

Traffic Automation in Urban Road Networks Using Consensus-based Auction Algorithms For Road Intersections

Fabio Molinari, Alexander Martin Dethof, Jörg Raisch

Abstract—This paper describes a decentralized control strategy for the automation of road intersections and studies its impact on traffic in a realistic urban road network. The controller incorporates a consensus-based auction algorithm (CBAA-M), which allows vehicles to agree on a crossing order at each road intersection, and an on-board model predictive controller that avoids collisions with other traffic participants, while trying to satisfy performance metrics over time. Randomized simulations show that this decentralized control approach guarantees efficiency, safety, and a higher throughput than traditional solutions.

I. INTRODUCTION

In the last decades, the study of fully automated vehicular traffic has been in the spotlight, see [1], [2]. Cutting the human driver out of the loop pledges more safety, higher traffic efficiency, and a decrease of air pollution. The automation of extra-urban traffic has encouraged studies, e.g., on path following [3] and autonomous overtaking with collision avoidance [4]. In an urban environment, where overtaking is usually not allowed, automating road intersections represents the main challenge. Traditionally, in the adjacency of an intersection, traffic lights are in charge of deciding - in a centralized fashion - the crossing order of vehicles. In the near future, when autonomous vehicles will be able to exchange information (according to the so-called V2V - vehicle to vehicle - and V2I - vehicle to infrastructure - communication), traffic lights can be replaced. One possible strategy is to design a centralized optimal controller which gives each vehicle a crossing order. Appropriate factors, such as actual and desired speeds or inter-vehicular distances, will be weighted while trying to maximize the intersections' throughput. However, employing a centralized controller creates real-time implementability issues. In fact, as problem complexity increases exponentially with the addition of more intersections and more vehicles, realistic scenarios defy real-time solutions. Moreover, in case the centralized controller fails, the whole traffic network will be affected.

F. Molinari is with the Control Systems Group - Technische Universität Berlin, Germany.

A. M. Dethof is with the Control Systems Group - Technische Universität Berlin, Germany.

J. Raisch is with the Control Systems Group - Technische Universität Berlin, Germany & Max-Planck-Institut für Dynamik Komplexer Technischer Systeme, Germany.

molinari@control.tu-berlin.de,
alexander.m.dethof@campus.tu-berlin.de,
raisch@control.tu-berlin.de

This work was funded by the German Research Foundation (DFG) within their priority programme SPP 1914 "Cyber-Physical Networking (CPN)", RA516/12-1.

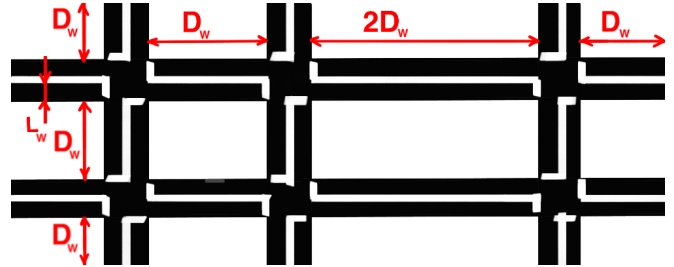


Fig. 1: Urban road network. L_W is the width of the lane, while the length of each road sector is an integer multiple of D_W .

In [5], a possible solution employing a decentralized control structure for the automation of a road intersection is presented. It builds an algorithm for task assignment from robotics (see [6]). In the following, this solution will be referred to as CBAA-M (Consensus Based Auction Algorithm Modified). Vehicles agree on a crossing order for each possible collision point in the intersection by participating in an auction without a central auctioneer. This is possible by employing a consensus protocol see e.g. [7]. Inspired by [8], each vehicle will use an on-board Model Predictive Controller (MPC) designed to avoid collisions with higher priority (and frontal) vehicles while minimizing some cost. This fully decentralized control strategy was shown to guarantee a real-time collision-free solution to the problem while providing a high throughput.

The following work aims to extend the results presented in [5] to the case of urban road networks with many intersections. In Section II, a realistic urban scenario, composed of a collection of adjacent road intersections is described. In Section III, CBAA-M is reviewed and proven to achieve finite-time convergence for groups of cars with a connected (but not necessarily fully connected) network topology. The optimal control structure is introduced in Section IV; differently from [5], vehicles are also allowed to turn left at intersections. Section V presents simulation results and quantifies the impact of this solution. Finally, an analysis of the impact of disallowing left turning is presented.

A. Notation

Throughout the paper, \mathbb{N}_0 denotes the set of nonnegative integers, \mathbb{R} the set of real numbers, and \mathbb{N} the set of positive integers. The set of nonnegative and positive real numbers are, respectively, $\mathbb{R}_{\geq 0}$ and $\mathbb{R}_{> 0}$. Given a set S , its cardinality is $|S|$. An undirected graph is a pair (S, \mathcal{A}) , where S is a set of nodes and $\mathcal{A} \subseteq [S]^2$ a set of arcs, with $[S]^2$ denoting

the set of all two-element subsets of S . Given a node $i \in S$, its neighbors set is $\mathcal{S}_i = \{j \mid \{i, j\} \in \mathcal{A}\}$. A path between nodes $i_1 \in S$ and $i_n \in S$ $i_1 \neq i_n$, is a set of arcs of the form $\Pi = \{\{i_1, i_2\}, \{i_2, i_3\}, \dots, \{i_{n-2}, i_{n-1}\}, \{i_{n-1}, i_n\}\}$. The graph (S, \mathcal{A}) is connected if there exists a path between any pair of nodes i, j , $i \neq j$. It is fully connected, if there is an arc between any two nodes, i.e., $\mathcal{A} = [S]^2$. The i -th entry of vector \mathbf{v} is $(\mathbf{v})_i$. Two sorting functions are used. The function $\text{sort} : \mathbb{R}_{\geq 0}^n \rightarrow \mathbb{R}_{\geq 0}^n$ organizes vector elements in decreasing order of magnitude; the function $\text{argsort} : \mathbb{R}_{\geq 0}^n \rightarrow \mathbb{N}^n$ displays vector indices in decreasing order of their respective entries' magnitude. The function $\text{argmax} : \mathbb{R}_{\geq 0}^n \rightarrow \mathbb{N}$ yields the index of the maximal entry in the vector. If more than one entry has maximum value, one index is selected randomly among the possible candidates.

