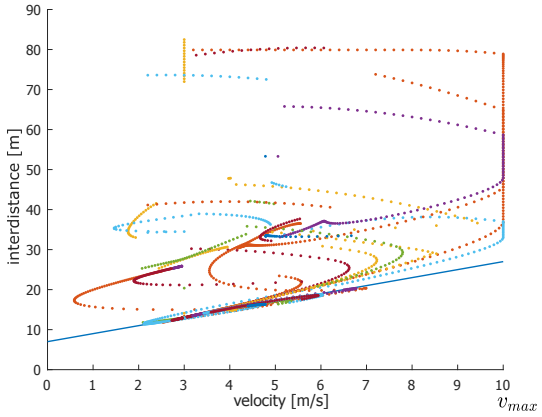


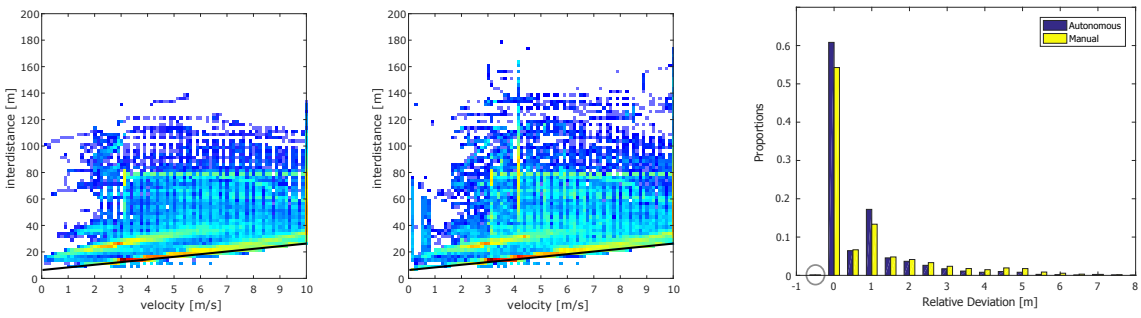
Fig. 7: Safety diagram for a stochastic simulation with a dozen vehicles.

control law may allow the vehicle to oscillate around the optimal inter-distance, we can see that it does not deviate significantly under the safety boundary.

In order to present an exhaustive analysis of this algorithm, we have implemented a Monte Carlo simulation. The aim is to extend the analysis to cover as many cases as possible, in order to study statistically the performances of the algorithm. We report the results of two Monte Carlo simulations:

1) autonomous vehicles only; (Fig. 8a)
2) autonomous and manually driven vehicles. (Fig. 8b)

In figures 8a and 8b, we display the empiric joint probability density function of the velocity and the corresponding inter-distance using a logarithmic scale. One can see that most of the points are located above the safety boundary. However, there are also a few points in the unsafe region. Several reasons can explain this behavior. First, the imperfect tuning of the controller can lead the vehicles to have oscillations along the optimal boundary. Second, the differently ordered case presented in section V-B also allows vehicles to have their inter-distances lower than the optimal ones. Figure 8c shows the distribution of the relative deviations



(a) Autonomous vehicles only.

(b) Autonomous and manually driven vehicles.

(c)

Fig. 8: (a-b) Empiric joint log-probability density function after a Monte Carlo simulation with 100 iterations. (c) Distribution of the relative deviation between the actual and optimal inter-distance. The circle depicts the unsafe region with a proportion of about 1%.

from the optimal inter-distance computed as follows:

$$e_r = \frac{d_v - (d_0 + hv)}{d_0 + hv} = \frac{d_v}{d_0 + hv} - 1, \qquad (10)$$

where d_v is the actual distance w.r.t. the first non virtual obstacle. Moreover, we have computed the proportion of points that deviate more than the 5% into the unsafe region, and we have found out that this percentage is in the order of 1%.

From these figures, one can see that the algorithm performed similarly in both cases. The autonomous only scenario allows the vehicles to drive with inter-distances closer to the optimal ones.

