

Traffic congestion reduction based on vehicle platoons and intelligent crossroads interactions

Wendan Du

▶ To cite this version:

Wendan Du. Traffic congestion reduction based on vehicle platoons and intelligent crossroads interactions. Other. Université Bourgogne Franche-Comté, 2021. English. NNT: 2021UBFCA020. tel-03704053

HAL Id: tel-03704053 https://theses.hal.science/tel-03704053

Submitted on 24 Jun 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.





THÈSE DE DOCTORAT

DE L'ÉTABLISSEMENT UNIVERSITÉ BOURGOGNE FRANCHE-COMTÉ

PRÉPARÉE À L'UNIVERSITÉ DE TECHNOLOGIE DE BELFORT-MONTBÉLIARD

École doctorale n°37

Sciences Pour l'Ingénieur et Microtechniques

Doctorat d'Informatique

par

Wendan DU

Traffic congestion reduction based on vehicle platoons and intelligent crossroads interactions

	Thèse présentée et soutenue à Belfort, le 17 December 2021				
	Composition du Jury :				
	QUAFAFOU MOHAMED	PU, Polytech Marseille-Université Aix-	Rapporteur		
		Marseille			
	MESSAI NADHIR	MCF HDR, Université de Reims Champagne-	Rapporteur		
		Ardennes, Reims			
	GALLAND STÉPHANE	PU, Université de Bourgogne Franche-Comté,	Examinateur		
		UTBM			
	ABED MOURAD	PU, Université Polytechnique Haut de France	Examinateur		
		Valenciennes			
	ABBAS-TURKI ABDELJALIL	PU. Université de Bourgogne Franche-Comté.	Directeur de thèse		
NIO		UTBM			
IN					

ACKNOWLEDGEMENTS

The research of this subject has benefited from the support of many aspects.

First of all, I would like to express my deep gratitude to my supervisor Prof. ABBAS-TURKI abdeljalil. It is precise because of his correct guidance that I have made clear the research plan of the subject. In the process of doctoral study and research, the professor gave me warm encouragement, help, and care, so that I can complete the research work with confidence.

Thanks to the Professors QUAFAFOU Mohamed and MESSAI Nadhir who reviewed this thesis in their busy schedule, gave me brilliant comments and came here to participate in the defense of this thesis.

Thanks to Prof. ABED Mourad for accepting to be a member of the committee and letting my defense be an enjoyable moment.

Thanks to the China Scholarship Council for its opportunity and support for this thesis.

At the same time, I would also like to thank the leaders and professors of the Université de Technologie de Belfort-Montbéliard and especialy Prof. Stéphane Galland the Deputy director of the laboratory CIAD for providing me with excellent learning conditions and environment. They gave me strong support during my doctoral study. I would also like to thank the professors KOUKAM Abder and GECHTER Franck for their precious advices at the first stages of the thesis.

Finally, I would like to thank my parents and my family members for their most selfless support and care. I want to share with them the joy of completing my doctoral study. Your presence and understanding was what sustained me this far.

DU Wendan.

CONTENTS

I.	Context et Problems			1
1	Introduction			3
2	State of the art and problem statement			
	2.1	Introd	uction	5
	2.2	Isolated intersection		
	2.3	Traffic lights		
		2.3.1	Control loop of isolated intersection	8
		2.3.2	Traffic light timing	9
		2.3.3	Existing approaches	10
		2.3.4	Inherent traffic light drawbacks with connected and autonomous ve-	
	hicles			
	2.4	Cooperative intersection management		
		2.4.1 Problem statement		
		2.4.2 Protocols : Cruise control		
			2.4.2.1 Stop and Go	18
			2.4.2.2 Reservation	19
			2.4.2.3 Virtual platoon	21
		2.4.3 Scheduling		
			2.4.3.1 Exact and heuristic approaches	24
			2.4.3.2 Policies	25
		2.4.4	Architecture	30
		2.4.5	Discussion	31
	2.5	Concl	usion	32

3 Extended Virtual Platoon

3.1 Introduction				35		
	3.2	Virtual	platoon parameters	36		
		3.2.1	Synchronization point	37		
		3.2.2	Adaptive cruise control	40		
		3.2.3	Cooperative cruise control	41		
		3.2.4	Architecture	45		
		3.2.5	Discussion	46		
	3.3	Simula	ator	47		
		3.3.1	Simulated environment	47		
		3.3.2	Classes and functions	49		
		3.3.3	Metrics	50		
		3.3.4	Simlation purpose	52		
	3.4	Policy		52		
		3.4.1	FIFS	53		
		3.4.2	FRO	53		
		3.4.3	VP-DCP	55		
		3.4.4	Comparison	59		
	3.5	Conclu		60		
4	Optimal virtual platoon 6					
	4.1	Introdu		63		
	4.2	Proble	m statement	64		
	4.3	Optima	al control point	67		
		4.3.1	Preliminaries of optimal and safe trajectory	68		
		4.3.2	Trajectory control design	71		
		4.3.3	Numerical example of optimal trajectory control	73		
		4.3.4	Sequence optimization	75		
	4.4	Optimi	zed cooperative cruise control	77		
		4.4.1	Optimal synchronization point	78		
		4.4.2	Acceleration computation	79		

		4.4.3	Process	. 79
4.5 Simulation				. 80
		4.5.1	OPFIFS	. 80
		4.5.2	OPDCPVP	. 82
		4.5.3	PSODCPVP	. 83
		4.5.4	Conclusion	. 86
П	II Conclusion 89			
5 General conclusion and prospects 91				
Bibliography 93				93
List of Figures 11				111
List of Tables 1				115
List of Definitions 1				117
Aut	Author's publications 11			

CONTEXT ET PROBLEMS

1

INTRODUCTION

Congestion is one of the main concerns of this century. This is due to energy consumption, delay, pollution, and stress. As the transportation demand grows, these problems dramatically increase. In the urban area, congestion is observed at intersections, where conflicting vehicles meet to share common road spaces. Intersection management has taken a big part in research works. From the end of the second world war, several papers contributed to the improvement of traffic management at the intersection, using traffic lights.

Traffic lights helped to improve the traffic condition at intersections in the last century. They allow a higher speed and improve the safety and the throughput of the nodes of the traffic network in an urban area. Many technologies have been introduced to control traffic. First, sensors (g.e. magnetic loops) have been added to measure the traffic. Second, a control loop is implemented to provide adaptive traffic lights. Green and red times are adapted to the traffic conditions. However, with the constant increasing mobility and motorization, congestion shows the limitation of traffic lights. Traffic lights can only manage the flows of vehicles that move on each lane.

The recent progress in the field of connected and autonomous vehicles brings a new way of managing the traffic in intersections. First, these vehicles can communicate with the surrounding environment. So, they can communicate their states such as their origin-destination and their position and speed. In turn, they can get a "green" that fits their current state. Second, they are able to control their movement to adjust their speed according to the received traffic sign. Both communication ability and driving automation allow these vehicles to organize themselves at intersections autonomously. This self-organization capability, at the intersection, was the subject of an active research community, for more than two decades. This new approach for managing intersections has several names, such as autonomous intersection management, cooperative intersection management, or unsignalized intersection.

Cooperative intersection management has a great potential to improve intersection smoothness. First, connected and autonomous vehicles negotiate together to get the

green. This makes the traffic signalization optimization more accurate, by considering each vehicle individually. Second, vehicles adjust their speed at intersection, to avoid as much as possible useless slowdown. This significantly improves the vehicle behaviors, minimizing the time wasted by the integral red and amber lights of the traditional traffic lights.

With this new technology, traffic management requires new approaches. Two main problems are raised by the cooperative intersection management. The first one is scheduling. How can vehicles decide in real-time which one go first and which one go after and so on. Because of the dynamic of the traffic and the hard real-time constraints, many papers rely on rule based system. The second problem is traffic control. Many strategies are proposed in the literature, such as model predictive control, optimal trajectory and so on. However, when the thesis was launched, both problems were almost studied separately. Our main objective is to make the intersection benefit from both improvements brought by connected autonomous vehicles: efficient scheduling rules and suitable cruise control.

This thesis is organized as follows:

First, the next chapter introduces the reader to the traffic management literature. It describes the studied intersection as well as the traffic lights optimization capabilities and limitations to manage an isolated intersection. Then, the chapter gives an overview of the state of the art of cooperative intersection management. The research works are discussed according to the negotiation protocols and scheduling approaches.

The third chapter studies the scheduling optimization of the virtual platoon protocol. It introduces the synchronization point to the reader through the problem formulation of the virtual platoon and presents the used cooperative cruise control. Later, it gradually introduces a rule based policy to efficiently schedule the vehicles. A simulator is built, and new metrics are used to compare the policies. The new introduced virtual platoon coupled with a distributed clearing policy shows a significant improvement of the intersection throughput.