II. PROBLEM DESCRIPTION

An *urban road network* is a system of interconnected roads which are designed to carry vehicular traffic, as in Figure 1. Each road is composed of two lanes, one per direction. For simplicity and without loss of generality, only a *Manhattan-like* grid, in which all roads are perpendicularly intersecting, will be analyzed. Overtaking and *u-turns* are forbidden. The traffic is composed of fully autonomous vehicles communicating with each other (V2V). The availability of a common shared clock is assumed.

A. Vehicle model

A set $\mathbf{N} = \{1, \dots, n\}$ of $n > 1$ vehicles driving on the urban road network is considered. Each vehicle, say $i \in \mathbf{N}$, is given a desired path in global coordinates, i.e. $\mathcal{P}_i = \{\mathbb{X}_i^g, \mathbb{Y}_i^g\} \subset \mathbb{R}^2$, which comes from a higher-level GPS navigation system. The latter provides vehicle i also with a desired cruising speed, i.e. $v_i^r \in \mathbb{R}_{>0}$. Each vehicle $i \in \mathbf{N}$ is modeled, as in [9], by a point-mass discrete-time linear system. Let $\mathbf{x}_i(k) = [p_i(k), v_i(k)]'$ be its state vector, where $p_i(k) \in \mathbb{R}_{>0}$ and $v_i(k) \in \mathbb{R}_{\geq 0}$ are, respectively, position and velocity along \mathcal{P}_i at discrete time instant $k \in \mathbb{N}_0$;

$$\mathbf{x}_i(k+1) = A\mathbf{x}_i(k) + Bu_i(k), \quad (1)$$

where

$$A = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ T_s \end{bmatrix}, \quad (2)$$

$T_s \in \mathbb{R}_{>0}$ is the sampling time, $\mathbf{x}_i(0) = \mathbf{x}_{i0}$, and $u_i : \mathbb{N} \rightarrow \mathbb{R}$ is the longitudinal acceleration of vehicle i along \mathcal{P}_i . In the following, we refer to p_i as position in *local coordinates*. Each vehicle i will have a local to global map $\mathcal{M}_i : \mathbb{R}_{\geq 0} \rightarrow \mathcal{P}_i$, which will associate p_i to the respective position in the global frame, i.e.

$$\mathcal{M}_i(p_i(k)) = \mathbf{x}_i^g(k) = (x_i^g(k), y_i^g(k)) \in \mathcal{P}_i. \quad (3)$$

The global to local map is also defined and denoted by $\mathcal{M}_i^{-1} : \mathcal{P}_i \rightarrow \mathbb{R}_{\geq 0}$. Given two vehicles i, j and a discrete-time index $k \in \mathbb{N}_0$, their distance is defined as

$$d_{i,j}^k = d(\mathbf{x}_i^g(k), \mathbf{x}_j^g(k)) = \sqrt{(x_i^g(k) - x_j^g(k))^2 + (y_i^g(k) - y_j^g(k))^2}. \quad (4)$$

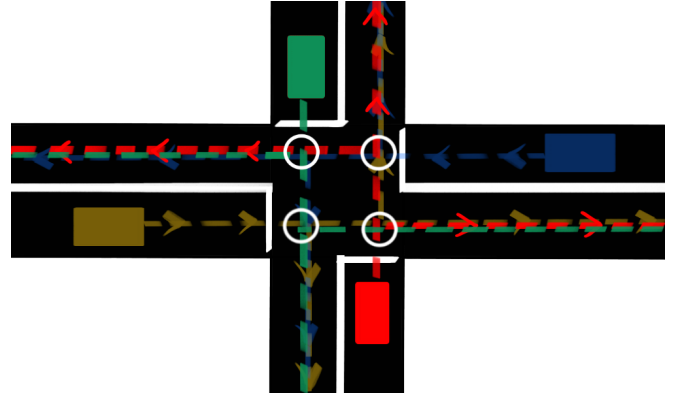


Fig. 2: Possible collision points in one intersection.

We say that, at time k , a collision between vehicle i and j occurs if $d_{i,j}^k < \underline{d}$, where $\underline{d} \in \mathbb{R}_{>0}$ is a given minimum allowed distance.

B. Frontal Vehicles

Clearly, different vehicles can simultaneously drive along the same lane. It is important that each vehicle i recognizes its current frontal vehicles, so that any possible bumper-to-bumper collision can be avoided. Let \mathcal{F}_i^k be the set of i 's frontal vehicles at time k , formally

$$\mathcal{F}_i^k = \{j \in \mathbf{N} \mid \mathcal{M}_j(p_j(k)) \in \mathcal{P}_i, \mathcal{M}_i^{-1}(\mathcal{M}_j(p_j(k))) > p_i(k)\}. \quad (5)$$

C. Collision Points

Vehicle i should be aware of all those vehicles driving on different road sections but going to intersect \mathcal{P}_i . All the possible paths, i.e. $\{\mathcal{P}_i \mid i \in \mathbf{N}\}$, will determine the set of all possible *collision points* in the network (cf. Figure 2),

$$\mathcal{H} = \{(h_x, h_y) \in \mathbb{R}^2\}. \quad (6)$$

Let G_i^k be the set of all collision points that vehicle i has still to cross at time k , i.e.

$$G_i^k = \{h \in \mathcal{H} \mid h \in \mathcal{P}_i, \mathcal{M}_i^{-1}(h) > p_i(k)\}. \quad (7)$$

On the other hand, given $h \in \mathcal{H}$, the set H_h^k collects all the vehicles that still have to cross, at time k , the collision point h , i.e.,

$$H_h^k = \{i \in \mathbf{N} \mid h \in \mathcal{H}, h \in \mathcal{P}_i, \mathcal{M}_i^{-1}(h) > p_i(k)\}. \quad (8)$$

Let \mathcal{L}_i^k be the set of all vehicles with a higher priority than vehicle i for crossing the collision point in G_i^k . A method to compute \mathcal{L}_i^k in a complete decentralized fashion will be presented in Section III.

