# Multi-level decision system for the crossroad scenario

Bofei Chen[1], Franck Gechter[1], and Abderrafiaa Koukam[1]

IRTES-SeT, 90010 Belfort Cedex, France
bofei.chen@utbm.fr, franck.gechter@utbm.fr, abder.koukam@utbm.fr

**Abstract**
Among the innovations aimed at tackling the transportation issues in urban areas, one of the most promising solutions is the possibility of making virtual trains of vehicles so as to provide a new kind of transportation system. Even if this kind of solution is now widespread in literature, some difficulties still need to be resolved. For instance, one must find solutions to make the crossing of the train possible while maintaining train composition (trains must not be split) and safety conditions. This paper proposes a multi-level decision process aimed at dealing with this issue. This proposal is based on dynamic adaptation of train parameters which lead to trains crossing without stopping any of them. Results, obtained in simulations, make the comparison with a classical crossing strategy.

*Keywords:* Multi-level decision, virtual train, platoon control, crossroad

## 1 Introduction

Traffic jams and air pollution are two main undesirable effects from which urban environments suffer. Traffic jams waste a lot of time and cause air pollution, impairing the health of humans. Safety issues are also concerned since traffic jams and road congestions lead to numerous incidents/accidents that make traffic conditions even worse. In this context, autonomous vehicle capabilities promise numerous improvements aimed at solving those problems. The faster response times increase the road capacity and fluidity the traffic flow, more foresighted driving and reduced fuel consumption and pollution, while collision avoidance systems decrease the number of accidents. [12]. In order to improve further urban road conditions, research work has been done dealing with both infrastructure and vehicles levels. On the infrastructure level, the focus is made on finding new management approaches to deal with the crossroad issues. Indeed, the increasing complexity of transportation networks leads to the inefficiency of the traditional management approaches which have become more and more expensive. Among the new approaches, one can cite the automatic management of status and duration of traffic lights such as those proposed in [3], [14] or in [9], where the weights of all the serial lanes are dynamically computed. In [11], a multi-agent traffic light control system based on a multi-objective sequential decision making framework and a traffic light controller based on the Bayesian interpretation of probability are developed. On the vehicle level, the research works are mainly focused on developing new autonomous capabilities. Among them, one of the most promising solutions is the possibility of making virtual trains of vehicles so as to provide a new kind of transportation system [3]. This solution, based on platoon control algorithm, has been explored in several research projects such as CRISTAL, CATS [1] or SafePlatoon [2]. The key element of this virtual train solution is the ability to control each vehicle to obtain a coherent global behavior on the train level. Basically, two main trends can be found in research literature. On one side, global approaches are based on a global reference frame shared by all the vehicles of the train. Then, each vehicle computes its behavioral parameters according to this shared reference which can be either the trajectory of the first vehicle of the train or a reference trajectory built off-line. These approaches generally require communication between vehicles such as in [10], [7] or [8]. On the other side, local approaches are based on vehicle local perception abilities. Some methods, which are based on classical control algorithms, are developed to implement the longitudinal and lateral control in [4] [15] [17] or physical-inspired and inter-vehicular interaction link [6] [2], [18]. When the works related to virtual train control are abundant, solutions for managing virtual trains at crossroads is very scarce. Generally, the crossing is based on alternations between roads, letting one entire train pass before giving priority to the other road. However, among the few references dealing with this subject, one can cite [3], [13] or [16] which present solutions based on the adaptation of longitudinal distance, each vehicle being considered as an agent. In this paper, we propose a multi-level solution, based on local platoon control algorithm, which aims at adapting virtual train parameters depending on the perception context, shown in Figure 1. The proposed low-level decision process deals with the hardware level of the vehicles. From one command provided by an upper level, the low-level decision process applies this command as fast as possible considering vehicle kinematics, dynamics and hardware constraints. The middle-level decision process is a local platoon control based on a physical inspired virtual linked as the one developed in [5] or in [1]. From a vectorial distance computed from perception data, this middle-level decision processor calculates a command to be applied to the vehicle by incorporating with the parameters of the virtual link (spring, damper, etc.) and the vehicle current state (speed, acceleration, etc.). This calculation takes also into account train parameters such as desired longitudinal distance or global train speed. These elements are the output of the top-level decision processor which deals with global train state. Then, the crossing abilities of the virtual train stems from the interactions at trains level and from their implications on lower levels. Consequently, this global behavior is not deterministic and can be considered as the result of many interactions at train, vehicle and hardware levels. The result obtained is an adaptive intersection of trains without alternation in road priority and where each vehicle adapts its behavior depending on the train, vehicle and hardware levels interactions.

