Automation of a T-intersection using virtual platoons of cooperative autonomous vehicles

Alejandro Ivan Morales Medina, Nathan van de Wouw, Henk Nijmeijer

Abstract—Both traffic throughput and the vehicle passenger safety can be increased by automating road intersections. We propose the virtual platooning concept to ensure a smooth, efficient and safe traffic flow through an automated intersection. The virtual platoon is formed by defining a virtual intervehicle distance between vehicles driving on different lanes. Such distance is employed by a Cooperative Adaptive Cruise Control (CACC) system which, in turn, generates the required safe "gaps" for the vehicles to cross the intersection. A simulation study demonstrates the functionality of the presented methodology, which is referred to as Cooperative Intersection Control (CIC).

I. INTRODUCTION

Road intersections represent a potential hazard to road users since they are the place in which trajectories of vehicles cross. So, it is not surprising that 36% of collisions happen at road intersections [1]. Current methods to regulate the flow of vehicles through intersections are: stop signs, roundabouts and traffic lights. These methods require the vehicles to stand still and wait for their turn to cross the intersection or to enter the roundabout. This fact obstructs the flow of vehicles, which in turn limits the throughput of the intersection. Note, in this respect, that the amount of time that the vehicles stand still increases with traffic congestion.

The crossing trajectories of the vehicles competing for access to the intersection generate "conflict zones" in the intersection, where the vehicles would collide if they entered these zones at the same time. Therefore, we can define the intersection problem as finding a crossing sequence such that the vehicles cross the intersection in a safe manner by avoiding such conflict situations. In the case of the current regulatory methods for intersection access control, the conflict zones are avoided by only allowing vehicles on the intersection that do not have crossing trajectories, such as all the vehicles in one road with straight trajectories. By automating the vehicular traffic flow through the intersection, we envision to increase the throughput of the intersection by minimizing the time that the vehicles stand still. We aim to achieve this goal by more frequently alternating between giving access (priority) to different lanes, while ensuring at the same time the safe passage of vehicles using cooperative vehicular control methods.

Existing approaches for automating an intersection can be categorized as being either centralized or decentralized. Centralized solutions keep the same approach as the traffic lights but include V2I (Vehicle to Infrastructure) communication. Such solutions typically involve an intersection agent that receives requests from vehicles to cross the intersection and decides on the best crossing sequence. There are several approaches for the underlying decision algorithm, such as: vehicle trajectory prediction to identify conflict situations [2], modeling the intersection as an scheduling problem [3], and the analysis of the so-called conflict zone plots to determine a priority map [4].

The above centralized solutions allow to use optimization algorithms to achieve a safe and optimal crossing sequence; for instance: optimization by minimizing the total travel time in the intersection [10], [11], or optimization by minimizing each lane queue length [12]. A drawback of such centralized solutions is that a Road Side Unit (RSU) is needed to manage the crossing sequence (it can be costly to equip every intersection with an RSU).

Existing decentralized solutions rely on vehicle cooperation, using V2V (Vehicle to Vehicle) communication, to determine a safe crossing sequence. These solutions focus on the interaction protocols between a pair of vehicles. In [5], a family of decentralized dead-lock free protocols is presented, while [6] defines a capture set (using the aforementioned conflict zones plots) to define which vehicle crosses first. In [7], a token-based protocol is used to allow vehicles to cross the conflict zones, and [8], [9] use virtual vehicles (which are a projections of vehicles onto other roads) and an inter-vehicle distance controller to determine which vehicle crosses first.

The above decentralized solutions benefit from V2V communication to perform cooperative maneuvers, that allow the vehicles to cross the intersection without having to predict the future behavior of the vehicles while crossing the intersection. A drawback of decentralized solutions is the difficulty of determining an optimized crossing sequence.

The approaches for intersection control discussed above typically focus on high level scheduling of the crossing sequence of vehicles through the intersection while paying little attention to the dynamical behavior of the vehicles while performing the maneuvers through the intersection; vehicles are only instructed to either accelerate, brake, or stand-still. One way to account for the dynamics of the vehicles is to use

A. I. Morales Medina, N. van de Wouw, and H. Nijmeijer are with the Department of Mechanical Engineering, Eindhoven University of Technology, the Netherlands. { a.i.morales.medina@tue.nl; n.v.d.wouw@tue.nl; h.nijmeijer@tue.nl}

N. van de Wouw is also affiliated with the Civil, Environmental & Geo-Engineering Department, University of Minnesota, U.S.A. and with the Delft Center for Systems and Control, Delft University of Technology, The Netherlands.