VII. Conclusion

This work has shown an adaptation of the virtual platooning concept to the roundabout crossing problem. This idea has the advantage to be easily implementable in a embedded system that exploits a map based approach. This work has also shown the importance of exploiting a map to model a roundabout since all the calculations are done in a curvilinear framework. As proven, the algorithm has no deadlock. We have also proposed a strategy to handle situations where both autonomous and manually driven vehicles are involved. Finally, a new safety diagram has been introduced to study the algorithm performances w.r.t. a given safety bound.

The approach described in this paper has been studied in the case of single lane roundabouts but can be generalized to multi-lanes scenarios by combining our method with the original intersection crossing virtual platoon.

Based on the obtained results, we can state that the algorithm has promising performances. Future perspective is to adapt this model to a real-world environment, in order to take into account the uncertainty aspects in the localization process, the possibility of an imperfect communication and the issues linked to the detection and tracking of uncertain objects. Traffic regulations have also to be added to the framework to make this approach effective.

Acknowledgment: This work was carried out in the framework of Equipex ROBOTEX (ANR-10- EQPX-44-01) and Labex MS2T (ANR-11-IDEX-0004-02). It was also carried out within SIVALab, a shared laboratory between Renault and Heudiasyc CNRS/UTC, through the TORNADO project.

References

[1] Retting Richard A., Persaud Bhagwant N., Garder Per E., and Lord Dominique. Crash and injury reduction following installation of roundabouts in the united states. *American Journal of Public Health*, 2001.

[2] Javier Alonso, Vicente Milanés, Joshué Pérez, Enrique Onieva, Carlos González, and Teresa de Pedro. Autonomous vehicle control systems for safe crossroads. *Transportation Research Part C pp.1095-1110*, 2011.

[3] Lejla Banjanovic-Mehmedovic, Edin Halilovic, Ivan Bosankic, Mehmed Kantardzic, and Suad Kasapovic. Autonomous vehicle-to-vehicle (v2v) decision making in roundabout using game theory. *(IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 7, No. 8*, 2016.

[4] P. Bender, J. Ziegler, and C. Stiller. Lanelets: Efficient map representation for autonomous driving. *In IEEE Intelligent Vehicles Symposium Proceedings, pp.420-425*, 2014.

[5] Alessandro Colombo and Domitilla Del Vecchio. Efficient algorithms for collision avoidance at intersections. *In Proceedings of the 15th ACM HSCC, pp. 145–154, New York, USA, 2012. ACM.*, 2012.

[6] 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. *2009 IEEE International Conference on Robotics and Automation Kobe International Conference Center Kobe, Japan pp.82-87*, May 12-17, 2009.

[7] Elwan Héry, Stefano Masi, Philippe Xu, and Philippe Bonnifait. Map-based curvilinear coordinates for autonomous vehicles. *IEEE 20th International Conference on Intelligent Transportation Systems pp.1-7*, 2017.

[8] L. Li and F. Y. Wang. Cooperative driving at blind crossings using intervehicle communication. *IEEE Transactions on Vehicular Technology, vol. 55, no. 6, pp.1712-1724*, 2006.

[9] Alejandro Ivan Morales Medina, Nathan Van de Wouw, and Henk Nijmeijer. Automation of a t-intersection using platoons of cooperative autonomous vehicles. *IEEE 18th International Conference on Intelligent Transportation Systems, pp.1696-1701*, 2015.

[10] Joshué Pérez Rastelli, Vicente Milanés, Teresa De Pedro, and Ljubo Vlacic. Autonomous driving manoeuvres in urban road traffic environment: a study on roundabouts. *IFAC Proceedings Volumes, pp.13795-13800*, 2012.

[11] Joshué Pérez Rastelli and Matilde Santos Peñas. Fuzzy logic steering control of autonomous vehicles inside roundabouts. *Applird Soft Computing Volume 35, pp.662-669*, 35, 2015.