This paper is structured as follows. The multi-level platoon control model is detailed in section 2. Then, experimental results are presented focusing on the evaluation of the improvements provided by our model as compared to a regular intersection management which let pass one train after the other. Finally, section 4 concludes the paper by giving some future extension of this model.

## 2 Multi-level decision system

In this paper, the intelligent crossroad system of virtual trains is treated as a multi-level decision system. This crossroad system includes all the necessary parts of the multi-level decision system:

[1] http://www.parc-innovation-strasbourg.eu/index.php/CATS-Project/welcome-on-cats-webpage.html
[2] http://web.utbm.fr/safeplatoon/

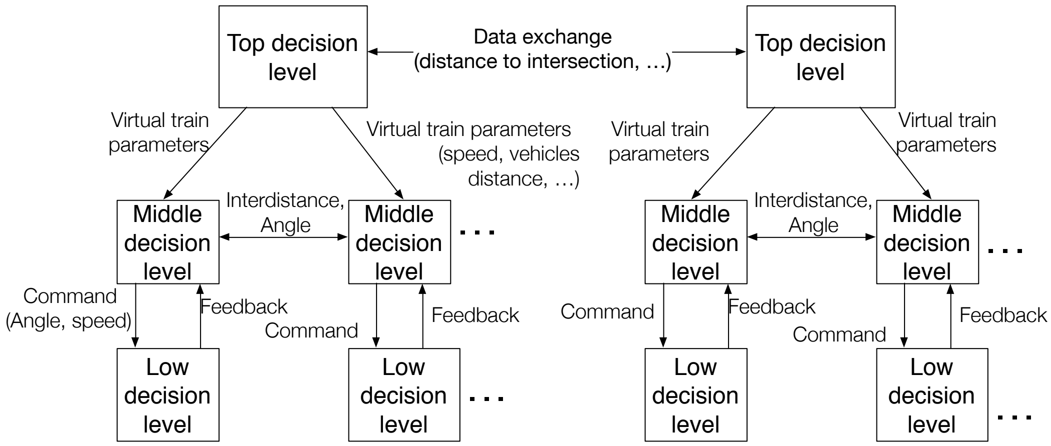



Figure 1: Multi level decision system

perception, multi-level decision processing (e.g. top, middle and low-level decision processor). The perception gives the information of the environment and vehicle status information. Based on the perception, the top-level decision process makes the platoon decision, which guides the whole platoon in crossing the intersection. The decision includes the speed of the platoon and the distances between vehicles. The middle-level decision process,which is the platoon control in this paper, gives demands applied to vehicles. The low-level decision process involves the basic vehicle control (e.g. vehicle dynamic control, motor control). Basic vehicle control models are widespread in the existing literature and are not discussed in this paper.

## 2.1 Coordinate Systems

When calculating the platoon control parameters and inferring the relative position among vehicles, at least two coordinate systems are needed, as shown in Figure 2(b).

1. $O-xy$ system: It is the North-East coordinate system. It is fixed to the earth and the axis $O-y$ and $O-x$ direct to the north and east earth direction respectively.

2. $O'-x'y'$ system:This coordinate system is also called Vehicle-Fixed coordinate system, the direction of $O-y'$ is the same as the vehicle longitudinal axis.