D. Control Structure

This information plays a crucial role within the hierarchically structured controller for vehicle i (see Figure 3). The ingredients of this controller are described below:

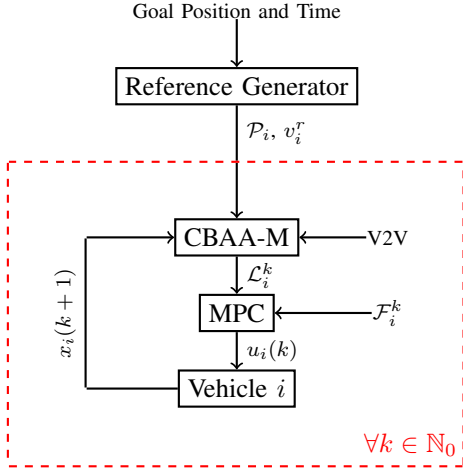


Fig. 3: Hierarchical control structure for vehicle i .

a) *Reference Generator*: Its task is providing the on-board controller with the desired path and cruising speed, given the goal position. It is outside the scope of this paper.

b) *CBAA-M*: this layer runs the so-called *Consensus-based Auction Algorithm Modified* (see Section III). Vehicle i will run this algorithm at every discrete-time step k thus providing the controller with the set \mathcal{L}_i^k of higher priority vehicles, introduced in Section II-C.

c) *MPC*: this on-board Model Predictive Controller minimizes a cost function (reflecting deviations from the desired speed, discomfort, etc) while avoiding collisions with vehicles in set $\mathcal{F}_i^k \cup \mathcal{L}_i^k$. Its output is the acceleration $u_i(k)$.

The proposed structure builds on the idea of avoiding collisions only with higher priority and frontal vehicles, which was originally proposed in [8].

III. CONSENSUS-BASED AUCTION ALGORITHM MODIFIED

CBAA-M is an algorithm that allows a multi-agent system modeled by a graph $(\mathcal{S}, \mathcal{A})$ to achieve an agreement between a set of agents. It has been presented in [5], which, in turn, takes inspiration from [6]. This algorithm is able to run an auction without requiring the presence of any central auctioneer. It can therefore be employed in a completely decentralized control approach, as the one adopted in our strategy. In the following, we will briefly summarize CBAA-M and prove that it converges for connected graphs. This is a sharper result than in [5], where full connectedness was assumed.

Each agent $i \in \mathcal{S} = \{1, \dots, S\}$ places a bid $c_i \in \mathbb{R}_{>0}$ that determines its position in the sequence. We assume that each agent has a distinct bid to place, i.e. $\forall i \neq j, c_i \neq c_j$. The higher the bid, the earlier that agent could appear in the resulting sequence. Each agent has two vectors of dimension S , i.e. \mathbf{v}_i and \mathbf{w}_i , which are updated at each iteration $\kappa \in \mathbb{N}$ and are initialized as zero vectors. The first vector (winners list) has to be filled with a sequence of agents' indexes, whilst the second vector contains their respective bids. This

algorithm is composed of two subsequent phases within every iteration step k :

- 1) *Local Auction [Algorithm 1]*: at each iteration κ , each agent $i \in \mathcal{S}$ places, if its index is not already stored in \mathbf{v}_i^κ , its own bid c_i in the earliest possible position of vector \mathbf{w}_i^κ . In the same position, it stores its index in vector \mathbf{v}_i^κ .

Algorithm 1 CBAA-M Phase 1 for agent i at iteration κ

```

1:  $\mathbf{v}_i^0 = \mathbf{0}_S, \mathbf{w}_i^0 = \mathbf{0}_S$ 
2: procedure BID( $c_i, \mathbf{v}_i^{\kappa-1}, \mathbf{w}_i^{\kappa-1}$ )
3:    $\mathbf{v}_i^\kappa \leftarrow \mathbf{v}_i^{\kappa-1}$ 
4:    $\mathbf{w}_i^\kappa \leftarrow \mathbf{w}_i^{\kappa-1}$ 
5:    $j \leftarrow 1$ 
6:   loop:
7:     if  $i \neq (\mathbf{v}_i^\kappa)_l, \forall l = 1 \dots S$  then
8:       if  $c_i > (\mathbf{w}_i^{\kappa-1})_j$  then
9:          $(\mathbf{v}_i^\kappa)_j \leftarrow i$ 
10:         $(\mathbf{w}_i^\kappa)_j \leftarrow c_i$ 
11:        $j \leftarrow j + 1$ 
12:     goto loop.
13: close;
```

- 2) *Consensus over the lists [Algorithm 2]*: after the first phase, each agent has its own version of \mathbf{v}_i^κ and \mathbf{w}_i^κ . The network need to reach an agreement on them. For that purpose, each agent $i \in \mathcal{S}$ sends its vectors to its respective neighbors (i.e. the nodes in the set \mathcal{S}_i) and receives theirs. Then, via a max-consensus protocol, it selects the best bid for each row of \mathbf{w}_i^κ and puts in the same position of \mathbf{v}_i^κ the respective agent's index.

Algorithm 2 CBAA-M Phase 2 for agent i at iteration κ

```

1: SEND  $(\mathbf{v}_i^\kappa, \mathbf{w}_i^\kappa)$  to  $j \in \mathcal{S}_i = \{j \in \mathcal{S} \mid (i, j) \in \mathcal{E}\}$ 
2: RECEIVE  $(\mathbf{v}_h^\kappa, \mathbf{w}_h^\kappa)$  from  $h \in \mathcal{S}_i = \{j \in \mathcal{S} \mid (j, i) \in \mathcal{E}\}$ 
3: procedure UPDATE( $\mathbf{v}_{h \in \mathcal{S}_i}^\kappa, \mathbf{w}_{h \in \mathcal{S}_i}^\kappa$ )
4:    $(\mathbf{a}_i^\kappa)_j \leftarrow \arg \max_{h \in \mathcal{S}_i} ((\mathbf{w}_h^\kappa)_j), \forall j = 1 \dots \max(\kappa, S)$ 
5:    $(\mathbf{v}_i^\kappa)_j \leftarrow (\mathbf{v}_{(\mathbf{a}_i^\kappa)_j}^\kappa)_j \quad \forall j = 1 \dots \max(\kappa, S)$ 
6:    $(\mathbf{w}_i^\kappa)_j \leftarrow \max_{h \in \mathcal{S}_i} ((\mathbf{w}_h^\kappa)_j) \quad \forall j = 1 \dots \max(\kappa, S)$ 
```

After terminating Phase 2, $(\mathbf{w}_h^\kappa)_j$ is the maximal bid for position j that agent i is aware of, and $(\mathbf{v}_h^\kappa)_j$ is the index of the agent having placed that bid.

In [5], a fully connected network topology was assumed and an agreement was shown to be reached in exactly S iterations. In the following, we will show that the algorithm converges under the weaker condition of $(\mathcal{S}, \mathcal{A})$ being connected.