The last chapter intends to gain time between two intersecting movements. An optimal control point is proposed to improve the intersection throughput. This point is studied into an elementary intersection and generalized to a simple ordinary intersection. Results show that the optimal control point improves the intersection speed, but it does not improve the throughput. To overcome the raised problem, a mobile synchronization point is introduced, using distributed particle swarm optimization and simulation results are commented.

STATE OF THE ART AND PROBLEM STATEMENT

2.1/ INTRODUCTION

Traffic congestion is one of the worrying problems of the coming years. In cities, intersections are at the core of the traffic slowdown and the resulting pollution. More than fifty years ago, a special attention has been paid to the optimization of traffic light in order to smooth the urban traffic. Many theories have been applied, such as control theory, combinatorial optimization, multi-agent systems and machine learning. Currently, several research works are focusing on the potential of new emergent technologies, mainly connected vehicles and autonomous driving, for preventing traffic congestion. First, wireless communication is very promising for improving the performance of traffic lights. The Green Light Optimized Speed Advisory (GLOSA) systems allow extending the control loop of traffic light to the speed of the incoming vehicles. Second, connected autonomous vehicles (CAV) are able to negotiate together to pass through the intersection, safely and efficiently.

CAV provides an opportunity to come up with a variety of innovative negotiation protocols at intersection. These protocols aim to establish a bridge between the self-driving objectives of a CAV and the surrounding traffic objectives for coexisting harmoniously and efficiently. On the one hand, the dynamic of each CAV is individualized to respond more flexibly and efficiently to the surrounding. On the other hand, the traffic is kept smooth, allowing each CAV to reach its destination at a minimum cost. More precisely, CAVs at intersection optimize together the following two key parameters:

- Sequence: CAVs negotiate together or all of them follow a given order which one pass through the intersection first, which one the second and so on, in order to free the intersection the soonest or to minimize the exit time of the priority vehicles.
- Speed profile: Each CAV adjust its speed to free the intersection zone the soonest.

The optimization of these two key parameters allows overcoming the current limitation of traffic lights that periodically distribute the green to a given flow of movements. The negotiation between CAVs allows a more accurate share of the intersection by considering each CAV state. However, because of the dynamic nature of the traffic, the optimization of both together is still a challenging issue.

This chapter is organized as follows. First, section 2.2 describes the isolated intersection geometry that will be the object of the study of the whole thesis. Later, section 2.3 reviews the current management solutions for enhancing its performance. The presented solutions focus on the traffic light that is today the key device for controlling the traffic in urban area. After a discussion of the inherent limitations of the traffic lights, this chapter introduces the cooperative intersection management, in section 2.4. The chapter discusses the literature contributions in terms of protocols of sharing the common spaces, the scheduling algorithms and the communication architecture to achieve the negotiation. Finally, the chapter ends with a conclusion that precise the starting point of the work.

2.2/ ISOLATED INTERSECTION

An isolated intersection is two or more roads (see the example presented in Figure 2.1 A) that meet in a conflict zone. A road is characterized by its length, its number of lanes, as well as its traffic directions. The example shows an intersection of 4 roads, with one lane for each traffic direction on the road. A movement is defined by its origin and destination. An approach is all the movements of vehicles that come from the same origin. The movements that belong to the same approach are admitted simultaneously without conflict and receive the same traffic signal. Some lanes may be dedicated to a particular movement, such as turning left. These lanes need to be considered separately. An intersection may consist of several diagrams relating to the geometry of the infrastructure and the crossing rules. However, in each intersection, we can identify three functional areas (see Figure 2.1 B):

- Conflict Zone (pink zone): In this zone, the movement trajectories intersect. In other words, it is the critical resource shared by all vehicles crossing the intersection, with a high potential of collision.
- Storage Zone (blue zone): It is located upstream of the conflict zone. It is the road before crossing the intersection.
- Exit zone (green zone): Downstream of the conflict zone. It allows the relief of the conflict zone.

A conflict point (see red points) results from the meeting, in the same place, of at least

two movement trajectories with a non-zero angle. The conflict zone of any intersection includes all the conflicting points. Conflict points are the potential area of the collision. For safety reasons, two vehicles coming from different lanes should not access to the conflict point simultaneously. Some conflicts between vehicles find a solution in the traffic regulation, such as priority to the right. However, conflicts must be carefully studied according to the field of view of the drivers and the importance of the conflicting movements in terms of traffic flow and speed limit. In many cases, some conflicts must be systematically eliminated, using traffic light signals. This is achieved through the implementation of traffic lights visible to the drivers from the storage zone. The traffic light is located at the border of each storage zone with conflict zone (between blue and pink zone). The principle is to provide different time access to the conflict zone to the major conflicting movements.



Figure 2.1: A- A top view of a four-way intersection, B- Illustration of the three functional areas and the terminology used to analyze the intersection: 4 approaches, 12 movements and 20 conflict points.

2.3/ TRAFFIC LIGHTS

The main purpose of traffic light is to improve the safety of intersections. As they can be controlled periodically or instantaneously, they were the subject of several research works that aim to optimize the traffic. Many techniques were used to optimize the traffic management through traffic lights, since the beginning of the last century [1], such as optimal control [2], metaheuristics [3], fuzzy logic [4], deep policy-gradient [5] and colored Petri nets [6]. The next section introduces the traffic enhancement through traffic lights management and discusses the limitations of this control approach.

2.3.1/ CONTROL LOOP OF ISOLATED INTERSECTION

Traffic management is carried out through the intersection control loop (see Figure 2.2). The loop is composed of intersection and the related signal system (traffic lights, priority panels, sensors...). The intersection is characterized by its geometry, movements and the corresponding demand. On the premise of ensuring traffic safety and smooth flow, the corresponding signal system is selected according to the city's policy. The signal system based on traffic light includes the following devices:

- Signalization system: It refers to a group of vertical and horizontal signals displayed to the driver to ensure safety. They are designed to avoid, as much as possible, conflict movements to cross simultaneously the conflict zone. This is mainly achieved through the traffic lights that the drivers observe from the storage zone.
- Measurement infrastructure: It reflects traffic conditions at different scales. The traffic measurements mainly give the traffic flow (number of vehicles per unit of time), the occupation rate (percentage of occupancy time of given loop) and the average speed of vehicles. These values are measured by sensors installed at the infrastructure, such as magnetic loops (embedded rectangles in the road presented in Figure 2.2).
- Controller: it is the brain that controls the traffic. According to the data get from the measurement infrastructure, it computes the scheduling strategy in real-time to achieve a precise goal (specifications) such as queue length and waiting time minimization.



Figure 2.2: Illustration of the control loop component through a four-way intersection.

The signal system is the visual part of the studied system. Among other horizontal and vertical permanent signalization systems, traffic lights should be installed at every place

authorized to enter the conflict zone. Therefore, in the storage area of the intersection, all vehicles can clearly see each traffic light. In general, the characteristic of traffic lights is that one color is illuminated at a time. The color of the light signal is either green, yellow or red. The meaning of each color is as follows:

- Green: green signal light allows vehicles to go carefully.
- Yellow: every driver should mark a stop in front of a fixed yellow signal as far as possible because it indicates the red light is coming soon.
- Red: Access to conflict zone is not allowed.

The illuminated colors define the state of the traffic light.

2.3.2/ TRAFFIC LIGHT TIMING

Most of the research efforts to improve traffic flow safety and efficiency focus on traffic light timing. Traffic light timing means the duration computation of each traffic light state, the selection of the suitable sequence of these states and their synchronization with other traffic lights. According to the city's policy and relevant authorities, traffic light timing can have several objectives. For example, the city's policy can aim to promote soft transportation modes, public transport or specific corridors. However, these choices must be in line with local travel characteristics. In other words, the implementation of traffic light timing should not result in congestion. Hence, traffic efficiency is always the main purpose of traffic light timing, after the safety purpose.

First, before the traffic light timing, phases are defined. This is called **stage specification**. Each phase allows a set of movements to pass through the conflict zone together. To this end, a matrix of conflict is computed. Each pair of conflicting movements are analyzed from the dangerousness standpoint. There are several levels of dangerousness. At the highest level comes the orthogonal intersecting straight movements. These movements should belong to different phases to get into the conflict zone separately. The intersection of turning movements might be considered less dangerous. These lower-level conflicts can be solved through traffic regulation rules such as the priority to the right. In such a case, movements must be studied carefully before allowing them to cohabit in the same phase. An example of phases of the intersection, presented in Figure 2.1, is given in Figure 2.3. In this figure, only the highest levels of dangerous conflict are eliminated through the phases' definition, whereas the left-turning movement needs to respect the rule of priority to the right.