The direction angle of the car is $\theta$. The vector $\vec{XY}$ in $O-xy$ coordinate system could be transformed to $X'\vec{Y}'$ in $O-x'y'$ system by rotation matrix $R(\theta)$ with the following equation: $X'\vec{Y}' = R(\theta)\vec{XY}$

## 2.2 Perception

Perception is the cornerstone of autonomous vehicle algorithms. It aims at retrieving information of the vehicle itself and the environment information surrounding the vehicle. One can also include communicating the exchange with perception ability. Thus, the perception can be classified into the following categories: (1) **Vehicle state perception:** Vehicles gain the information of themselves, e.g. speeds, positions, direction, (2) **Environment perception:** Each vehicle can perceive all necessary information about its surrounding environment with sensors,

(3) **V2V communication:** In this way, the vehicle communicates with others nearby. They can exchange or share information, (4) **V2I communication:** All information is sent to the intersection. After the decision processing, the command is returned to all the cars.

In this paper, each car can, through network communication with others, perceive and get other cars information around the intersection. In this manner, they can infer the relative position and the speed direction.

## 2.3 Top level decision process

The top-level decision process involves the coordination of one or several virtual trains. The top-level process decides the priorities of the virtual trains, the inferred vehicle's distances, the train's speeds, and so on. Figure 2(a) depicts a situation of deciding the priorities, distances and speeds of the related trains. In this paper, the rules of crossing intersection by two virtual trains, each of them contains 2 vehicles, are discussed. We suppose that the transportation systems can run safely and smoothly if all the virtual trains run under the same decision processing.

We decide to use the widespread rule "right vehicle first" and to adapt it to virtual train. Considering that the right virtual train has the priority, each train nearby the crossroad will compute the adapted speed and longitudinal distance to ensure non-collision.

### 2.3.1 Speed limit

In order to avoid vehicles from being crushed, the speed limit is necessary. When the virtual train crosses the road, the speed limit aims at improving the transportation efficiency, avoiding the jam and hence improving safety. Before the vehicles begin to cross the intersection, period when the virtual trains adjust the speeds and vehicles distances is called the adaptation period.

**Speed before crossing** As shown in Figure 2(a), the intersection information could be inferred according to the positions and directions of the two trains. Consequently, it is easy to calculate the distance between the virtual train head and the intersection point. At the time $t_0$, the left heading vehicle has a $s_{t_0}$ distance to the point and the right heading vehicle is $l_{t_0}$. At the time $t_1$; the left train has $s_{t_1}$ and the right train has $l_{t_1}$. Since the right first rule has been chosen. The key point is to adjust the left train speed.

For the left train, it can arrive the intersection point $O$ before the right train finished to crossing. The speed of the right train should be as

$$v_r = (l_{t_0} - l_{t_1})/(t_0 - t_1)$$

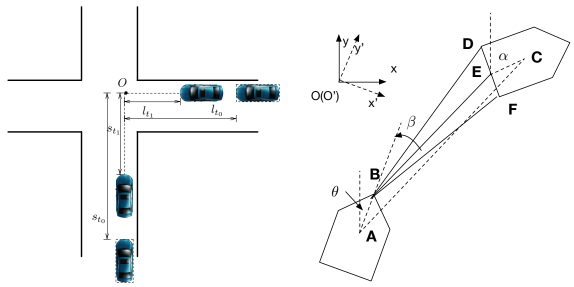
according to the time changing $(t_0 - t_1)$ and the distance changing $(l_{t_0} - l_{t_1})$. Then the right train needs to arrive the intersection point in this time:

$$t_r = l_{t_1}/v_r$$

If let the right train pass first, the max speed of the left train is:

$$v_{lmax} = s_{t_1}/t_r \tag{1}$$

**Crossing speed:** When the virtual trains are crossing, a maneuverable method keeps all the vehicles at the same speed. In one side, the same speed keeps the steady constant distance between vehicles in the same virtual train. On the other hand, the identical speed can avoid disturbance of another vehicle in the transportation system. However, it is also necessary to compute an ideal longitudinal distance between vehicles so as to ensure safety.




(a) Speed calculation (b) GPS information to lidars information

Figure 2: Virtual train information calculation

#### 2.3.2 Nominal longitudinal distance

The nominal longitudinal distance is the distance between the two successive vehicles in the same train. For safety reasons, vehicles should not be too close. However, if the distance is too long, the transportation system efficiency is reduced. Therefore, the distance must change under different conditions. The crossing distance is carried out when the virtual trains cross the intersection.