Research supported by the European Union Seventh Framework Programme [FP7-ICT-2013-10] under grant agreement no. 612035. Interoperable GCDC (Grand Cooperative Driving Challenge) AutoMation Experience (i-GAME).

cooperative driving techniques, such as Cooperative Adaptive Cruise Control (CACC) [16]. This allows an automated vehicle either to follow a reference velocity (referred as Cruise Control (CC)), or to realize a reference inter-vehicle distance. The centralized solutions in [13] and [14], achieve a safe crossing by calculating an optimized velocity profile for the vehicles to follow using the CC functionality.

The current paper presents a decentralized solution that achieves a safe crossing sequence through a T-intersection by means of virtual platoons of cooperative autonomous vehicles. A virtual platoon is formed by defining a virtual intervehicle distance between vehicles driving on different lanes and realizing the virtual inter-vehicle distance by CACC on the individual vehicles. We refer to this cooperative control strategy as Virtual CACC (VCACC). The difference with respect to the work in [8], [9] is that rather than mapping vehicles onto different lanes, we use the concept of a (scaled) traveled distance through the intersection to define the socalled virtual inter-vehicle distance. This approach allows us to leave the "gap-making" task (i.e. to ensure safe passage of a vehicle with higher priority) solely in the hands of the CACC system.

The main contribution of this work is the transformation of the intersection problem into a virtual platooning problem for which we design cooperative control strategies (i.e. VCACC) that can achieve a smooth and efficient traffic flow through the intersection.

The presented solution is referred to as Cooperative Intersection Control (CIC) and schematically represented in Figure 1. It is divided into two levels: a supervisory level that manages the formation of virtual platoons of vehicles approaching the intersection, and an execution level that takes care of the coordinate transformations needed to define the virtual gaps between vehicles on different lanes, and implements the cooperative virtual platooning. This work focuses on the execution level, so just a general explanation of the supervisory level will be given.

The outline of the paper is as follows. Section II concisely presents the T-intersection problem description. In Section III, we will give a concise description of the supervisory control level. Section IV focuses on the execution control level and details the calculation of the virtual inter-vehicle distance and the control methodologies used to automate the cooperative autonomous vehicles. In Section V, we present a simulation that exemplifies the virtual platooning concept. Finally, this paper is concluded in Section VI.

II. T-INTERSECTION PROBLEM DESCRIPTION

Consider the T-intersection depicted in Figure 2. For each lane (1, 2, and 3) two trajectories are possible: from lane 1 we can go either left $(t_{1,l})$ or right $(t_{1,r})$, from lane 2 we can go either straight $(t_{2,s})$ or right $(t_{2,r})$, and from lane 3 we can go either straight $(t_{3,s})$ or left $(t_{3,l})$. So any vehicle can have either straight (s), left (l), or right (r) intention (denoted by the intention variable $\eta \in \{s, l, r\}$) depending on the lane



Fig. 1. Control system architecture.



Fig. 2. T-intersection with possible vehicle trajectories.

(with index $k \in \{1, 2, 3\}$) on which the vehicle entered the socalled Cooperation Zone (CZ). Hence, any vehicle will follow a trajectory $t_{k,\eta}$ through the intersection.

We define a pair of trajectories as crossing trajectories if they share at least one point (see e.g. the trajectory pairs $t_{1,l}$ and $t_{3,s}$, or $t_{1,l}$ and $t_{3,l}$ in Figure 2), or as non-crossing trajectories if they do not share any point (e.g. $t_{1,r}$ and $t_{2,r}$). A set of automated vehicles can cross the intersection without cooperation if all their trajectories are non-crossing; cooperation is needed for the cases in which two vehicles have crossing trajectories. Therefore, the goal of cooperation is to ensure that two or more vehicles, with crossing trajectories, will be at a safe distance from each other, while accommodating a smooth passage through the intersection.

III. SUPERVISORY LEVEL

Figure 1 shows a representation of the architecture of the control system that will be used to solve the cooperative intersection problem. The supervisory level consists of two subsystems: the Target Vehicle Assignment (TVA), and the Control Reconfiguration (CR). Since this paper focuses on the execution level we will give a concise description of the aforementioned subsystems.