Proposition 1. A multi-agent system represented by a connected graph $(\mathcal{S}, \mathcal{A})$ executes CBAA-M. An agreement is reached in $\bar{\kappa} \in \mathbb{N}$ iterations. Formally, $\exists \bar{\kappa} \in \mathbb{N}$:

$$\forall i, j \in \mathbb{N}, \quad \mathbf{v}_i^{\bar{\kappa}} = \mathbf{v}_j^{\bar{\kappa}} = \mathbf{v}^* = \text{argsort}(\mathbf{c}), \quad (9)$$

$$\mathbf{w}_i^{\bar{\kappa}} = \mathbf{w}_j^{\bar{\kappa}} = \mathbf{w}^* = \text{sort}(\mathbf{c}), \quad (10)$$

where, $\forall i \in \mathcal{S}$, $(c)_i = c_i$.

Proof. Given a pair $i \in \mathcal{S}$, $\kappa \in \mathbb{N}$, such that $\mathbf{v}_i^\kappa = \mathbf{v}^*$ and $\mathbf{w}_i^\kappa = \mathbf{w}^*$, the following holds:

$$\forall j \in \mathcal{S}_i \cup \{i\}, \mathbf{v}_j^{\kappa+1} = \mathbf{v}^*, \mathbf{w}_j^{\kappa+1} = \mathbf{w}^*. \quad (11)$$

In fact, if (11) does not hold, Algorithm 2 would imply that, for an arbitrary $j \in \mathcal{S}_i$, $\exists g = 1, \dots, S$, $(\mathbf{w}_j^g)_g > (\mathbf{w}^*)_g$. The latter, by Algorithm 1, means that $\exists h = 1, \dots, S : c_g < c_h < c_{g+1}$, which is in contradiction with the definition of \mathbf{w}^* , i.e. $\mathbf{w}^* = \text{sort}(\mathbf{c})$. A similar analysis can be conducted for \mathbf{v}_j^κ .

The agreement phase of CBAA-M follows a max-consensus protocol for each row of \mathbf{v}_i and \mathbf{w}_i . By [10], max-consensus is achieved in a connected network in at most l steps, where

$$l = \max_{i,j \in \mathcal{S}} \{|i, j|_{l, \min}\}, \quad (12)$$

where $|i, j|_{l, \min}$ denotes the length of the shortest existing path between node i and node j . Therefore, by (11) and (12), if agent i_0 at iteration κ_0 is the first agent to have $\mathbf{v}_{i_0}^{\kappa_0} = \mathbf{v}^*$ and $\mathbf{w}_{i_0}^{\kappa_0} = \mathbf{w}^*$, then $\bar{\kappa} \leq \kappa_0 + l$.

Let's now estimate κ_0 in the worst case scenario. By Algorithm 1, intuitively, i_0 is the first agent to fill the last row of \mathbf{w}_i . This is only possible, by Algorithm 1, if i_0 is not contained in \mathbf{v}_i and if the first $S - 1$ rows of \mathbf{w}_i have already been filled. By this, agent i_0 must be the agent having the minimum bid in the network, i.e. $c_{i_0} = \min_{i \in \mathcal{S}} c_i$. In the worst case scenario, by [10] and (12), for retrieving the information of each row j of \mathbf{v}_i and \mathbf{w}_i , agent i needs maximum l steps. Therefore, always in the worst case scenario, $\kappa_0 \leq (S - 1)l$. Finally, we can state that the multi-agent system achieves an agreement as (9)-(10) in $\bar{\kappa}$ steps, where

$$\bar{\kappa} \leq \kappa_0 + l \leq (S - 1)l + l = Sl.$$

This concludes the proof. \square \square

Due to the need of real-time implementation, a fast convergence to the agreement vector is often required. In [11], it was shown how the convergence rate of a max-consensus protocol can be drastically increased by harnessing the interference.

As claimed in Section II-D, each vehicle $i \in \mathcal{N}$, at every discrete time step $k \in \mathbb{N}_0$, runs one CBAA-M for each collision point $h \in G_i^k$, thus retrieving $|G_i^k|$ priority vectors. These can be grouped in a set $\{\mathbf{v}_h^* \mid h \in G_i^k\}$, where $\mathbf{v}_h^* \in \mathbb{N}^{|H_h^k|}$ is the agreed crossing priority list for collision point h . From this collection of vectors, vehicle i can retrieve the set \mathcal{L}_i^k (cf. Section II-C) that collects all the vehicles having higher priority than i at some collision points. Formally,

$$\mathcal{L}_i^k = \{(\mathbf{v}_h^*)_j \mid h \in G_i^k, j < m, (\mathbf{v}_h^*)_m = i\}. \quad (13)$$

As in [5], the bid $c_{i,h}^k$ placed by vehicle i at time k in the auction for crossing $h \in G_i^k$ is determined from its current

velocity and its distance from h :

$$c_{i,h}^k = \frac{p_v v_i(k) + p_d}{d_{i,h}^k + \epsilon}, \quad (14)$$

where $p_v \in \mathbb{R}_{>0}$, $p_d \in \mathbb{R}_{>0}$, and $\epsilon \in \mathbb{R}_{>0}$ are chosen parameters (equal for all vehicles). The quantity $d_{i,h}^k$ is the distance at time k from vehicle i to the collision point $h \in \mathcal{H}$, i.e., $d_{i,h}^k = \sqrt{(x_i^g(k) - h_x)^2 + (y_i^g(k) - h_y)^2}$. The reader can refer to [5] for a comprehensive analysis of $c_{i,h}^k$ and potential coherency problems in the auction procedures.

IV. ONBOARD MODEL PREDICTIVE CONTROLLER

As shown in Figure 3, each vehicle $i \in \mathcal{N}$ has an onboard MPC controller that computes an optimal longitudinal acceleration at every discrete step $k \in \mathbb{N}_0$. It is provided with the desired path \mathcal{P}_i , the desired cruising speed v_i^r , the current vehicle state $\mathbf{x}_i(k)$, the sets \mathcal{L}_i^k and \mathcal{F}_i^k , and the states $\mathbf{x}_j(k)$ of all vehicles $j \in \mathcal{L}_i^k \cup \mathcal{F}_i^k$.

A. Prediction Model

MPC employs a prediction model of the traffic situation extrapolated from the current states and evolving along a horizon $T_h \in \mathbb{N}$. In the following, we describe the optimal control problem to be solved at every $k \in \mathbb{N}_0$ by vehicle $i \in \mathcal{N}$.