**Platoon distance:** It is defined so as to ensure safety under various conditions. The distance should vary with the vehicle speed, environment conditions and the vehicle type. For instance, one can take a two-second rule for the following car which means the following vehicle should drive at least two seconds behind the previous one. This rule is adapted depending on the type of algorithm used and as the hardware performances.

**Crossing distance:** When crossing road, there is a minimum distance in order to make sure all the vehicles can cross without being crushed. The minimum distance could be given by the following equation:

$$D_{min} = l + 2w \tag{2}$$

in which, $l$ is the length of the vehicle and $w$ represents the width of the vehicle. One must also take into account the time response of the train in longitudinal distance change. These two computations (speed and longitudinal distance) are sent to each vehicle composing the train.

### 2.4 Middle level decision process

The middle-level decision process concerns the vehicle itself, and outputs the speed and steering angle. The middle-level decision process corresponds to a classical platoon control algorithm. To perform all test, we chose a linear platoon model adapted from the one developed in [6] (cf. Figure 3). The virtual link, between two closed vehicles, is described by a physically inspired interaction model composed of two springs and a damper.

#### 2.4.1 Physical model

The parameters involved in this model are: $m$ the mass of the vehicle, $k_1$ and $k_2$ the stiffness of each one of the springs, $h$ the damping coefficient, $l_0$ the spring's resting length (both springs

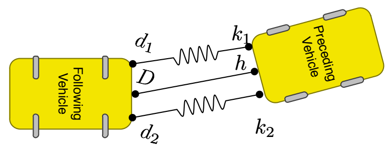



Figure 3: Physical interaction model

have the same resting length), $\nu$ and $\gamma$ the speed and the acceleration of the vehicle respectively.

The following vehicle receives the position of the previous vehicle and itself. Hence, the three distances, $d_1$, $d_2$ and $D$ could be acquired, shown in Figure 3. $d_1$ and $d_2$ are the lengths of the two springs and $D$ is the length of the damper. Hence, the three forces acting on the following vehicle are:

1. Forces of both springs: $\overrightarrow{F_i} = \vec{X_i} = k_i\overrightarrow{(d_i - l_0)}$, $i \in \{1, 2\}$.
2. Force of damper: $\overrightarrow{F_d} = h\vec{\nu} = h(\| \frac{\overrightarrow{\Delta D}}{\Delta t} \|)$.

Using the Newton's law of motion, one obtains a 2nd order differential equation, the resolution of which allows computation of the speed and angle to be applied to the vehicle [6].

The interaction model is used for two main reasons. First, to maintain stable the desired vehicles distance. Second, to guarantee a good trajectory matching, by making the follower vehicle follow the same trajectory as its predecessor, the virtual leader. The parameters of the model are learned so as to ensure safety and stability. The cars vary over time depending on running condition (curvature, speed), see [6] for more details.

### 2.5 Low-level decision process

The low-level decision process is based on the vehicle kinematics, dynamics models and hardware constraints. First, the process should shorten the vehicle response time, e.g. increasing the acceleration. Second, the vehicle motor system is steadied under the low-level decision. In our system, the low-level decision process includes the PI control algorithm to reduce the response time. Plus, the algorithm, developed in [2], limits the step-movement scale for precise and smooth tracking.

## 3 Experiment and Results

### 3.1 Simulation platform

When studying the vehicles dynamics, the simulation system is necessary not only for the safety but also for the economy. The VIVUS (Virtual Intelligent Vehicle Urban Simulator) [3] developed by System and Transportation Laboratory of UTBM is qualified for this work. It can simulate vehicles and sensors within their physical properties. The artificial intelligent algorithms such as platoon solutions and obstacle avoidance have already been tested in VIVUS.

Some parameters are listed as follows: the vehicle width is 1.5 meters, the length is 3 meters, the crossing speed is set to 5 kilometers per hour. Considering the two-second rule, the platoon distance should be 3 meters. After that, the minimum crossing distance is 6 meters according to the Equation 2.