The most simple traffic light timing in an isolated intersection is periodic. The period is called a cycle time that refers to the duration that the traffic lights need to back to the initial



Figure 2.3: Example of phases' definition of a 4 way-intersection.

state (see Figure 2.4). The computation of **the cycle time** raises the issue of finding a compromise between the average waiting time and the intersection throughput [7]. This is due to the fact that there is a lost time at each transition from one phase to another. The lost time includes the time for the clearance of the conflict zone and the acceleration time of the first vehicles that get the green. Moreover, because of the yellow light, some vehicles stop before the end of the stage. Hence, the effective green is the sum of actual green time, yellow time (actual green) and integral red time for clearance minus the lost time. As the cycle time gets longer, the vehicles need to wait more time to get the green but, simultaneously, the ratio of effective green time to the cycle time becomes bigger, which increases the throughput. Figure 2.4 illustrates the cycle analysis. Once the cycle length is computed, the time of the effective green given to each phase is proportional to the demand of each phase. This is called traffic light **split**.

2.3.3/ EXISTING APPROACHES

With the stage specification, the cycle time computation and the split, the traffic light timing is determined. Many approaches are used to adapt the traffic light timing to the traffic demand. The following folds recall the most important ones:

• Daily fixed time plan [8]: This control method is based on regular variations in traffic demand. Traffic demand variations during the day are studied. A fixed cycle traffic light is calculated according to the corresponding traffic demand at a given period of the day. For example morning peak hour, the off-peak hour of the day, evening peak hour and night off-peak hour. The fixed time cycles are then applied at



Figure 2.4: Cycle time analysis

the relevant time. This approach can be extended to consider real-time variations in traffic demand. For example, a fixed cycle time is chosen with the nearest assumed demand to the current measured traffic demand.

- Vehicle-actuation [9]: This control method is mainly used for isolated intersections. Minimum and maximum green time are computed for each phase. As long as vehicles are detected by the electromagnetic loop at a given time interval, the green is extended, until the maximum green time is reached. More sophisticated approaches are used to extend the vehicle-actuation approach in order to consider the synchronization with other intersections [10].
- **Real-time optimization** [9]: Several approaches consider the responsive traffic light timing as an optimization problem. The problem formulations as well as the used techniques are different according to the authors. One example of the considered approach is the one proposed by the authors of PRODYN [11]. They use dynamic programming coupled with the Model Predictive Control to decide whether the switch from one phase to another must occur. Other similar approaches [12, 13, 14] differ in terms of sampling time, optimization and the chosen rolling horizon approach. Another example is given in [15]. It consists in using a queue theory and reinforcement learning to choose the best policy, according to the observed traffic state. The real-time optimization approaches are either applied to an isolated

intersection or a network of intersections, such as the ones discussed in [16, 17].

Note that the computation of the traffic light timing requires solving two problems: traffic demand forecasting and traffic performance optimization. The traffic light timing is still of interest to many research works. In particular, machine learning-based techniques for control and for predicting traffic are being developed [18, 19] in order to improve the traffic safety and efficiency. In addition, the complexity of city policies, such as adding bus priority [20] or saving energy [21], makes the traffic signal timing problem particularly challenging.

2.3.4/ INHERENT TRAFFIC LIGHT DRAWBACKS WITH CONNECTED AND AU-TONOMOUS VEHICLES

Traffic lights contributed to significantly improve urban traffic conditions. The various techniques developed by researchers to specify stages and to compute the responsive durations of cycles and phases allow obtaining safer intersections with higher speed and throughput. First connectivity allows to include the vehicle in the control loop. The Green Light Optimized Speed Advisory (GLOSA) [22] informs the vehicle about the traffic light timing. This allows to display to the driver the optimal vehicle's speed to reach the green, avoiding useless stops and encouraging Eco-Driving. However, the traffic light mechanism suffers from several problems related to the lack of responsiveness [23, 24] and accuracy that we discuss hereafter. These traffic light limitations are raised with the perspective of using connected and autonomous vehicles (CAV). Here are the most important ones:



Figure 2.5: Cycle time analysis

• **Hard coded phases**: Once the stage specification is computed, it is not possible to introduce new phases according to the current traffic demand. The signalization

system and the measurement infrastructure are expensive hardware devices that are implemented for some years, according to the stage specification. The responsiveness of the traffic light is then limited to either the extension of the duration of the current phase or to the choice of the next predefined phase. No new phase can be defined dynamically. Figure 2.5 gives an example of this situation. Assume that the phases are defined according to Figure 2.3 A, CAVs coming from the road R_1 can safely cross the intersection, but they get the red color.

- Phase safety issues: Because of the lost time when switching from one phase to another and infrastructure limitations, the number of phases in a cycle is limited. As a result, some conflicting movements are admitted in the same phase (e.g. turning left with the opposite going straight). Even if the meeting of these movements should not be highly dangerous, this does not avoid collisions [25]. To solve the problem, advanced driver-assistance systems and new expensive layouts of intersections are needed [26, 27].
- Inaccurate signaling system: The traffic light is seen by all vehicles coming from the same approach. The number of vehicles that can cross the intersection isn't deterministic. Some vehicles should leave the intersection, but they don't. Similarly, some vehicles go through the intersection, but they shouldn't. From Figure 2.3 B, it is impossible to predict whether the third and the fourth vehicles will pass. If both vehicles leave the storage they block the box junction.
- Traffic control: Traffic light timing approaches are based on the expected traffic demand. They almost use the historical data of the traffic demand to cross the intersection. Even vehicle-actuation approach requires to steady the time headway at each approach. Moreover, the phase extension is based on the flow (number of vehicle per unit of time) and the lane capacity (maximum flow ¹). flow measurement can not accurately give the current traffic situation. For the same flow, traffic is either smooth or congested [28]. Besides, the lane capacity is only an estimation. It varies according to the time of the day, the sampling time, types of vehicles and their movements. In traffic engineering, passenger car equivalent (PCE) is used to get an average estimation of the capacity [29, 30]. However, this need to be thoroughly studied according to the behavior of the traffic at the controlled intersection [31].

From the details of the four highlighted limitations of the intersection management system based on traffic light, one can note that CAVs allows a new intersection management paradigm that solves them all. The novelties are brought by the driving autonomy coupled with the connectivity. Both autonomy and connectivity allow CAV to communicate their destination earlier, such that in the situation of Figure 2.3 A, CAVs that turn left can pass

¹Maximum here isn't the maximum maximum but an average of maximum

through the intersection. Also, they can receive the signal, so that only allowed CAVs can cross the intersection. In the example given in Figure 2.3 B, only the two first CAVs in R_4 are allowed to go, to avoid blocking the box junction. More precisely, the phase logic to manage the intersection can be removed, which avoids simultaneous green given to conflicting movement and increases the opportunity of non-conflicting movement to go together.

The other advantage of CAVs is the minimization of the lost time between conflicting movements. The connectivity allows CAVs to inform the others when they exactly clear the conflict points. This avoids the estimation of the integral red (see Figure 2.4) that is designed to cover the theoretical time needed to clear the conflict zone. Thus, as soon as the intersection is clear, the CAVs get into the conflict zone without waiting for the end of the integral red. When the conflict zone is empty, CAVs must simply negotiate together to avoid simultaneous access to conflict movements.

From the previous discussion, CAVs allow **theoretically** much more accurate intersection management and also safer, by negotiating together the right of way : Which one can go first and exactly when it is possible. With the premise assumptions of connection and driving autonomy hold, there is no need for traffic lights. We call, in the following, this kind of intersection management based only on the negotiation of the right of way between CAVs, Cooperative Intersection Management (CIM). CIM brings at least the following novelties:

- CIM allows to precisely decide which vehicle has the right-of-way and addresses the right-of-way to only these vehicles rather than decide the duration of phases or decide whether the phase is still kept for another period of time [12].
- The allowed CAVs to pass through the intersection is computed according to the actual traffic demand instead of using prediction of the traffic volume based on the past statements of the inductive loops.
- CIM facilitates the deployment of priority policies between vehicles because each vehicle is considered individually.

CIM is described in the next section, by considering the most noticeable contributions.

2.4/ COOPERATIVE INTERSECTION MANAGEMENT

CIM deals with CAV intersection. Even if some papers consider ordinary vehicles [32, 33, 34, 35], the intersection of only CAVs is still an open topic and many contributions are ongoing to provide an optimized CIM. The capabilities of CAVs to communicate with

their neighbors and to control their trajectory have been the focus of research to bring many improvements to the traffic in terms of safety and efficiency. These capabilities offer unprecedented potential that significantly outperforms current approaches used to manage intersections.

Firstly, this potential is offered by wireless communication that allows CAVs to predict future obstacles. Indeed, communication allows the observation of the trajectories of CAVs that are not in the field of vision (hidden behind a building or enough far away to be detected by the sensors). By overriding the limitation of the existing traffic rules, CAVs can also wirelessly schedule together with their passage, before they arrive in the conflict zone.

Secondly, the driving autonomy, if well exploited, allows a significant minimization of the time lost between the passage of two conflicting movements. The CAV can plan to enter the conflict zone when it is free. Also, it can avoid unnecessary stops in the storage area, if CAV cooperates well with the other CAVs. Thus, the concept of the right of way is more complex than the one used in daily traffic control systems. The right of way can be expressed as a red or a green, but can also be extended to a more complex signalization system that challenges CAV longitudinal control. For example, CAV receives a right of way that allows it to access to the conflict zone after 10 seconds. The CAV uses a complex longitudinal control to respect it.