The predicted state and input of vehicle i itself are $\tilde{\mathbf{x}}_i(t) = [\tilde{p}_i(t), \tilde{v}_i(t)]$ and $\tilde{u}_i(t)$, $t \in [0, T_h]$. The prediction model evolves according to

$$\tilde{\mathbf{x}}_i(t+1) = A\tilde{\mathbf{x}}_i(t) + B\tilde{u}_i(t), \quad (15)$$

where $t \in [0, T_h - 1]$. The prediction model of other vehicles is based on a constant acceleration assumption: in fact, $\forall \zeta \in \mathcal{F}_i^k \cup \mathcal{L}_i^k$,

$$\tilde{\mathbf{x}}_\zeta(t+1) = A\tilde{\mathbf{x}}_\zeta(t) + B\tilde{u}_\zeta, \quad (16)$$

where $t \in [0, T_h - 1]$ and $\tilde{u}_\zeta = u_\zeta(k)$. Clearly, the prediction variables are initialized according to the current measurement at instant k , i.e.,

$$\tilde{\mathbf{x}}_i(0) = \mathbf{x}_i(k), \quad (17)$$

$$\forall \zeta \in \mathcal{L}_i^k \cup \mathcal{F}_i^k, \tilde{\mathbf{x}}_\zeta(0) = \mathbf{x}_\zeta(k). \quad (18)$$

Remark 1. Let \mathcal{F}_i^t , $\forall t \in [0, T_h]$, be the set of vehicles $j \in \mathcal{L}_i^k \cup \mathcal{F}_i^k$ that are in front of vehicle i at prediction time t . Formally, as in (5),

$$\mathcal{F}_i^t = \{j \in \mathcal{L}_i^k \cup \mathcal{F}_i^k \mid \mathcal{M}_j(\tilde{p}_j(t)) \in \mathcal{P}_i, \mathcal{M}_i^{-1}(\mathcal{M}_j(\tilde{p}_j(t))) \geq \tilde{p}_i(t)\}. \quad (19)$$

Since any vehicle in $\mathcal{L}_i^k \cup \mathcal{F}_i^k$ always has priority in front of i , requiring $\mathcal{M}_i^{-1}(\mathcal{M}_j(\tilde{p}_j(t))) \geq \tilde{p}_i(t)$ in (19) is redundant. Accordingly, (19) can be rewritten as

$$\mathcal{F}_i^t = \{j \in \mathcal{L}_i^k \cup \mathcal{F}_i^k \mid \mathcal{M}_j(\tilde{p}_j(t)) \in \mathcal{P}_i\}, \quad (20)$$

thus making it clear that the usage of \mathcal{F}_i^t does not affect the convexity of the problem, since it does not depend on the choice of $\tilde{u}_i(t)$.

B. Obstacle Avoidance

We employ, similarly to [8], the idea of avoiding collisions with higher priority and frontal vehicles. The safety distance to be kept is defined via the *continuous-time headway rule*, as in [12]. Then, a general collision avoidance constraint between vehicle i and an arbitrary vehicle ζ at prediction time t is of the form

$$d(\tilde{\mathbf{x}}_i^g(t), \tilde{\mathbf{x}}_\zeta^g(t)) \geq \lambda \tilde{v}_i(t) + \underline{d} + \delta(t), \quad (21)$$

where λ is the so called *time-headway* (measured in seconds) and \underline{d} is the fixed *bumper-to-bumper* distance defined in Section II-A. The remaining term, $\delta(t)$, is a slack variable that will be weighted in the cost function and that is traditionally used in the formulation of soft constraints. This allows the optimal controller to increase, if possible, the safety distance. In Section IV-C, $\delta(t)$ will be constrained to a given set.

Clearly, having $d(\tilde{\mathbf{x}}_i^g(t), \tilde{\mathbf{x}}_\zeta^g(t))$ formulated as in (4) makes the problem non-convex. In order to prevent this, in [5], collision avoidance constraints were reformulated. By [5, Proposition 3], constraints to avoid collisions with frontal vehicles can be rewritten as

$$\forall t \in [0, T_h], \forall \zeta \in \mathcal{F}_i^t, \quad \mathcal{M}_i^{-1}(\mathcal{M}_\zeta(\tilde{p}_\zeta(t))) - \tilde{p}_i(t) \geq \lambda \tilde{v}_i(t) + \underline{d} + \delta(t). \quad (22)$$

Clearly, (22) is convex.

In [5], left turns are disallowed, since allowing left turns is claimed to lower traffic efficiency. In this paper, we do allow left turns at intersections and analyze the effect of this in Section V-B. First, we introduce a general convex collision avoidance constraint for vehicle i and all higher priority vehicles crossing \mathcal{P}_i i.e. $\zeta \in \mathcal{L}_i^k \setminus \mathcal{F}_i^k$. The following proposition extends the outcome of [5, Proposition 4].

Proposition 2. Given a vehicle $\zeta \in \mathcal{L}_i^k \setminus \mathcal{F}_i^k$, the constraint, $\forall h \in G_i^k \cap G_\zeta^k$:

$$\mathcal{M}_i^{-1}(h) - \tilde{p}_i(t) \geq \lambda \tilde{v}_i(t) + \underline{d} + \delta(t), \quad \forall t \in [0, T_h] : \zeta \notin \mathcal{F}_i^t \wedge \mathcal{M}_\zeta^{-1}(h) - \tilde{p}_\zeta(t) \geq -\underline{d}, \quad (23)$$

guarantees that any collision between i and ζ throughout the prediction horizon is avoided.

Proof. A collision between i and ζ occurs at prediction time t if, as in Section II-A,

$$d(\mathbf{x}_i^g(t), \mathbf{x}_\zeta^g(t)) \leq \underline{d}. \quad (24)$$

We can distinguish three different cases. (i) In the case $\zeta \in \mathcal{F}_i^t$, any collision is avoided if (22) holds. (ii) if $\zeta \notin \mathcal{F}_i^t$, but $\mathcal{M}_\zeta^{-1}(h) - \tilde{p}_\zeta(t) < -\underline{d}$, vehicle ζ at prediction time t does not lie on \mathcal{P}_i and has driven at a distance larger than \underline{d} far from \mathcal{P}_i , (24) does not hold. (iii) as long as $\zeta \notin \mathcal{F}_i^t \wedge \mathcal{M}_\zeta^{-1}(h) - \tilde{p}_\zeta(t) \geq -\underline{d}$, the distance between i and \mathcal{P}_ζ is larger than $\lambda \tilde{v}_i(t) + \underline{d} + \delta(t)$, which is clearly larger than \underline{d} . This concludes the proof. \square \square

TABLE I: Problem data.