A. Target vehicle assignment

Every vehicle that enters the CZ is assigned with a vehicle counter m (determined in a first come first serve fashion), a lane of entering k_m , and an intention η_m . Using this information we can determine the virtual platoon index given by $i_m = m - f_m + 1$ designating the place of a vehicle in a virtual platoon (the meaning of f_m will be explained below). In other words, we assign the vehicle $m-f_m$ as the target vehicle of vehicle m. The variable f_m is determined as follows: first $f_m = 1$; subsequently f_m is updated as $f_m := f_m + 1$ in every iteration until the vehicle $m-f_m$ has a crossing trajectory with the vehicle m (this is determined by comparing the lane and intention of both vehicles). If two vehicles enter the CZ at the same time, the virtual platoon index i_m is first assigned to the vehicle on the highest priority lane (without loss of generality, we choose the lane priority the same as the lane number).

B. Control reconfiguration

The vehicle has several control modes for its longitudinal movement such as: Collision Avoidance Control which stops the vehicle if it is too close to surrounding objects, Cruise Control (CC) which is a velocity controller, and Cooperative Adaptive Cruise Control (CACC) which is an inter-vehicle distance controller. When the vehicles enter the CZ, a control reconfiguration is commanded depending on the assigned target vehicle. To achieve a smooth and comfortable transition between control modes we use the control reconfiguration with mixing methodology as presented in [15].

IV. EXECUTION LEVEL

This section presents the details behind the calculation of the virtual inter-vehicle distance which is the key concept behind the VCACC approach that aims to automate the T-intersection. Additionally, this section describes the vehicle dynamics and controller design.

A. Virtual inter-vehicle distance

We start by defining the possible trajectories to cross the T-intersection. Figure 3 depicts the geometry of the Tintersection, where w_p is the width of the principal road, w_s is the width of the secondary road, r_{cz} is the radius of the circle that defines the boundary of the CZ, and $S^0 = \{O^0, \vec{e}^0\}$ is the Intersection Reference Frame (IRF). Note that the secondary road is perpendicular to the principal road¹. The origin O^0 of the IRF is located at the intersection between the principal road middle line and the secondary road middle line. The orthonormal basis of the IRF is $\underline{\vec{e}}^{0} = \begin{bmatrix} \vec{e}_{x}^{0} & \vec{e}_{y}^{0} & \vec{e}_{z}^{0} \end{bmatrix}^{T}$, where $\vec{e}_{z}^{0} = \vec{e}_{x}^{0} \times \vec{e}_{y}^{0}$ (pointing out of the page) and its orientation is such that the unitary vector \vec{e}_x^0 is parallel to the principal road.

Each vehicle m is assigned with a set of two frames $S_m =$ $\{S^{k_m}, S'^m\}$, where k_m is the lane on which the vehicle mentered the CZ; S^{k_m} is a stationary frame, and S'^m is a bodyfixed frame. As an example, consider the set of three vehicles $\{V_m | m \in \{1, 2, 3\}\}$, depicted in Figure 3. The vehicle V_1 entered the CZ on the lane $k_1 = 2$ so it is assigned with the set of frames $S_1 = \{S^2, S'^1\}$. The assignment of the set of frames for vehicles V_2 and V_3 follow the same logic so,





Fig. 3. T-intersection geometry.

 V_2 is assigned with $S_2 = \{S^3, S'^2\}$, and V_3 is assigned with $S_3 = \{S^1, S'^3\}.$

The stationary frame of lane k is defined as $S^k = \{O^k, \vec{e}^k\},\$ where O^k is the frame's origin², that is the point in which each vehicle enters the CZ, and $\underline{\vec{e}}^{k}$ is the frame's basis. The position of the frame's origin with respect to the origin of S^0 is given