With this huge potential, the work in the literature differs not only in terms of the techniques used by the authors but also through the concept of cooperation between CAVs. This diversity makes it particularly difficult to give an exhaustive overview of the CIM literature. To overcome this difficulty, we give a definition of CIM that covers the large majority of works, and we classify the assumed concept, according to the addressed problems.

2.4.1/ PROBLEM STATEMENT

In order to contextualize our work as well as the ones presented in the literature, we use the definition 1. This definition extends the one given in [36]. From this definition, there are two problems that need to be solved by CAVs and/or intersection manager when a CAV gets into the storage zone of the intersection (see Figure 2.1-B). These two problems are as follows:

- **Scheduling**: CAVs and/or the intersection manager must decide wich CAV pass through the intersection first, which one the second and so on.
- Longitudinal control: CAVs and/or the intersection manager must decide the speed profile that allows CAVs to respect safely the schedule.

Both scheduling and longitudinal control are subject to optimization. However, the theories behind both optimizations are different. Scheduling optimization uses the theories of combinatorial optimization, whereas the longitudinal control of CAVs is mainly based on the optimal control theories of continuous or discrete-time systems. Solving both optimization problems together remains an open challenge, for several reasons. One is the dynamics of the traffic system. When a new CAV arrives, it questions both already computed sequence and longitudinal control. The other one is real-time constraints because of vehicles move forward during the computation time. Hence, both problems needs to be solved under a very short time. To remedy these problems, first works in the literature tries to find a balance between optimal solution and computation time, as well as a balance between scheduling and longitudinal control. The various CIM concepts can then be described first through the approach they use to deal with each of both problems, i.e. scheduling and longitudinal control. They are also different from the way in which computation and negotiation are carried out.

Definition 1: Cooperative Intersection management

We call CIM, the CAV traffic management at a single intersection that consider all the following features together:

- The accesses to the intersection are ordered (sequence) according to the requests received from the vehicles via wireless communication.
- Each vehicle individually receives its own right-of-way, with respect to the sequence
- Each CAV respects the received right of way by performing the suitable longitudinal control.
- Each vehicle instantaneously participates in the decision-making process by requesting the right of way and/or by communicating its current state.
- Regardless of technological and practical limitations, such as the road capacity or the restriction of the wireless communication, the traffic management at intersections is able to consider an unbounded number of vehicles without preprogrammed schedule.

In order to introduce the CIM literature, we use the following classification items:

- Protocol: The protocol refers to the way in which two conflicting CAVs share the common space to plan their successive passages. This defines the longitudinal control problem of the second vehicle in order to avoid collision with the former one.
- Policy: It defines the way in which the passage sequence is negotiated between the

CAVs under real-time constraint.

 Architecture: It refers to the level of involvement of CAVs in the decision-making process. The level varies not only upon the problem statement but also upon the assumed safety level.

Before discussing each item according to the different approaches in the literature, it should be noticed that compared to the abundance of theoretical work on CIM, real tests remain few. Despite this contrast between theory and practice, tests on robots, mixed reality tests and demonstrations on real vehicles allow us to have a glimpse of the flaws of some hypotheses. In the following, we consider the documented real tests presented in Table 2.1.

References	test	vehicles	Number of vehicles
[37]	8-shaped intersection	mini robots	6
[38]	8-shaped intersection	fully automated vehicles	2
[39]	4-way virtual intersection	fully automated vehicle	1
[40]	4-way intersection	Robots	4 (1 per lane)
[41]	2-way intersection	Ordinary connected vehicles	4
[42]	2-way intersection	robots	3
[43]	8-shaped intersection	fully automated vehicles	3
[44]	4-way intersection	Robots	4 (1 per lane)
[45]	4-way intersection	Robots	Unlimited
[46, 47]	4-way intersection	fully automated vehicle	3 (1 per lane)

Table 2.1:	Real t	tests of	CIM
------------	--------	----------	-----

2.4.2/ PROTOCOLS : CRUISE CONTROL

Many protocols have been proposed for negotiating the successive passages of two CAVs. Figure 2.6 shows the most important protocols in the literature, through a simple intersection. They are mainly as follows:

- Stop and go: The second CAV receives a stop sign until the first one leaves the conflict zone (see Figure 2.6 A).
- Reservation: The first CAV sends the time when it leaves the conflict space and the second one manage to get through the space later, through a given cruise control approach (see Figure 2.6 B).
- Virtual platoon: The second CAV considers the first one as a virtual obstacle and adjusts its speed accordingly (see Figure 2.6 C).

Some protocols seem more complex than the above classification, in particular, the one proposed in [48]. More precisely, this protocol split the storage zone into two successive

areas: deceleration and acceleration areas. The latter allows CAV to cross the conflict zone at maximum speed. This protocol is a mix of virtual platoon and reservation. First, it belongs to the virtual platoon because CAVs receive deceleration instructions in order to avoid collision with the precedent ones. Second, because the maximum speed allows a known occupancy time of the conflict zone, CAVs are managed according to the expected time of the free conflict zone. Except a few papers, to the best of our knowledge, most of the literature works can be easily classified in one of the three families of protocols given above.



Figure 2.6: Illustration of the three main protocols to share the intersection. A-Stop and go, B- Reservation, C- Virtual platoon : Conflicting CAVs are considered as virtual obstacles

2.4.2.1/ STOP AND GO

This is the first proposed protocol [49]. It is analogous to the ordinary traffic light. Indeed, we can map "stop" to the red light and "go" to the green. It is different from the traffic light because each CAV negotiates its own green and gets it individually. In practice, this protocol improves the traffic light efficiency by avoiding useless red and by exploring all possible phases (see Figure 2.7). Despite its simplicity, it is necessary to define some of its features. For example, in the first version of the stop and go protocol, some details are missed. In [49] vehicles calculate their priority dynamically. The vehicle with the highest priority passes, i.e. gets the green. However, this assumption has to take into account the safety issues linked to vehicle dynamics. It is difficult, even dangerous, to lose permission during the CAV movement in the storage area. For example, a vehicle that has the red at the last seconds may not be able to stop before the conflict zone. More details about the stop and go protocol are given in [50, 41]. In [50], the authors assume a V2V (Vehicle to vehicle) communication, with a green by default. When the vehicle is alone, it has the green. Otherwise, when the vehicle detects conflicting vehicles, if it is the leader or is alone on the line, and it is able to stop, it has the red, and it becomes a server to distribute signals to the new incoming vehicles from the other lanes. The vehicles that receive the



Figure 2.7: Example of adaptive phase with Stop and Go protocol (B) compared to static phase of traffic lights (B)

traffic signal, i.e. green or red, must respect it.

For safety issue, in [41], the authors use an external server: V2I (Vehicle to infrastructure) communication. The server is only used to propose a default deny policy. That means, when a CAV enters the storage zone, it has the red by default. Then, it needs to negotiate with the other CAVs through the server later, to get the green. If there are no conflicting vehicles with the green, the server gives the green to the first request.

Both protocols consider two kinds of vehicles. They are used for traffic of CAVs [51, 52] as well as for traffic of ordinary vehicles [53, 54, 55]. For ordinary vehicles, the color is displayed on the screen inside the vehicle. For CAVs, the red is the maximum distance that the CAV can travel (movement authority). The CAV must control its trajectory to come to a complete stop before the end of the movement authority.

2.4.2.2/ RESERVATION

The reservation protocol aims to fully use the potential of driving automation and connectivity. Rather to get simple stop-and-go signs, the CAV gets the time when it is allowed to access to the conflict zone and respects it. Regardless of the used approach, in the reservation protocol, the input control of the CAV is the time of entering and exiting the shared spaces. The reservation protocol was first introduced by Dresner and Stone in [56]. In this version, the conflict zone is split into tiles which are shared between CAVs. The authors use a multiagent approach to manage the intersection. Each CAV books the tiles needed to reach the destination. The reservation time is computed according to the position and the speed of CAV. If tiles aren't available (already booked), the CAV updates its reservation request, until it gets its reservation. The main advantage of the approach is the small time-interval between two passages of conflicting CAVs. However, the reservation protocol raises the issue of respect for the time schedule. To this end, cruise control was studied in order to make CAVs meet the reservation time. Hence, the reservation protocol requires performing two tasks: The time-schedule of the CAV passages through the shared spaces and the longitudinal control to meet the due times. The time scheduling is discussed later in Section 2.4.3.