$L_w = 3.5m, D_w = 30m, T_s = 0.25s, p_c = 1, p_d = 0.1, \epsilon = 0.1$
$\lambda = 1s, \bar{\lambda} = 0.5s, \bar{\delta} = 10m, \underline{d} = 2.1m, \underline{v} = 0, \bar{v} = 130km/h,$
$T_h = 10, \underline{a}_i = -9m/s^2, \bar{a}_i = 5m/s^2, q = 0.1, r = 0.01, \omega = -0.1$
$v_{min} = 52km/h, v_{max} = 56km/h, \wp = 0.5$

The resulting convex collision avoidance constraint then becomes:

$$\begin{aligned} \forall h \in G_i^k \cap G_\zeta^k, \quad \mathcal{M}_i^{-1}(h) - \tilde{p}_i(t) &\geq \lambda \tilde{v}_i(t) + \underline{d} + \delta(t), \\ \forall t \in [0, T_h] : \zeta \notin \mathcal{F}_i^t \wedge \mathcal{M}_\zeta^{-1}(h) - \tilde{p}_\zeta(t) &\geq -\underline{d}. \end{aligned} \quad (25)$$

C. Other Constraints and Cost Function

Traditional constraints on the allowed speed and acceleration can be formulated as follows:

$$\forall t \in [0, T_h], \tilde{u}_i(t) \in [\underline{a}_i, \bar{a}_i], \quad (26)$$

$$\tilde{v}_i(t) \in [\underline{v}, \bar{v}], \quad (27)$$

$$\tilde{v}_\zeta(t) \in [\underline{v}, \bar{v}], \quad (28)$$

where $\underline{v} > 0$, thus guaranteeing that vehicles do not drive backwards.

The slack variable $\delta(t)$ is also constrained to a given set, i.e.

$$\forall t \in [0, T_h], \delta(t) \in [-\bar{\lambda} \tilde{v}_i(t), \bar{\delta}], \quad (29)$$

where $0 \leq \bar{\lambda} < \lambda$ and $\bar{\delta} > 0$. By this, the time headway in (21) can be diminished down to $\lambda - \bar{\lambda}$, although this will be negatively weighted in the cost function, thus aiming to increase the safety distance.

The controller performance can be evaluated by the following cost function

$$J = \sum_{t=0}^{T_h} q(\tilde{v}_i(t) - v_i^r)^2 + r \tilde{u}_i^2(t) + \omega \delta(t), \quad (30)$$

where $q, r \in \mathbb{R}_{>0}$ and $\omega \in \mathbb{R}_{<0}$ are design parameters. The first term of the sum punishes deviations from the desired speed v_i^r ; the second term discourages high values of acceleration or deceleration, thus pledging a better comfort on-board. The last term punishes small $\delta(t)$, hence contributing to crease safety.

The optimal control problem is then formulated as follows:

$$\begin{aligned} \min_{\tilde{u}_i(0) \dots \tilde{u}_i(T_h)} \quad & \text{cost function (30)} \\ \text{s.t.} \quad & \text{Prediction model (15-18)} \\ & \text{Safety constraints (22, 25)} \\ & \text{Input and state constraints (26- 29).} \end{aligned}$$

A vector of inputs composed of $\tilde{u}_i(0) \dots \tilde{u}_i(T_h)$ will solve the optimal control problem. Coherently with the MPC structure, only the first optimal control input will be executed in (1) at each $k \in \mathbb{N}$, i.e. $u_i(k) = \tilde{u}_i(0)$.

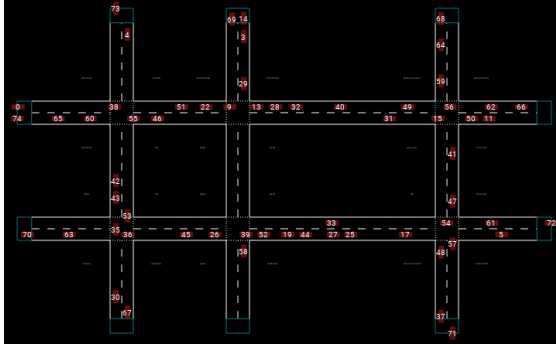
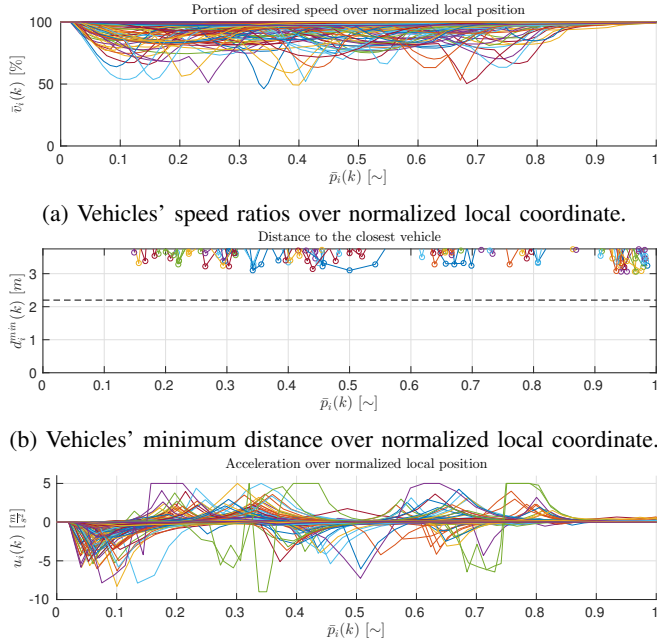


Fig. 4: Using the open-source software YATSim for modeling the urban road network.