by $\vec{r}_{O^k} = \underline{r}_{O^k}^{0^T} \vec{e}^0$, from which we can determine the associated coordinates as $\underline{r}_{O^k}^{0^T} = \vec{r}_{O^k} \cdot \vec{e}^{0^T} =: \begin{bmatrix} x_{O^k}^0 & y_{O^k}^0 & z_{O^k}^0 \end{bmatrix}$. So, we have that $\underline{r}_{O^1}^{0^T} = \begin{bmatrix} \frac{1}{4}w_s & -r_{cz} & 0 \end{bmatrix}, \underline{r}_{O^2}^{0^T} = \begin{bmatrix} -r_{cz} & -\frac{1}{4}w_p & 0 \end{bmatrix}$, and $\underline{r}_{O^3}^{0^T} = \begin{bmatrix} r_{cz} & \frac{1}{4}w_p & 0 \end{bmatrix}$. Also, $\vec{e}^{-1} = \begin{bmatrix} \vec{e}_y^0 & -\vec{e}_x^0 & \vec{e}_z^0 \end{bmatrix}^T$, $\vec{e}^{-2} = \begin{bmatrix} \vec{e}_x^0 & \vec{e}_y^0 & \vec{e}_z^0 \end{bmatrix}^T$, and $\vec{e}^{-3} = \begin{bmatrix} -\vec{e}_x^0 & -\vec{e}_y^0 & \vec{e}_z^0 \end{bmatrix}^T$, see Figure 3.

The body-fixed frame which is attached to vehicle m when it enters the CZ is defined as $S'^m = \{O'^m, \vec{\underline{b}}^m\}$, where O'^m is the frame's origin with initial condition $O'^m(t_m) = O^{k_m}$, where t_m is the time at which V_m enters the CZ, and $\vec{b}^{''}$ is the frame's basis. The position of the origin of the frame S'^m with respect to the origin of the frame S^{k_m} is given by $\vec{r}_{O'^m/O^{k_m}} = \underline{r}_{O'^m/O^{k_m}}^{k_m^m} \vec{e}^{k_m}$, from which we can determine the coordinates as

$$\underline{r}_{O'^{m}/O^{k_{m}}}^{r} = \vec{r}_{O'^{m}/O^{k_{m}}} \cdot \underline{\vec{e}}^{k_{m}^{T}} =: \begin{bmatrix} x_{m} & y_{m} & z_{m} \end{bmatrix}.$$
(1)

The frame $\underline{\vec{b}}^m$ is given by $\underline{\vec{b}}^m = R^T(\theta_m)\underline{\vec{e}}^{k_m}$, where θ_m is the rotation angle between $\overline{\vec{b}}^m_x$ and $\overline{\vec{e}}^{k_m}_x$, and $R(\cdot)$ is the rotation matrix associated with a rotation about $\vec{e}_z^{k_m}$.

Let s_m be a path coordinate of the trajectory t_{k_m,η_m} (associated with vehicle m). It can be defined using the coordinates of the position vector $\vec{r}_{s_m/O^{k_m}}$ (with respect to the frame S^{k_m}) defined as

$$r_{s_m/O^{k_m}}^{k_m^T} =: \begin{bmatrix} x_{s,m} & y_{s,m} & z_{s,m} \end{bmatrix}.$$
 (2)

Straight trajectory. In this case, the path coordinate is simply defined as

$$s_m = x_{s,m}.\tag{3}$$

²Note that the origins O^k , $\forall k \in \{1, 2, 3\}$, are located at the intersection point between the lines perpendicular to the road (which delimit the CZ) and the middle line of each lane, rather than at the circumference of the CZ with radius r_{cz} .



Fig. 4. Left and right trajectories for lane 1.

Left-turn trajectory. In this case, the path coordinate s_m is a curvilinear coordinate. Figure 4 shows an example of a leftturn trajectory $(t_{1,l})$. Generically a left-turn trajectory consists of two straight sections with lengths $s_o = r_{cz} - \frac{1}{2}w_f$, and $s_l = r_{cz} - r_l + \frac{1}{4}w_o$, where w_o is the width of the road on which the vehicle enters the CZ, and w_f is the width of the road on which the vehicle exits the CZ; and a left turn about an angle of $\pi/2$ radians with constant radius $r_l = \frac{3}{4}w_f$. Note that if the vehicle enters the CZ on lane 1, then $w_o = w_s$, and $w_f = w_p$. On the other hand, if the vehicle enters on lane 3, then $w_o = w_p$, and $w_f = w_s$.