In [57], the cruise control is based on the computation of the time-velocity diagram with 3 pieces of a linear function of velocity. Each piece of the linear function is adjusted to meet the reserved time. A more complex cruise control was studied in [58], by using Reinforcement Learning with discrete acceleration and speed values. More precisely, CAVs are trained according to the different scenarios of positions and speeds. In [59], the authors use Pontryagin's maximum principle, to minimize the control effort. The Hamiltonian is used to compute the optimal trajectory. In [60], the authors use a non-linear programming (NLP) based on the infinitesimal method to minimize the energy consumption. A rolling strategy is used to overcome the problem of the schedule feasibility during the CAV movement and the update of the entrance time because of the new incoming CAVs. Moreover, many papers use Model Predictive Control (MPC) to compute the speed profile, such as in [61, 46, 62, 63, 47]. The MPC is used for two stages. First, it is used for negotiating the access time to the common space. Second, it is used to compute the speed profile, allowing the respect of the scheduled time.



Figure 2.8: Potential collision area

There are three kinds of common spaces considered in the literature. The first one is the set of tiles of the conflict zone. The conflict zone is split into several squares called tiles, with a given resolution. The CAVs book the expected occupancy time of tiles. The key rule is that one tile cannot be occupied by two CAVs simultaneously. The second is the conflict points inside the conflict zone (see Figure 2.1-B). Two approaches are used to model the conflict point. Some authors consider it as a point so that the reservation consists of the passage time plus a safety time, as in [63]. Other works consider, instead, the intersection surface of vehicles itineraries, as in [47, 60]. We call it, in the following, the potential zone of collision (see Figure 2.8). As for tiles, CAV books the occupancy time of the conflict potential zone of collision. More conservative approach makes CAVs reserve the whole conflict zone, as in [59]. CAV reserves the occupancy time of the zone. In such a case, the time constraint to book the conflict zone depends on whether the precedent CAVs are conflicting or not.

The main advantage of the reservation protocol is that it allows the optimization of the speed profile. In [59], the authors minimize the control effort $\frac{1}{2} \int_{t_0}^{t_f} u(t)^2 dt$, where u(t) is the CAV acceleration. Other authors minimize the deviation from the reference velocity [64]. The energy consumption was also used as a criterion to compute the optimal cruise control [65, 66, 35, 67]. However, one of the most important raised issues with the reservation protocol is the intersection safety. Indeed, if a CAV does not respect the time, a collision will happen [68]. This issue worsens when the speed of CAV in the booked common space is difficult to determine [38]. For instance, a CAV that meets congestion at the next road needs to slow down and thus delays its exit time. One solution to this problem was proposed in [48], where the storage zone of the intersection is split into three zones: A high-speed zone at the exit of the intersection, a control zone in the middle of the storage zone and another high-speed zone at the end of the storage zone to get through the intersection. Hence, the conflict zone is used at the maximum speed. However, this requires more lanes for considering the turning movements and a longer storage zone. Other solutions were designed according to real tests. In [39], the authors use a buffer time when CAV reserves tiles as well as a low tile resolution (bigger tiles). In the tests described in [47], the authors add safety distances upstream and downstream of the booked potential zone of collision. These safety distances allow CAV to perform an emergency braking.

2.4.2.3/ VIRTUAL PLATOON

This protocol was initially introduced by [69] and tested by [51] through real robots. It was first called Transparent Intersection management. After the contribution of [70], it is currently known as a virtual platoon protocol. In order to overcome the safety issues raised by the reservation protocol, instead of using a time schedule, CAVs use sequences to control their speed. The sequence gives which CAV goes first, which one goes after, and so on. CAVs share an ordered list of presence that gives the sequence. In the list of presence, each CAV periodically writes its current position and speed, as well as

its destination. Accordingly, CAV identifies, from the presence list, the precedent CAVs as obstacles. The virtual platoon protocol works as follows. Each CAV considers all precedent conflicting vehicles as obstacles (virtual obstacles). So, there is a position mapping allowing to consider the conflicting CAV like they are moving in the same lane. Figure 2.9 gives an example of the mapping. In [51], the stop line is considered by the CAV while it cannot keep a safe distance with the virtual obstacles. This makes CAV resort to a complete stop if it is not safe to cross the potential zone of collision. The cruise control of the CAV allows them to keep a safe distance before getting through the potential zone of collision.



Figure 2.9: Example of virtual platoon, where the CAV considers the real obstacle (rl) virtual obstacles vl, includind the next leader nl and the stop line sl. nl is the last CAV that moving in the distension lane [71]

In [69], the authors use Gipps model [72, 73] to control the simulated virtual platoon. Later, in [51], simulations of virtual platoon are based on an enhanced IDM (Intelligent Driver Model) [74] to compare virtual platoon with stop and go protocol. In [75, 76, 43], the authors introduce RT-CVC (Reaction Time-based Collaborative Velocity Control) [77] to consider communication delay and driving automation time. Recently, more usual control techniques are used. In [70, 78, 79] the authors use linear cruise control with respect to errors, whereas in [80] the authors use sliding mode controller. Simulations show that the virtual platoon is more efficient than the stop and go protocol. They show also that the virtual platoon is less efficient than reservation [80]. However, both simulation results need to be thoroughly studied according to safety constraints (See Section 2.4.2.2) and the scheduling algorithm, as it will be detailed in Section 2.4.3.

2.4.3/ SCHEDULING

The protocol determines the way CAVs share the common spaces, either by yielding the way, by booking or by determining a sequence to form a virtual platoon. It is obvious that the protocol contributes to the performance of the intersections. However, this is not sufficient to totally explore the potential of wireless communication, mainly when the traffic flow is high. Indeed, from the fundamental traffic diagram, when the traffic flow goes beyond the capacity, the CAVs will come to a complete stop [81]. Scheduling is the

other key for improving the intersection performance.

The scheduling problem of CAVs in the intersection is a combinatorial optimization problem that needs to be studied and optimized. First, some assumptions are formulated to model the problem. Most existing studies prohibit overtaking for two CAVs that move in the same lane. So, the order of arrival at a given lane is conservative. More recent studies [82] introduced overtaking, but it splits the scheduling optimization into two stages. The first one is for changing the order of CAV exits and the second one is for the scheduling CAVs according to the new order. Mathematically, the sequence conservation allows the minimization of the number of combinations as follows:

$$\frac{\left(\sum_{i=1}^{L} n_{i}\right)!}{\prod_{i=1}^{n} (n_{i}!)},$$
(2.1)

with *L* and n_i are the total number of lanes in the storage zones and the number of CAVs in the lane *i*, respectively. In the equation 2.1, the quotient is owed to the order conservation. Even with the quotient, the equation 2.1 shows a combination explosion. The other assumption concerns the time when the CAV is ready to cross the conflict



Figure 2.10: Lost time computation when the traffic light switches to green

zone. This time depends on the assumed protocol. This time was first addressed by using traffic engineering theories to define the headway time, according to the state of the precedent CAV. The time lost when the traffic light passes to the green [83, 84] is used in the stop and go protocol [85, 54] (see Figure 2.10). The microscopic simulation [86] is used for the scheduling virtual platoon protocol [69, 75]. Finally, time constraints and/or distance with precedent CAV are defined in the reservation protocol. Note that there is no consensus on security constraints in the reservation protocol. Indeed, it strongly depends on the assumptions made about the ability of CAVs to respect their reservation [39]. Moreover, some authors assume a lower time between conflicting vehicles than the time between two following vehicles that move on the same lane [63]. This leads to

significant performance gains. However, this assumption is highly questionable.

Two approaches to schedule the intersection are proposed in the literature:

- Exact and heuristic approaches: Depending on the arrival times of CAVs, their speeds and positions, the intersection server calculates the optimal (near optimal) sequence and time for getting into the common space.
- Policy: Because the intersection is dynamic and because of the real-time constraints, simple rules are defined to allow CAV to get the right of way according to the protocol (see Section 2.4.2).

2.4.3.1/ EXACT AND HEURISTIC APPROACHES

In order to address the complexity issue, [85] introduces a dynamic programming algorithm that solves the C_{max}^2 scheduling problem of cooperative intersection in a polynomial time and memory space according to the number of CAVs. However, the computation time and the memory space are exponentially increasing as the number of lane increases. The complexity of the dynamic programming is $O(n^L)$, where *n* is the total number of CAVs ($n = \sum_{i=1}^{L} n_i$)³. The proposed dynamic programming models the problem by a directed weighted graph [87] and solves it, using the well-known shortest path algorithm is that the nodes and links of the graph can model several assumptions of CAV cruise control and headway [69] as well as priority CAVs [92, 93]. Another dynamic programming algorithm was suggested in [94]. It aims to minimize the number of nodes by putting together CAVs that can cross the conflict zone simultaneously. However, its complexity is still $O(n^L)$.

From the scheduling theory, *C_{max}* is the completion time of the last job (CAVs) in the system. Other more complex objective functions exists such as the total weighted completion time, the average waiting time or the maximum lateness [95, 96, 97, 98]. To solve the problem with the other objectives, the authors resort to mixed integer linear programming (MILP) model [99]. For instance, in order to minimize the total travel time delay, [100] seeks the optimal vehicle scheduling at a multi-conflict area, considering heterogeneous vehicle headways and values of time. A MILP model is proposed to solve the exact optimal solution to this problem. Although small instances of the proposed model can be solved by existing commercial MIP solvers, their computational time increases exponentially as the numbers of vehicles and approaches increase. A similar approach was used

 $^{{}^{2}}C_{max}$ in the context of intersection scheduling means freeing the soonest the storage zone and the conflict zone, with a given set of CAVs in the storage zones.