[3] http://www.vivus-simulator.org/Main_Page

### 3.2 Simulation of lidars using GPS information

In the beginning, the following car is navigated by lidars, which detect three distance vectors,i.e., each vector includes angles Iand distance information. Then the platoon control algorithm calculates the stiffness of the two springs, $k_1$ and $k_2$, and the damping coefficient $h$, like presented in Section 2.4.1. However, the lidars can only return information about the closest vehicle. This VIVUS can lead to unsure situation, for instance, when the closest vehicle is a vehicle composing of the other crossing train. Thus, we decided to enhance information provided to vehicles by using GPS data exchange between vehicles.

It is easy to get the GPS information for the vehicles and there is a standard package for the GPS information to the network communication system. After the car receives the previous vehicle GPS information, it simulates the lidars using the two vehicles GPS information, as shown in Figure 2(b). Then the three distance vectors are sent to the platoon controller to make a decision.

**Distance vectors** The position information of the two cars, as same with the two points $A$ and $C$, are obtained through GPS, and the directions of the two cars were sent at the same time. The method of getting the three vectors, called virtual lidars information, is showed in Figure 2(b), where vector $\vec{BE}$ is taken as an example:

In Figure 2(b), $C$ and $A$ are the center points of the previous and following vehicles respectively. The point $B$ is the position of lidar, usually it is in the center point of the front of the vehicle. $D$, $E$ and $F$ are in the tail of the vehicle, being in the left, center and right respectively.

According to the vector operation, the equation to get one of the distance vectors $\vec{BE}$ is:

$$\vec{BE} = \vec{BA} + \vec{AC} + \vec{CE} \tag{3}$$

The unit vector in $y$ direction is $u_y = [0, 1]$, so the rotation matrix is:

$$R(-\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \tag{4}$$

Then:

$$\vec{BA} = -u_y R(-\theta) * l/2 \tag{5}$$

Because the direction of the leading car is $\alpha$, so the $\vec{CE}$ is:

$$\vec{CE} = -u_y R(-\alpha) * l/2 \tag{6}$$

$l$ is the length of the vehicle.

In the same way, the vectors $\vec{BD}$ and $\vec{BF}$ are: $\vec{BD} = \vec{BA} + \vec{AC} + \vec{CD}$ ; $\vec{BF} = \vec{BA} + \vec{AC} + \vec{CF}$

**Coordinate transformation** In order to get the angles form vectors $\vec{BD}$, $\vec{BE}$ and $\vec{BF}$ to $\vec{AB}$, all the vectors are transformed to O'-x'y' coordinate system according to equation using the rotation matrix:$\vec{B'D'} = R(\theta)\vec{BD}$ ; $\vec{B'E'} = R(\theta)\vec{BE}$ ; $\vec{B'F'} = R(\theta)\vec{BF}$

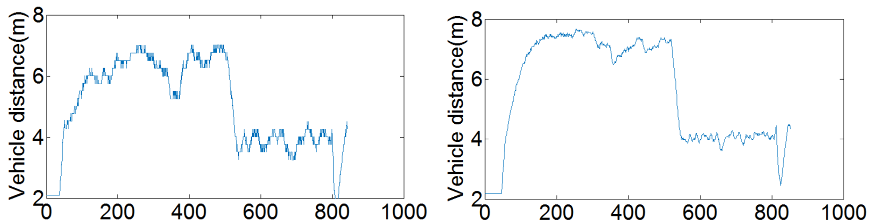
The angles of $\vec{B'D'}$, $\vec{B'E'}$ and $\vec{B'F'}$ to the axis O'-x' are the three lidars angles. If the vector is on the left of the axis O'-x', the angle is negative; otherwise the angle is positive. The three distances of the lidars information are denoted as $\vec{B'D'}$, $\vec{B'E'}$ and $\vec{B'F'}$.

### 3.3 Simulation results

During the test, two virtual train features, distance and passing time, have been measured. The distance should change with the different condition, handed out in Section 2.3.2.