Now we can define the curvilinear path coordinate s_m as a function of the coordinates in (2) as follows:

$$s_{m} = \begin{cases} x_{s,m} & x_{s,m} \le s_{o}, \quad y_{s,m} = 0\\ s_{o} + \psi_{l}(x_{s,m}, y_{s,m})r_{l} & x_{s,m} > s_{o}, \quad y_{s,m} \le r_{l} \\ s_{o} + \pi r_{l}/2 + y_{s,m} - r_{l} & y_{s,m} > r_{l} \end{cases}$$
where $\psi_{l}(x_{s,m}, u_{s,m}) = \arctan\left(\frac{x_{s,m} - s_{o}}{2}\right)$

$$(4)$$

where $\psi_l(x_{s,m}, y_{s,m})$ $\operatorname{arctan}\left(\frac{\overline{r_l-y_{s,m}}}{}\right)$

Right-turn trajectory. In this case, the path coordinate s_m also is a curvilinear coordinate. Figure 4 shows an example of a right-turn trajectory $(t_{1,r})$. Generically a right-turn trajectory consists of two straight sections with lengths $s_o = r_{cz} - \frac{1}{2}w_f$, and $s_r = r_{cz} - r_r - \frac{1}{4}w_o$, where w_o is the width of the road on which the vehicle enters the CZ, and w_f is the width of the road on which the vehicle exits the CZ; and a right turn about an angle of $\pi/2$ radians with constant radius $r_r = \frac{1}{4}w_f$. Note that if the vehicle enters the CZ on lane 1, then $w_o = w_s$, and $w_f = w_p$. On the other hand, if the vehicle enters on lane 2, then $w_o = w_p$, and $w_f = w_s$.

Now we can define the curvilinear path coordinate s_k as a function of (2) as follows:

$$s_{m} = \begin{cases} x_{s,m} & x_{s,m} \leq s_{o}, \quad y_{s,m} = 0\\ s_{o} + \psi_{r}(x_{s,m}, y_{s,m})r_{r} & x_{s,m} > s_{o}, \quad y_{s,m} > -r_{r}\\ s_{o} + \pi r_{r}/2 - y_{s,m} - r_{r} & y_{s,m} \leq -r_{r} \end{cases}$$
(5)

where $\psi_r(x_{s,m}, y_{s,m}) = \arctan\left(\frac{x_{s,m} - s_o}{r_r + y_{s,m}}\right)$.

Let us now express the total traveled distance along these type of trajectories in terms of the parameters of the intersection geometry. Let D_p be the total traveled distance, inside the Cooperation Zone, of a vehicle traveling along the principal

road with intention to go straight; and D_s be the total traveled distance of a vehicle, either on the principal or secondary road, taking a left or right turn (so D_s can take different values depending on the type of turn taken). Then, we have that

$$D_p = 2r_{cz},\tag{6}$$

and

$$D_s = s_o + c + s_f \tag{7}$$

where $s_o = r_{cz} - a$, $a = \frac{1}{2}w_f$, $c \in \{\pi r_l/2, \pi r_r/2\}$, $s_f = r_{cz} - b$, $b \in \{r_l - \frac{1}{4}w_o, r_r + \frac{1}{4}w_o\}$. By calculating $s_o + s_f = 2r_{cz} - a - b \Rightarrow 2r_{cz} = s_o + a + a$

 $b + s_f$, and using (6) to obtain

$$D_p = s_o + (a+b) + s_f$$
 (8)

we observe (by comparing (7) and (8)) that generically $D_s \neq D_p$ due to the fact that generically $c \neq a + b$. In other words, the distance traveled by a vehicle following a straight trajectory is not equal to the distance traveled by a vehicle taking a turn. Hence, comparing both traveled distances we introduce a mapping $T(\cdot)$ that defines a scaled distance $s'_m = T(s_m)$ such that

$$T(D_s) = D_p. (9)$$

In defining $T(\cdot)$ s_m is only scaled before a vehicle has finished taking a turn ($s_m \leq s_o + c$). This leads to the following scaling of the path coordinate s_m , while complying with the requirement in (9):

$$T(s_m) = \begin{cases} \left(\frac{s_o + a + b}{s_o + c}\right) s_m & s_m \le s_o + c\\ s_m + (a + b) - c & s_m > s_o + c \end{cases}$$
(10)

The proposed scaling in the traveled distance in (10) will have an effect in the velocity $v_m = \dot{s}_m$ and acceleration $a_m =$ \ddot{s}_m . Therefore, the scaled velocity is given by $v_m' = \dot{s}_m'$ and the scaled acceleration is given by $a'_m = \ddot{s}'_m$.