³A detailed analysis of the dynamic programming is given in [75]

in [60] to schedule the reservation of the potential collision area. The optimal scheduling is used for both following purposes:

- Comparison with the other scheduling approaches: Since the computation time does not meet the requirement of the real-time application, other real-time scheduling approaches are compared to the results of the optimal solution.
- Rolling horizon: In order to overcome the computation time of exact algorithms, it
 is only used for a few CAVs. Two approaches are used in the literature. The first
 one reschedules the sharing of the common zone, for each new incoming vehicle,
 considering that some are already scheduled. E.g. the one that has already gotten
 the green in the stop/go approach keeps the green. The second one splits the
 storage zone into two zones. The first one is used for scheduling and once the CAV
 is in the second zone, they are considered as they are already scheduled.

Note that despite the simplification provided by dynamic programming, it remains challenging to obtain the optimal solution as the number of considered CAVs increases. This is not an issue of computation time when the optimal solution is computed offline for comparison. However, the required memory space limits the number of the considered CAVs. Hence, the comparison with the exact algorithms can only be used for a short simulation period. This motivates many papers to introduce heuristics to obtain near optimal solutions.

In [36], the authors give an analogy between the scheduling of cooperative intersection and the well-known Travelling Salesman Problem (TSP) [101], where each CAV is considered as a city to be visited. This analogy invites the authors to use an ant colony system [102] to decide which CAV goes first, which one goes second, and so on. The ant colony system was adapted to form groups of CAVs that can pass together. The results of the proposed ant colony system were compared with the ones of exact optimal solution in a real intersection. They show that they are close to the optimal ones. An ant colony system was introduced later in [103], for a decentralized negotiation between CAV (see section 2.4.4). More precisely, each CAV participates in the decision-making process. Genetic algorithms [104] are also used to schedule either a single intersection [105], priority vehicles [106] or a network of intersections [107]. In [107], groups of CAVs were formed to pass through the intersection together.

2.4.3.2/ POLICIES

In this thesis, we refer to a scheduling algorithm as a policy if it is in the form of relatively simple rules that define the priority between each pair of conflicting CAVs (See Definition 2). There are many motivations to use policies rather than classical algorithms. In

[63], a policy is used in order to respect real-time constraints. Rules are also used very early, considering CIM as a multiagent system [56]. Under this modeling approach, the objective of using policy is the definition of simple agent's behaviors able to make the emergence of a global behavior that is close to the optimal [85]. The last motivation is that the system is dynamic. It has been shown through an example in [54] that each new arrival of a CAV in the storage zone can modify the previously found optimal solution. Some common features of optimal solutions can be used to define simple rules that keep a high level of performance. This process aims at avoiding the costly computation overhead of algorithms or of heuristics, knowing that the sum of two optimal solutions is not necessarily an optimal solution [108]. These different motivations led to different policies. To the best of our knowledge, it is possible to group the policies used into four families: First In first Served (FIFS), First Ready Out (FRO), Time To React (TTR) and Distributed Clearing Policy (DCP). Some other policies are discussed also.

Definition 2: Policy

In CIM, a policy is a set of finite priority rules that are sufficient to solve a conflict between two CAVs that need to cross the same potential collision area coming from different storage zone. The conflict is solved by designating which one goes first and which one goes after. This set of rules must be sufficiently welldefined so that it does not result in a circular dependency when scheduling all CAVs. Mathematically, a policy must be able to set a transitive order relation [109] between CAVs.



Figure 2.11: FIFS deadlock examples: A-FIFS deadlock is the result of the reservation of tiles A, B and C, by CAVs 1, 2 and 3 (The number on CAVs are only ID) B-FIFS deadlock is due to communication issues in Virtual Platoon that raises the problem of order inconsistency (The number on vehicles indicates their ranks according to the time they were discovered: 2 waits for 1 who is behind 3)

Here are the most known CIM's policies:

 First In First Served: FIFS is the most popular policy, especially for reservation and virtual platoon protocols. It was introduced in CIM by [56]. In its first version, the "First In" does not mean the first CAV in the storage zone. Instead, the proposed policy means that when the CAV books a given tile, this tile is no more available for the others until the reservation is canceled or fulfilled. From this standpoint, there is a kind of competition of booking tiles between CAVs. As a result, the reservation may not respect the order of CAVs in the storage zone. In such a case, some CAVs need to reserve the tiles again because they are slowed down by the leading CAV that move in the same lane. The protocol was improved to solve the issue in [110]. It states that the priority is proportional to the time spent in the storage zone. In [54], CAVs respect the order of their arrival in the storage zone, except when they can cross the intersection without conflict. In [111, 112, 113], the authors consider the deadlock that results from FIFS. Both former papers solve the circular weight due to the reservation of more than one potential collision areas (see Figure 2.11-A). The last one considers the deadlock in the virtual platoon, when messages are lost (see Figure 2.11-B). The solutions proposed in the three papers are based on the elimination of cycles in the graph [114] that represents the priorities. However, the solutions aren't trivial. Except the above-mentioned works, in the majority of papers, FIFS policy orders the vehicles according to their arrival time in the storage area. If there are no wireless communication issues, this policy is very convenient for both reservation and virtual platoon protocols. Each CAV is informed, when it gets into the storage zone, about the constraints to be considered for the optimization of its trajectory (speed profile).

- · First Ready Out: This policy means that when a pair of vehicles are in conflict, the CAV that is able to cross the intersection first will have the highest priority. The priority depends on the speeds and positions of both conflicting vehicles at the time the priority is calculated. It also depends on when the vehicles in front leave the intersection. Thus, in [115], the CAV has at least a lower priority than the one in front, in the same conflict zone. It is obvious that when the priority of CAVs is computed according to when they are able to cross the intersection, the control problem becomes more complex. Indeed, it is more about a competition to be the first in the exit zone. Simultaneously, they must be able to come to a complete stop if they lose. Thus, to the best of our knowledge, this policy is not fully implemented in CIM in a dynamic way. Nevertheless, we can find its principle partially applied, especially when it comes to conflicts between a small group of CAVs. In [116, 117], vehicles follow an auction principle for reserving tiles. In a competing set of CAVs, the CAV which is able to clear the intersection at the earliest is the one that books the tiles first. In [63], due to the non-linearity of the constraint of the conflicting CAVs, the MPC is distributed so that each vehicle tries to exit with a relaxed constraint. As soon as an exit time can be ordered, the CAVs in the group respect gradually the safety constraint (constraint relaxation). The new group of CAVs competes again with respect to the exit dates of the previous ones and so on.
- *Time To React:* This policy was introduced for distributed MPC, in [61, 62, 64]. From [64], "*TTR is defined as the duration to the furthest point in time* t_x where the
vehicle can still decelerate enough to come to a standstill in front of the intersection". Each CAV computes TTR which defines its priority. The CAV that has the lowest TTR goes first. The optimization function includes the safety constraint to be respected by CAV to do not collide with more priority CAVs. Comparison was performed with a centralized MPC, showing that the decentralized approach is more efficient in terms of computation time and even the deviation to the reference speed [64]. In [61], the authors compare TTR to FIFS and to the priority based on the remained distance to the potential zone of collision. Both last policies were not feasible in terms of safety. However, to the best of my knowledge, the TTR policy was only simulated for a few CAVs (6 at most).

· Distributed Clearing Policies: DCP is introduced for the stop and go protocol. It is based on a set of rules. The rules aim to form a group of CAVs that are allowed to cross the intersection together [54]. The lane that has the oldest CAV is chosen. The group is first formed by CAVs that belong to the lane if they are close together. If the leader CAVs that move on the other lane can cross the intersection in parallel to the formed group, they join the group, and so on. It was mathematically proven that DCP minimizes instantaneously the queue length if the headway time between two conflicting CAVs is at least two times bigger than the headway time between two CAVs coming from the same lane [118, 54, 119]. DCP is an extension of the wellknown vehicle actuated traffic signal [120, 121]. From this standpoint, the green is given to a CAV if this one is able to cross the intersection before a given step of time. Otherwise, the green is given to the oldest CAV in the storage zone. It was later generalized to a network of intersections [122]. Some works attempt to adapt the DCP to virtual the platoon protocol. [42] proposed two steps for forming groups in a real simple intersection of mini-robots. First, there is a small amount of time during which the new incoming robots negotiate together in order to form groups with the closest precedent robot in the same lane. Later, the robots adjust their speed together to avoid collisions. [123] extended the approach by providing a spatial decomposition of the problem. When a CAV arrives, in the first zone, it joins the group called a bubble. Then, in the mid-zone, the bubbles are ranked by a brunch and bound algorithm. Because of the complexity of the problem, the number of bubbles is bounded for obtaining a feasible sequence in a reasonable time. Later in the third zone, CAVs respect the local vehicular control to cross safely the conflict zone. It is obvious that the DCP is difficult to use in a protocol without a complete stop of the vehicles, like reservation and virtual platoon protocols. The main reason is that the group needs to be formed dynamically. The CAV must either delay its exit time or include new obstacles, as it moves forward. Because of the efficiency of DCP for a high traffic flow, the adaptation of DCP to the reservation and virtual platooning protocols remains a hot topic [124, 125, 126, 127].