(a) Vehicles' speed ratios over normalized local coordinate.
(b) Vehicles' minimum distance over normalized local coordinate.
(c) Vehicles' accelerations over normalized local coordinate. The dashed line indicates the minimum allowed distance,

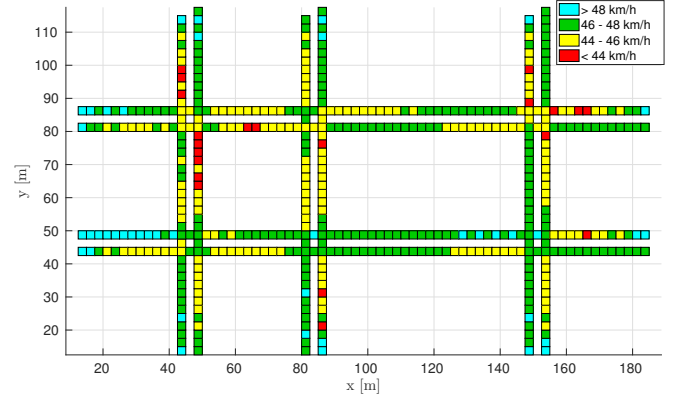
Fig. 5: Analysis of individual vehicles' characteristics.

V. SIMULATION

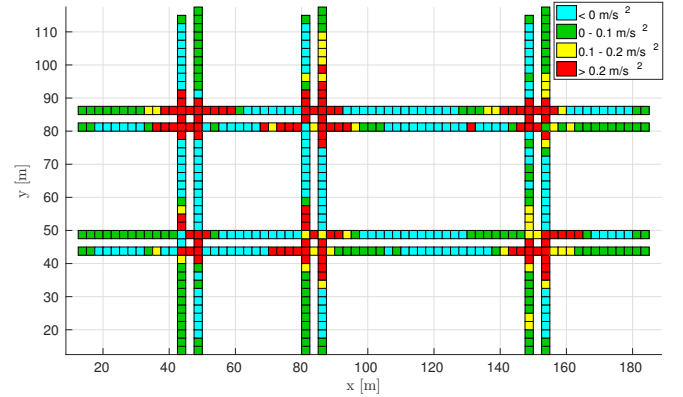
The traffic flowing in the urban road network sketched in Figure 1 can be simulated by means of the open-source programme YATSim [13], as in Figure 4. Vehicles will employ the decentralized control strategy presented in this paper. In the following, a macroscopic analysis of the traffic (in the sense of [14]) will be conducted. Accordingly, the traffic will be studied as a flow. This allows to quantify the impact that this control strategy has on the urban road network as a whole, hopefully resulting in an improvement of flow performance indexes, such as average or minimum speed and acceleration.

A. Macroscopic Analysis

Vehicles will be randomly injected into the network; at each discrete time step from each road, provided that there is



(a) Macroscopic analysis of the average speed in road network sections.



(b) Macroscopic analysis of the average acceleration in road network sections.

Fig. 6: Macroscopic analysis.

enough space to avoid trivial initial collisions, a vehicle, say $i \in \mathbb{N}$, will enter the road network with a given probability $\varphi \in (0, 1)$. Its desired speed v_i^r will be drawn out of a uniform distribution $\mathcal{U}_{[v_{min}, v_{max}]}$. Also its desired path \mathcal{P}_i will be chosen in a random way, as illustrated in [13]. The parameters used for the simulation are contained in Table I. The simulation is run until more than 500 vehicles complete their paths. For what concerning simulation results, the average speed of the traffic is $46.8[\frac{km}{h}]$, which, by [15], is much higher than the current average traffic speed in New York City ($28.32[\frac{km}{h}]$) or Boston ($33.64[\frac{km}{h}]$). We use these two cities as benchmark, since the road network structure composed of perpendicularly intersecting streets is similar to the one considered here. Moreover, φ has been chosen to replicate real congestion conditions. Vehicles are injected in the simulation environment with their desired speeds. Intuitively, they will have to slow down in order to avoid collisions. In fact, the average acceleration of the traffic is $-0.0212[\frac{m}{s^2}]$, thus showing a braking behavior.

Given a vehicle $i \in \mathbb{N}$, let $\bar{p}_i(k)$ be its normalized local

coordinate, i.e. $\forall k \in \mathbb{N}$,

$$\bar{p}_i(k) = \frac{p_i(k)}{\max_k p_i(k)} \in [0, 1],$$

let $\bar{v}_i(k)$ be its speed ratio, i.e. $\forall k \in \mathbb{N}$,

$$\bar{v}_i(k) = \frac{v_i(k)}{v_i^r} 100\%,$$

and let $d_i^{min}(k)$ be the minimum distance towards other vehicles at instant k , i.e. $\forall k \in \mathbb{N}$,

$$d_i^{min}(k) = \min_{j \in \mathbb{N} \setminus \{i\}} d_{ij}(k).$$

Although this section presents a macroscopic analysis, a detailed evaluation of individual vehicles' characteristics is also reviewed. Figure 5a shows the evolution of speed ratios as function of normalized local coordinates for each individual vehicle. The large majority of vehicles keep a speed above the 80% of their desired ones, while no vehicle is slowing down below the 48% of its desired speed. By comparing this result to the traditional solution with traffic lights, the improvement is explicit since no vehicle is actually stopping. This might intuitively result in decreasing the traffic safety (which can be defined as the likelihood of having collisions). However, Figure 5b shows that all vehicles keep a safety distance larger than \underline{d} . Moreover, the amount of vehicles getting close to this minimum allowed value is marginal. Figure 5c outlines individual vehicles' accelerations as function of $\bar{p}_i(k)$. As already motivated above, many vehicles brake as soon as they enter the simulation environment. Vehicles are often saturating their acceleration, in those cases when collision avoidance constraints allow to do so (e.g. as soon as that vehicle wins the decentralized auction). This saturating behavior can be intuitively mitigated by increasing r . Only in one point (e.g., for $p_i(k)$ between 0.3 and 0.4), a vehicle is saturating the braking power. This could be prevented by increasing T_h , length of the prediction horizon, or ω , weight of the hold safety distance.

An analysis of which portions of the urban road network suffer the most of speed drops or robust braking is presented. First, the road network has to be divided into small fragments (e.g. squares of side $2.5[m]$) on which we compute the average characteristics of the traffic. As in Figure 6a, average speeds lower than $44[\frac{km}{h}]$ are met in some roads entering an intersection (e.g. $(x = 50, y = 77.5)$ or $(x = 155, y = 77.5)$), while, along those roads leaving an intersection, higher speeds are found. Figure 6b shows that traffic along roads entering an intersection exhibits a braking behavior. On the other hand, vehicles try to leave the intersection as soon as possible once inside of it, thus resulting in higher average acceleration.