(a) Prepare for the crossing (b) Crossing road (c) After crossing

Figure 4: The distance changing when crossing road




(a) Distance changing in train A (b) Distance changing in train B

Figure 5: Distance changing when crossing

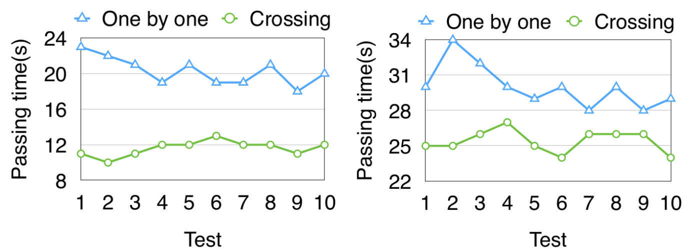
**Distance:** The vehicle distances concern the safety of people and the efficiency of the transportation system. In multi-level decision system, the distance changes with the different situations. Usually, the distance between vehicles is 3 meters, see in Section 3.1. While crossing roads, the distance is changed to the road crossing distance of 6 meters.

The train, running from right to left, is called train A, the another is train B. In Figure 4(a), it shows the situation where vehicles are preparing for the road crossing. In this situation, the distance between vehicles is adjusted to crossing distance, from 3 to 6 meters. In figure 4(b), the crossing road was executed. Vehicles kept the same speed, 5 kilometers per hour, and the crossing distance, 6 meters. After all the vehicles passed the crossroad, vehicles at normal speed and the distance between vehicles was changed to platoon distance, 3 meters, shown in Figure 4(c). The vehicle distances changing of the two trains are also shown in Figure 5.

**Passing time:** In order to evaluate the passing time, two approaches are compared in the simulation: the first is the normal one, i.e., the two virtual trains pass the crossroad one by one even without the time waiting for the traffic light, called one by one in this paper. The second one is developed in this paper called crossing method. In addition, two different crossing distances, 6 and 18 meters respectively, are also carried out so as to compare the passing time in different crossing distance. The 6 meters is the minimum crossing distance. The 18 meters is just a meaningless distance longer than minimum crossing distance.

Forty points, represent forty times test, are shown in Figures 6. The Figure 6(a) gives the twenty results of two method, where the crossing distance is set to six meters. As well as, the results, shown in Figure 6(b), are the results of eighteen meters. For each situation, the average passing times are shown in Table 1.

As show in Table 1, compared with the traffic light control, the crossing method saves time.




(a) Crossing distance = 6 meters (b) Crossing distance = 18 meters

Figure 6: Passing time of each test

Table 1: Mean crossing time

| time(s) \ Method / distance(m) | One by one | Crossing |
|---|---|---|
| 6 | 20.3 | 11.6 |
| 18 | 30.0 | 25.4 |

With the increasing of the distance, the passing time is also increasing.

# 4 Conclusion and Future work

In this paper, the virtual train crossing problem was addressed. A multi decision level strategy based on platoons control was proposed. It contains the perception, top, middle and low decision level. After the vehicles perceive sufficient information, the top decision level gives the command to the virtual trains including the speed of the train and the distances between vehicles. The middle decision level adjust every vehicle pace to fit the top-level decision and to fit the platoon control algorithm requirement. The experiment was also carried out including two algorithms: one was the normal way that the virtual trains pass the crossroad one train after the other, and the second is a crossing method based on the strategy developed in this paper. In an addition, two crossing distance situations were studied. The results proved that the crossing control saves more time than normal method. For the future work, on the one hand, the proposal should be tested with harder conditions, e.g. various numbers of trains and various numbers of vehicles. On other hand, we also thought about using the crossroad so as to make dynamic train reconfiguration, i.e. allowing vehicles to change trains while crossing the intersection.

# References

[1] Jean-michel Contet, Franck Gechter, Pablo Gruer, and Abderrafiaa Koukam. Reactive Multi-agent approach to local platoon control : stability analysis and experimentations. *Int. J. of Intelligent Systems Technologies and Applications*, 10:231–249, 2011.

[2] Baudouin Dafflon, Bofei Chen, Franck Gechter, and Pablo Gruer. A self-adaptive agent-based path following control Lateral regulation and obstacles avoidance. In *2014 International Conference on High Performance Computing & Simulation (HPCS)*, number April, pages 452–459, 2014.