• Other Policies: The other policies are more complex than the one quoted previously. A first example of the other policies is the Dynamic Priority (DP). DP is a policy where each CAV compute periodically its priority as a function of values that evaluate the correspondence of its state with predefined criteria. The CAV that has the highest priority goes first. This was first introduced by [49], where the priority is the weighted sum of the vehicle readiness and the sum of the priority of vehicles in the same lane. The readiness itself is computed as a weighted sum of the distance covered in the storage zone, the vehicle's velocity and the idling time. A similar approach were used in [23]. The PD is used to solve conflicts between busses, in order to avoid a bus that has a delay to move behind a bus that is in advance. Rather than using a weighted sum, the paper uses a max function. However, DP, if is not properly defined, can lead to permanent changes in the order of vehicles during their movement in the storage zone. In order to avoid unstable priority of busses, [23] adds a filter. Another noticeable policy is the one presented in [48, 128, 129, 130]. The policy is based on game theory [131]. Instead of deciding who goes first, the system delays one conflicting CAV for each period (0.5 sec). This CAV is chosen from the set of conflicting vehicles: those that collide if they do not slow down. The group of conflicting vehicles selects the CAV that delays the most other CAVs (number of delayed CAV). The chosen CAV decelerates at the beginning of the next period. This CAV-by-CAV selection requires a stopping and an accelerating distance. Thus, the author proposes an adapted architecture of the intersection with a slowing down and acceleration zone in the storage area. It should be noted that simulations show that the policy performs well in the case of low and medium traffic flow. When the traffic demand is high, the system tends to perform like traffic lights.

Remark ¹. At this stage of the state of the art, let us draw the reader attention to the following folds:

- The scheduling problem remains a hot topic to achieve an efficient CIM
- there is a kind of opposition between scheduling optimization and trajectory control optimization: On the one hand, the reservation protocol seems to be the most suitable for the optimization of the speed profile (longitudinal control), but it needs the exit dates quite early to allow CAVs optimally planning its trajectory to reach the conflict zone. Only no dynamic priority policies, such as FIFS, can fulfill this requirement. On the other hand, other policies such as DCP provide good performance when the traffic flow is high. However, CAVs are prone to a complete stop before the end of the conflict zone.
- Finally, Some policy/protocol pairs require more lanes or longer storage areas, such as game theory/reservation [129] and DCP/Virtual platoon [123].

2.4.4/ ARCHITECTURE

Architecture refers to the way in which CAVs communicate together to negotiate their right of way and get it. Architecture defines both communication and computational hardware and software required to achieve the negotiations. Currently, there are several communication channels [132] that can be used for CIM, such as 5G (cellular network) [133, 134] and G5 (Dedicated Short-Range communications: DSRC) [135, 136] to quote a few. Telematics Control Units (TCU) [137, 138] are used to carry the CAV connectivity with others (V2X: Vehicle to others). Moreover, embedded high-performance computing system in vehicle is becoming more widespread [139, 140] for vision, high-resolution display and vehicle control. At the beginning of CIM studies, there were two opposing visions of the negotiation architecture, i.e. decentralized and centralized architecture. Later, a mixed architecture was proposed.

Decentralized architecture

The first architecture was decentralized [49, 141, 142, 143], where CAVs are able to selforganize their access to the conflict, without the need for an external agent. This architecture avoids the use of external devices (ITS RoadSide Unit), to manage the intersection. In this architecture, there is only V2V (Vehicle to vehicle) communication. However, the approach is questionable in terms of intersection safety. Because there is no intersection server, it is difficult to use a default deny rule. Each CAV that is not able to communicate with others will react as it is alone. This results in collisions, in the tests of the intersection of mini-robots presented by the authors of [37]. One solution is to oblige the CAV that is alone to slow down near the conflict zone to detect whether there are other CAVs. However, this solution may considerably lower the performance of the intersection.

Centralized architecture

The second architecture is centralized. In this architecture, there is an intersection server that communicates with CAVs to allow them to reach the conflict space safely. This architecture is entirely based on V2I (Vehicle to infrastructure) communication. The task of the intersection is the optimization of the whole intersection, according to the used algorithm or policy.

This kind of architecture suites the use of optimization algorithms. The centralized approach was used for the scheduling optimization, through algorithms [92, 94, 100, 60] and heuristics [36, 104]. From this point of view, the centralized architecture saves communication and calculation overhead compared to the decentralized one. The centralized architecture avoids that each CAV communicates its status to all surrounding CAVs and schedules the other's access time. The server performs these tasks for the whole. A centralized architecture is also an opportunity to optimize together both schedule and trajectory (See Remark given in Section 2.4.3.2). However, except treating both problems

into two separate steps as presented in [60], several papers witnessed that solving both problems together is currently unfeasible because of either the computation time [61, 62] or convergence [63].

The centralized architecture alone is not sufficient to obtain a collision free intersection. For instance, in [48], after the computation of the schedule, the intersection manager sends acceleration to only CAVs that need to decelerate or accelerate. CAVs that maintain their speed are not contacted by the server. Simply speaking, the CAV that does not receive a message from the server because of a communication issue, will just continue its movement. However, the centralized architecture contributes to the safety enhancement through the default deny rule. The CAVs are resorted to a complete stop before the end of the storage zone if they cannot contact the server as discussed in [40, 41, 42, 111].

Mixed Architecture

The last possible architecture is a mixed architecture, where both V2I and V2V are used. This architecture was mentioned in [75, 76]. It can be an alternative to ensure the safety of the intersection while allowing negotiation of the rights of way between vehicles. This approach will allow the implementation of distributed intersection management algorithms, where CAVs take full part in the decision-making process. It is also suitable for policies for adapting the priorities and trajectory control to the traffic dynamic. For instance, the RSU broadcasts the presence list to all surrounding CAVs and CAVs negotiate their priorities together. When they exit, they remove themselves from the presence list. The presence list can be built using geo-networking protocols of DSRC [144].

2.4.5/ DISCUSSION

There is a consensus that CAV allows a new approach to managing the intersections that have a great potential to smooth the traffic. There are two opposing visions to achieve the objective. The first vision consists of planning all the vehicles in an optimal way. This vision comes up against the computation complexity and the traffic dynamic, which questions the obtained solutions. The other vision consists in making the CAV smarter so that the interaction of individual behaviors results in a collective behavior able to enhance the traffic conditions. This vision in turn faces problems of intersection safety, i.e. without collisions and deadlock. These two opposing visions partly explain the variety of negotiation protocols, scheduling approaches and the two dominant architecture families.

Contributions must be consistent in the entire decision-making process, that is how CAVs share the conflict space (protocol), how they get solutions (scheduling) and finally through which material and software infrastructure (architecture). This consistency requirement explains partially the lack of real tests in the literature, since more than three decades (see table 2.1). This requirement raises a deeper theoretical problem, namely how to

optimize simultaneously both speed profile (trajectory) and sequence (schedule). The two problems are of different natures and even of distinct research areas. The first problem belongs to the theory of control of the continuous system (acceleration, speed, position), whereas the second belongs to the wide field of combinatorial optimization, which deals with discrete decisions. In addition to this theoretical complexity, the dynamics of traffic question the solutions taken, however perfect they were at the previous time.

Furthermore, although several papers show significant performance improvements through simulation, the gains are difficult to quantify from the traffic engineering perspective. In particular, there is no consensus on the time between successive passages of two conflicting vehicles. It is not an issue of estimating the exact values, that depends on the used material. However, at least it is mandatory to reach a consensus in a mathematical formulation that makes this time compared to the time between two successive passages of CAVs that move on the same lane. It is obvious that if the conflicting time is lower, the succession of conflicting CAVs improves the throughput. In this case, the research work in CIM must be directed towards achieving this low time as safe as possible, without collisions and deadlock. Several real tests need to be achieved to reach the idealistic conflict time. Otherwise, it is necessary to work fully on the scheduling of CAVs, by searching for policies that address computation time and traffic dynamics.