B. Impact of Left-Turns

In [5], left turnings were forbidden. Also, [16] revealed that a world-wide known delivery company has historically prohibited left-turnings to its drivers. This was due to the assumption that allowing left-turns affects traffic efficiency

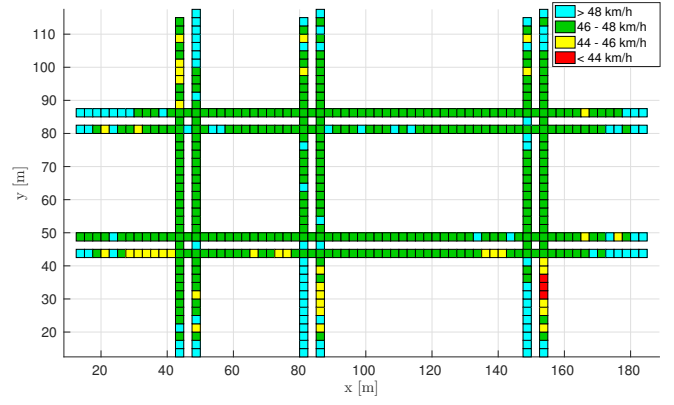


Fig. 7: Macroscopic analysis of the average speed in road network sections. Left-turnings disallowed.

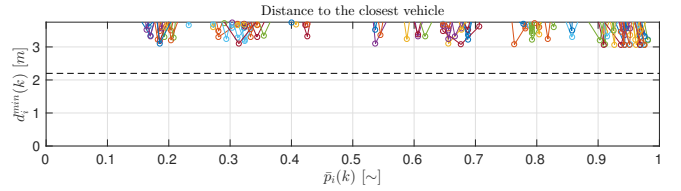


Fig. 8: Vehicles' minimum distance over normalized local coordinate. Left turnings disallowed. The dashed line indicates the minimum allowed distance,

in a negative way and increases hazards. In the following, we repeat our simulation for the case when left-turns are forbidden. We simulate the same urban road network where cars have the same parameters (same initial and goal position) and are generated according to the same probability \wp ; however, left turnings are in this case not allowed. The average speed of the traffic is $47.8[\frac{km}{h}]$ and the average acceleration is $-0.002[\frac{m}{s^2}]$. Additionally, Figure 7 shows that in almost all the portions composing the road network, vehicles keep a higher speed than in Figure 6a. With regards to the outcome of the previous section, it is therefore clear that preventing vehicles from turning left increases the average speed and decreases their absolute acceleration. If the average speed is a measure of traffic efficiency, then the assumption presented by [16] seems reasonable.

On the other hand, under a safety-related point of view, we do not get to the same conclusion. Let us pick, as safety index, the minimum distance that each vehicle keeps towards the others, and let this be plotted, as function of the normalized local coordinates, in Figure 8. Its outcome is not much different than what Figure 5b yields. Therefore, left-turnings do not seem to increase hazards in the autonomous road traffic. One can guess that, although the overall average speed is affected by the presence of a left-turning traffic, safety is anyhow guaranteed by the presence of the on-board MPC controller.

VI. CONCLUSION

In this paper, we suggested the use of a decentralized control scheme for vehicles in a urban traffic network. This scheme is based on a consensus-based protocol (CBAA-M), which gives vehicles crossing orders at the intersections, with an on-board MPC controller, responsible for avoiding collisions and tracking some performance. CBAA-M has been here proven to achieve an agreement in cases when the network topology is connected. This is in contrast to [5], where full connectedness was required. Finally, the impact of disallowing left-turning has been analyzed.

Future work will focus on real-time implementability issues by improving the convergence rate of the consensus protocol and decrease the complexity of the on-board MPC.

REFERENCES

- [1] P. A. Ioannou and C.-C. Chien, "Autonomous intelligent cruise control," *IEEE Transactions on Vehicular technology*, vol. 42, no. 4, pp. 657–672, 1993.
- [2] A. Katriniok, J. P. Maschuw, F. Christen, L. Eckstein, and D. Abel, "Optimal vehicle dynamics control for combined longitudinal and lateral autonomous vehicle guidance," in *European Control Conference (ECC)*. IEEE, 2013, pp. 974–979.
- [3] S. A. Arogeti and N. Berman, "Path following of autonomous vehicles in the presence of sliding effects," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 4, pp. 1481–1492, 2012.
- [4] F. Molinari, N. N. Anh, and L. Del Re, "Efficient mixed integer programming for autonomous overtaking," *American Control Conference (ACC)*, 2017, pp. 2303–2308, 2017.
- [5] F. Molinari and J. Raisch, "Automation of road intersections using consensus-based auction algorithms," *American Control Conference (ACC)*, pp. 5994 – 6001, 2018.
- [6] H.-L. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *IEEE transactions on robotics*, vol. 25, no. 4, pp. 912–926, 2009.
- [7] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Systems*, vol. 27, no. 2, pp. 71–82, 2007.
- [8] A. Katriniok, P. Kleibaum, and M. Joševski, "Distributed model predictive control for intersection automation using a parallelized optimization approach," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5940–5946, 2017.
- [9] N. Murgovski, G. R. de Campos, and J. Sjöberg, "Convex modeling of conflict resolution at traffic intersections," in *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*. IEEE, 2015, pp. 4708–4713.
- [10] B. M. Nejad, S. A. Attia, and J. Raisch, "Max-consensus in a max-plus algebraic setting: The case of fixed communication topologies," in *Information, Communication and Automation Technologies, 2009. ICAT 2009. XXII International Symposium on*. IEEE, 2009, pp. 1–7.
- [11] F. Molinari, S. Stańczak, and J. Raisch, "Exploiting the superposition property of wireless communication for max-consensus problems in multi-agent systems," *7th IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys18)*, 2018.
- [12] C. Chien and P. Ioannou, "Automatic vehicle-following," in *American Control Conference, 1992*. IEEE, 1992, pp. 1748–1752.
- [13] A. M. Dethof and F. Molinari, "Yatsim: an open-source simulator for testing consensus-based control strategies in urban traffic networks," *arXiv preprint arXiv:1810.11380*, 2018.
- [14] A. D. May, *Traffic flow fundamentals*, 1990.
- [15] J. Kleint. (2009) Cityspeed at sourceforge. [Online]. Available: <http://sourceforge.net/projects/cityspeed/>
- [16] A. A. Reich, "Transportation efficiency," *Strategic Planning for Energy and the Environment*, vol. 32, no. 2, pp. 32–43, 2012.