[3] Baudouin Dafflon, Franck Gechter, Jean-michel Contet, Abdeljalil Abbas-Turki, and Pablo GRUER. Intelligent crossroads for vehicle platoons reconfiguration. In Abdelhamid Bouchachia, editor, *International Conference on Adaptive and Intelligent Systems*, Lecture Notes in Computer Science, pages 203–214, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[4] P. Daviet and M. Parent. Longitudinal and lateral servoing of vehicles in a platoon. In *Proceedings of Conference on Intelligent Vehicles*, pages 41–46. IEEE, 1996.

[5] Madeleine El Zaher, Jean-Michel Contet, Pablo Gruer, and Franck Gecheter. Towards a compositional verification approach for multi-agent systems: Application to platoon system. *First International workshop on Verification and Validation of multi-agent models for complex systems*, pages 1–15, 2011.

[6] Madeleine El Zaher, Franck Gechter, Pablo Gruer, and Mohammad Hajjar. A new linear platoon model based on reactive multi-agent systems. In *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, pages 898–899. Ieee, November 2011.

[7] Hans Fritz. Longitudinal and lateral control of heavy duty trucks for automated vehicle following in mixed traffic: experimental results from the CHAUFFEUR project. In *Proceedings of the 1999 IEEE International Conference on Control Applications (Cat. No.99CH36328)*, volume 2, pages 1348–1352. IEEE, 1999.

[8] Audrey Guillet, Roland Lenain, Benoit Thuilot, and Philippe Martinet. Adaptable Robot Formation Control: Adaptive and Predictive Formation Control of Autonomous Vehicles. *IEEE Robotics & Automation Magazine*, 21(1):28–39, March 2014.

[9] Cheng Hu and Yun Wang. A Novel Intelligent Traffic Light Control Scheme. In *2010 Ninth International Conference on Grid and Cloud Computing*, pages 372–376. IEEE, November 2010.

[10] Myung Jin, Jae Weon, Myung Jin Woo, and Jae Weon Choi. A relative navigation system for vehicle platooning. In *SICE 2001. Proceedings of the 40th SICE Annual Conference. International Session Papers (IEEE Cat. No.01TH8603)*, pages 28–31. Soc. Instrum. & Control Eng, 2001.

[11] Mohamed A. Khamis, Walid Gomaa, and Hisham El-Shishiny. Multi-objective traffic light control system based on Bayesian probability interpretation. In *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pages 995–1000. IEEE, September 2012.

[12] Thorsten Luettel, Michael Himmelsbach, and Hans-joachim Wuensche. Autonomous Ground Vehicles - Concepts and a Path to the Future. *Proceedings of the IEEE*, 100(Special Centennial Issue):1831–1839, May 2012.

[13] N. Neuendorf, T. Bruns, and J. Luckel. The vehicle platoon controller in the decentralised, autonomous intersection management of vehicles. In *IEEE International Conference on Mechatronics*, pages 375–380. Ieee, 2004.

[14] Florent Perronnet, Abdeljalil Abbas-Turki, Jocelyn Buisson, Abdellah El Moudni, Renan Zeo, and Mourad Ahmane. Cooperative intersection management: Real implementation and feasibility study of a sequence based protocol for urban applications. In *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pages 42–47. IEEE, September 2012.

[15] Shahab Sheikholeslam and Charles A Desoer. Longitudinal control of a platoon of vehicles with no communication of lead vehicle information: a system level study. *IEEE Transactions on Vehicular Technology*, 42(4):546–554, 1993.

[16] Mohamed Tlig, Olivier Buffet, and Olivier Simonin. Stop-Free Strategies for Traffic Networks: Decentralized On-line Optimization. In *ECAI 2014 - 21th European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS-2014)*, August 2014.

[17] M. Tomizuka. Adaptive vehicle traction force control for intelligent vehicle highway systems (IVHSs). *IEEE Transactions on Industrial Electronics*, 50(1):37–47, February 2003.

[18] Soo-Yeong Yi and Kil-To Chong. Impedance control for a vehicle platoon system. *Mechatronics*, 15(5):627–638, June 2005.