Above, we have defined a consistent set of path coordinates in the sense that the total traveled distance of a vehicle through the intersection after scaling now is always D_p . Next, we can define a virtual inter-vehicle distance as the difference between these scaled path coordinates. For instance, let a vehicle with the intention to take a turn, with path coordinate s_i , be the target vehicle; also, let a vehicle with the intention to go straight, with path coordinate s_i , be the host vehicle. Note that we will require the host vehicle to follow the target vehicle at a certain desired virtual inter-vehicle distance. The virtual inter-vehicle distance δ_j between the target and host vehicle is now defined as $\delta_j = s'_i - s_j$, where $s'_i = T(s_i)$. The same logic applies to any pair of vehicles driving on any pair of lanes.

B. Host vehicle dynamics

In support of controller design (see Section IV-C) based on the virtual inter-vehicle distance (see Section IV-A) we will combine two vehicle models: a linearized model for the longitudinal behavior and a nonlinear unicycle model.

We will use the longitudinal linearized model presented in [16] which is given by³

$$\delta_m = v_{m-f_m} - v_m$$

$$\dot{v}_m = a_m$$

$$\dot{a}_m = -\frac{1}{\tau}a_m + \frac{1}{\tau}u_{x,m},$$
(11)

where $\delta_m = s_{m-f_m} - s_m$ is the virtual inter-vehicle distance, s_{m-f_m} is the target vehicle path coordinate, s_m is the host vehicle path coordinate. Moreover, v_{m-f_m} is the target vehicle velocity, v_m is the host vehicle velocity, a_m is the host vehicle acceleration, τ is a time constant related to the vehicle's driveline dynamics, and $u_{x,m}$ is the acceleration input to the host vehicle.

The nonlinear unicycle model with respect to a Frénet frame is given by (see [17] for details)

$$\dot{s}_m = \frac{v_m}{1 - d_m c(s_m)} \cos \theta_{e,m}$$

$$\dot{d}_m = v_m \sin \theta_{e,m}$$

$$\dot{\theta}_{e,m} = u_{y,m} - \dot{s}_m c(s_m)$$
(12)

where s_m is the path coordinate of the point $\{x_m, y_m\}$ (defined in (1)) obtained by projecting it orthogonally on t_{k_m,η_m} . Moreover, $c(s_m)$ is the curvature of t_{k_m,η_m} at s_m, d_m is the distance from $\{x_m, y_m\}$ to t_{k_m,η_m} , $\theta_{e,m} = \theta_m - \theta_{s_m}$ where θ_m is the rotation angle between b_x^m and $e_x^{k_m}$, θ_{s_m} is the angle of the tangent line at s_m with respect of $e_x^{k_m}$, and $u_{u,m}$ is the yaw rate input.

C. Controller design

1) Longitudinal control: The cooperative autonomous vehicles have three different longitudinal control modes that are mixed together to define the longitudinal control input $u_{x,m}$.

Collision Avoidance Control. Its objective is to stop the vehicle if it gets too close to any object (other vehicles or standing objects). This controller is needed for safety reasons and, for this work, it is sufficient to assume that such controller is implemented and works properly. The control signal associated to this control mode is given by $u_{1,m}$.

Cruise Control. The objective of this control mode is to follow a reference velocity. The CC control law is given by

$$u_{2,m} = k_{cc}(v_m^{cc} - v_m) + a_m^{cc}, \tag{13}$$

where $k_{cc} > 0$ is a design constant, v_m^{cc} is the reference velocity, and a_m^{cc} is a feed-forward acceleration.

Virtual Cooperative Adaptive Cruise Control. The objective of this control mode is to follow a reference inter-vehicle distance that ensures that the virtual inter-vehicle distance δ_m converges to the desired distance $\delta_m^{vcacc} = r + hv_m$, where r is the stand-still inter-vehicle distance, and h is the headway time. The VCACC control law is given by

$$\dot{u}_{3,m} = h^{-1} [-u_{3,m} + a_{m-f_m} + k_p (\delta_m - r - hv_m) + k_d (\dot{\delta}_m - ha_m)],$$
(14)

³For the sake of simplicity, we present here models describing a pair of vehicles with straight trajectories. For vehicles with curved trajectories, the scaled kinematics have to be used.

where h is the headway time, k_p and k_d are design constants, r is the stand-still inter-vehicle distance, and a_{m-f_m} is the target vehicle acceleration. This controller yields a stable closed-loop for $k_p > 0$, $k_d > 0$, and $k_d > k_p \tau$. The details of this control law can be found in [16].

Using the mixing methodology in [15] the longitudinal input is given by $u_{x,m} = \sum_{j=1}^{3} \beta_{j,m} u_{j,m}$, where $\beta_{j,m} > 0$ are time varying the mixing signals, and $\sum_{j=1}^{3} \beta_{j,m} = 1$.

2) Lateral control: To design a suitable lateral control law we first need to apply the change of coordinates $(s_m, d_m, \theta_{e,m}, u_{x,m}, u_{y,m}) \mapsto (z_1, z_2, z_3, v_{x,m}, v_{y,m})$ defined by

$$(z_1, z_2, z_3) = (s_m, d_m, [1 - d_m c(s_m)] \tan \theta_{e,m})$$

($v_{x,m}, v_{y,m}$) = (\dot{z}_1, \dot{z}_3), (15)

which transforms the model in (12) of a unicycle-type vehicle into chained form [17]. Then, the lateral control law is given by

$$v_{y,m} = -|v_{x,m}|k_0 \int_0^t v_{x,m} z_2 - v_{x,m} k_2 z_2 - |v_{x,m}|k_3 z_3,$$
(16)

where $k_0, k_2, k_3 > 0$. Note that this controller ensures that the vehicle follows its trajectory through the intersection.

V. SIMULATION RESULTS

To demonstrate the functionality of the total Cooperative Intersection Control system let us consider a scenario involving six vehicles, two per lane, with the following characteristics:

- V_1 , with intersection counter m = 1, lane $k_1 = 1$, intention $\eta_1 = l$, and virtual platoon index $i_1 = 1$.
- V_2 , with intersection counter m = 2, lane $k_2 = 2$, intention $\eta_2 = s$, and virtual platoon index $i_2 = 2$.
- V_3 , with intersection counter m = 3, lane $k_3 = 3$, intention $\eta_3 = s$, and virtual platoon index $i_3 = 2$.
- V_4 , with intersection counter m = 4, lane $k_4 = 1$, intention $\eta_4 = r$, and virtual platoon index $i_4 = 3$.
- V_5 , with intersection counter m = 5, lane $k_5 = 3$, intention $\eta_5 = l$, and virtual platoon index $i_5 = 3$.
- V_6 , with intersection counter m = 6, lane $k_6 = 2$, intention $\eta_6 = r$, and virtual platoon index $i_6 = 4$.

Note that the vehicle counter and virtual platoon index (related to priority) are determined by the TVA subsystem. Moreover, note that the virtual platoon leader crosses the intersection with constant longitudinal velocity.

A representation of the virtual platoon (involving three relevant sub-platoons) is depicted in Figure 5. Figure 6 shows the evolution in time of all the scaled path coordinates s'_m (note that for the vehicles with straight trajectories $s'_m = s_m$).

The point $s'_m = r_{cz}$ represents the vehicles crossing the middle of the intersection (since after scaling all vehicles travel $2r_{cz}$ inside the CZ). It can be seen that the vehicles with the same platoon index i_m cross the intersection at the same time, this is due to the fact that they do not have crossing trajectories. Hence, no unnecessary waiting times for vehicles with non-crossing trajectories are induced. It is evident that none of the



Fig. 5. Virtual platoon representation.



Fig. 6. Scaled path coordinates as a function of time.

vehicles comes to stand still while crossing the intersection, thereby supporting a smooth and efficient traffic flow.

We can conclude that the vehicles cross the intersection in a smooth and safe manner by forming a virtual platoon.

VI. CONCLUSION AND FUTURE WORK

This paper presents the Cooperative Intersection Control methodology that generates virtual platoons of vehicles in an intersection, so that the vehicles can cross the intersection in a safe manner. This methodology relies on the definition of a virtual inter-vehicle distance between vehicles driving on different lanes. Using the virtual distance concept in the so-called Virtual Cooperative Adaptive Cruise Control, we achieve a smooth and efficient flow of vehicles through the intersection, as presented in a simulation study.

We envision to upgrade the methodology by using a carlike kinematic model (instead of a unicycle model) and an algorithm for the target vehicle assignment that aims at optimizing the traffic throughput. The change to a car-like model is intended to further facilitate the experimental implementation on Cooperative Autonomous Vehicles. Moreover, with the optimization algorithm, we intend to take into account the overall flow of vehicles through the intersection such that we can efficiently regulate the desired vehicle flow of each lane of the intersection. With the aforementioned upgrades we can simulate more challenging scenarios where different velocity profiles can be studied and compared with a normal signalized intersection. Such comparison will aid the analysis of the performance of the Cooperative Intersection Control methodology.

REFERENCES

- E.-H. Choi, "Crash factors in intersection-related crashes: An on-scene perspective," National Highway Traffic Safety Administration, Tech. Rep. DOT HS 811 366, 2010.
- [2] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," *Journal of artificial intelligent research*, vol. 31, pp. 591–656, 2008.
- [3] A. Colombo and D. D. Vecchio, "Efficient algorithms for collision avoidance at intersections," in *Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control*, 2012, pp. 145–154.
- [4] J. Gregoire, S. Bonnabel, and A. D. L. Fortelle, "Priority-based coordination of robots," *CoRR*, vol. abs/1306.0785, 2013.
- [5] S. Azimi, G. Bhatia, R. Rajkumar, and P. Mudalige, "Reliable intersection protocols using vehicular networks," in *Cyber-Physical Systems* (ICCPS), ACM/IEEE International Conference on, 2013, pp. 1–10.
- [6] M. R. Hafner, D. Cunningham, L. Caminiti, and D. D. Vecchio, "Cooperative collision avoidance at intersections: Algorithms and experiments," in *Intelligent Transportation Systems, IEEE Transactions on*, vol. 14, no. 3, 2013, pp. 1162 – 1175.
- [7] R. Naumann, R. Rasche, J. Tacken, and C. Tahedi, "Validation and simulation of a decentralized intersection collision avoidance algorithm," in *Intelligent Transportation System, IEEE Conference on*, 1997, pp. 818 – 823.
- [8] A. Uno, T. Sakaguchi, and S. Tsugawa, "A merging control algorithm based on inter-vehicle communication," in *Intelligent Transportation Systems, IEEE/IEEJ/JSAI International Conference on*, 1999, pp. 783 – 787.
- [9] L. Li and F.-Y. Wang, "Cooperative driving at blind crossings using intervehicle communication," *Vehicular Technology, IEEE Transactions* on, vol. 55, no. 6, pp. 1712 – 1724, 2006.
- [10] Q. Jin, G. Wu, K. Boriboonsomsin, and M. Barth, "Management for connected vehicles using an optimal scheduling approach," in *Connected Vehicles and Expo (ICCVE), International Conference on*, 2012, pp. 185 – 190.
- [11] I. H. Zohdy and H. Rakha, "Game theory algorithm for intersectionbased cooperative adaptive cruise control (cacc) systems," in *Intelligent Transportation Systems (ITSC), 15th International IEEE Conference on*, 2012, pp. 1097 – 1102.
- [12] J. Wu, F. Perronnet, and A. Abbas-Turki, "Cooperative vehicle-actuator system: a sequencebased framework of cooperative intersections management," *Intelligent Transport Systems, IET*, vol. 8, no. 4, pp. 352 – 360, 2014.
- [13] J. Lee and B. Park, "Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 1, pp. 81–90, 2012.
- [14] I. H. Zohdy, R. K. Kamalanathsharma, and H. Rakha, "Intersection management for autonomous vehicles using icacc," in *Intelligent Transportation Systems (ITSC)*, 15th International IEEE Conference on, 2012, pp. 1109 – 1114.
- [15] A. I. Morales-Medina, "Control reconfiguration with mixing applied to a longitudinally-automated vehicle," Master's thesis, Eindhoven University of Technology, 2013, DC 2013.059.
- [16] J. Ploeg, B. Scheepers, E. van Nunen, N. van de Wouw, and H. Nijmeijer, "Design and experimental evaluation of cooperative adaptive cruise control," in *Intelligent Transportation Systems (ITSC), 14th International IEEE Conference on*, 2011.
- [17] B. Siciliano and O. Khatib, Eds., *Handbook of Robotics*. Springer-Verlag Berlin Heidelberg, 2008, ch. Motion control of wheeled mobile robots.