The other difficulty of linking the results to traffic engineering is the choice of intersection geometry. One of the expected advantages of CAVs is that they are able to save space dedicated to vehicles in urban areas [145]. Thus, it is difficult to quantify the advantage of control solutions that require a significant investment in additional urban space reserved for vehicles. Instead, the performance gains should be in the space that the autonomous vehicle gives to other activities (terraces for shops, home gardens...) and other mobility systems (wider pavements for pedestrians, dedicated lanes for soft transportation modes...). Thus, from this point of view, it is important to study capacity gains on simple intersections, such as the one presented in Figure 2.1, rather than considering a huge intersection with several lanes. Simple intersections raise the difficulty of considering several movements in the same lane, such as turning left movement that often blocks the other vehicles going straight. Such problems are avoided when studying a huge infrastructure.

2.5/ CONCLUSION

This chapter has reviewed the state of the art of automated control of elementary urban intersections. Currently, the most adaptive intersection control is based on traffic lights, which have some inherent limitations, such as the limited number of phases, due to the time lost, and the inadequacy of the traffic light with the vehicles actually present, be-

2.5. CONCLUSION

cause of the lack of communication with vehicles. The perspective of autonomous and connected vehicles seems to bring new features that have a strong potential to improve traffic conditions. Vehicles are able to adapt together phases according to their state and control speeds to save time.

The literature review shows that these two advantages are not easy to be harnessed. Theoretically, they raise new and complex problems. Depending on the point of view, the problem can be associated with the control of hybrid systems (continuous-discrete) [49, 146], the collective intelligence [147] of multiagent systems [56], the control of constrained non-linear systems [63], etc. Let us add to these theoretically complex problems, the traffic dynamics that raise a new problem at each time-step, as well as the cyber-physical and real-time constraints. In order to overcome this complexity, in parallel to purely mathematical approaches seeking for optimal solutions, several approaches are based on simple protocols and policies. Our contribution aims to find simple policies that harness both velocity profiles and sequence. Our work will start from a virtual platoon based on FCFS policy [146, 78].

3

EXTENDED VIRTUAL PLATOON

3.1/ INTRODUCTION

Autonomous vehicles that are able to communicate together allowed the emergence of a new way of controlling intersections. Recently an active research community stakes on cooperative intersection management. Vehicles and intersections are able to communicate together in order to improve traffic conditions at intersections. Many simulations have shown that the cooperative intersection management outperforms traffic lights because (i) there are no limited stages as within traffic light and (ii) authorized movements are adapted to the current situation.

This chapter focuses on virtual platoon formation at intersections. More precisely, a virtual platoon is formed by vehicles coming from different lanes so that each vehicle considers the precedent CAV moving in another lane as it is moving before it in the same lane. When traffic flow is low this allows avoiding useless stops and thus the speed adjustment saves energy and increases the average speed. However, experiments show that as the traffic flow grows, vehicles need to stop and cooperative cruise control is no more efficient. The main reason for this drawback is that the studied concepts are limited to First In First Served.

This chapter aims to extend the existing policy, i.e. First In First Served. Only scheduling rules are used. To this end, many challenges need to be addressed and parameters to be set. First, a simulator of the virtual platoon was developed with a visual interface that displays vehicles in the intersection. This aims to (i) avoid biased results due to hidden collisions and to (ii) thoroughly set the protocol parameters. Second, the evaluation process to compare different policies is defined. The main objective of this step is to fit as much as possible the known approaches for the capacity evaluation in traffic engineering. Finally, the extension of the virtual is done gradually, with respect to the encountered issues.

This chapter is organized as follows. First, it introduces the virtual platoon protocol and

the choice made, in terms of cooperative cruise control and communication architecture. Second, it presents the most important features of the simulator. Later, it details the newly introduced policy. Finally, it discusses simulation results before concluding.

3.2/ VIRTUAL PLATOON PARAMETERS

Virtual platoon, as introduced in [42], extends the platoon of CAVs that move in the same lane [148, 149, 150], by introducing the conflicting CAVs to the platoon. This is possible through wireless communication, see figure 3.1. In this section, we will detail through the problem formulation the four parameters that need to be stated to design the virtual platoon. These parameters answer the following questions:

- The synchronization point: From where the CAV needs to keep a safe distance with the conflicting leader, in the storage zone?
- Advanced cruise control: How the CAV keeps a safe distance from the precedent ones?
- Cooperative cruise control: How the CAV consider several obstacles, with some are virtual and others are real?
- Architecture: How can CAVs modify their schedule?

The four parameters are fixed gradually, by keeping our objective to allow CAVS a wider opportunity of scheduling negotiation, before getting to the conflict zone.



Figure 3.1: Virtual platoon formed by two CAVs that does not move in the same storage zone: CAV₂ adjusts in R_4 its speed according to CAV₁ that moves in R_1

3.2.1/ SYNCHRONIZATION POINT

let us give a general mathematical formulation of the problem that two conflicting CAVs presented in Figure 3.1 have to solve. Without loss of generality, assume that both CAVs_{*i*} ($i \in \{1,2\}$) have to cross a potential collision area. The dynamic of both CAVs can be expressed as follows:

$$\dot{X}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} X(t) + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} U(t).$$
(3.1)

Such that in (3.1), $X^{T}(t) = [p_{1}(t), v_{1}(t), p_{2}(t), v_{2}(t)]$ and $U^{T}(t) = [u_{1}(t), u_{1}(t)]$, with $p_{i}(t)$, $v_{i}(t)$ and $u_{i}(t)$ being the traveled distance from the beginning of the storage zone, the speed and the acceleration of the CAV_i, respectively. Speed and acceleration constraints need to be added as follows:

$$\underline{u_i} \le u_i(t) \le \overline{u_i},\tag{3.2}$$

$$0 \le v_i(t) \le \overline{v_i},\tag{3.3}$$

with $\overline{u_i}$ and $\underline{u_i}$ designate the maximum and the minimum acceleration (deceleration), respectively. The maximum speed is $\overline{v_i}$. Equation (3.3) avoids both CAVs to move back but allows them to come to a complete stop. $\overline{u_i}$, $\underline{u_i}$ and $\overline{v_i}$ are defined, in equations (3.2 and 3.3), not only according to physical limitation of CAVs but also by taking into account criteria such as the safety and convenience of passengers and good on the autonomous vehicle and maintenance costs.

With equations (3.7), (3.2) and (3.3) in mind, it remains the safety constraint to avoid collision of both CAVs when they cross the intersection. Recall that, as discussed in section 2.4.5, there is no consensus in the way the safety constraint is formulated. In this thesis, we consider the most general mathematical formulation of the constraint as discussed in [151, 152]. Let $d_{i,inter}$ and $d_{i,exit}$ designate the position of the CAV_i to get into¹ and out the potential collision area, respectively. The conflict constraint is as follows:

$$\sum_{i=1}^{2} f_{i,enter}(t) - f_{i,exit}(t) - 1 \le 0,$$
(3.4)

with $k \in \{enter, exit\}, f_{i,enter}$ and $f_{i,exit}$ are defined as follows:

$$f_{i,k}(t) = \begin{cases} 0 & \text{if } p_i(t) < d_{i,k} \\ 1 & \text{if } p_i(t) \ge d_{i,k} \end{cases}$$
(3.5)

 $^{^{1}}d_{i,enter}$ requires considering the safety distance to avoid the collision.

Equations (3.4) and (3.14) show that the system is hybrid. However, even by stating which CAV goes first, other parameters need to be set before using only continuous state variables ($p_i(t)$ and $v_i(t)$) instead of $f_{i,k}(t)$. There are two ways to simplify the constraints of the stated problem:

- Reservation: Setting the times when CAVs are entering and exiting the potential zone of collision. With these times, it is possible to compute the optimal trajectory under a reservation protocol as formulated in [59]². However, as discussed in section 2.4.2.2, the safe time between two conflicting is unknown and collisions may happen if the time is underestimated [39]. For instance, this time is set manually, according to the traffic flows, through the variable *m* in [60].
- Virtual platoon: Considering that both CAVs are virtually moving in the same lane.

In the second case, a slack variable must be introduced. If we consider that the index *i* gives the rank of the CAV_{*i*}, so that CAV_{*i*} precedes CAV_{*i*+1}, the constraint (3.4) is simply brought to the following one:

$$p_i(t) - p_{i+1}(t) \ge d_{i,exit} - d_{i+1,enter}(v_{i+1}(t)) + \delta_{i,j}(p_{i+1}(t)),$$
(3.6)

with $\delta_{i,j}(p_{i+1}(t))$ is a slack negative variable that allows the initial state to respect the constraint. When the CAV_{i+1} is at the beginning of the storage zone, this variable equals $d_{i+1,enter}(v_{i+1}(t)) - d_{i,exit}$. This allows both CAVs to have the same relative position at the beginning of the storage zone. As CAV_{i+1} is close to the conflict zone $\delta_{i,j}(p_{i+1}(t))$ tends to equal zero. [78] introduces such a slack variable to control the cruise of the follower CAV. However, the choice of the formulation of this slack variable isn't detailed.

Note that the curve of the slack variable modifies the way CAVs occupy the storage zone. The slack variable acts as a cursor on where the safety distance is respected. If it tends towards zero in the middle of the storage zone, the follower CAV prepares the safety distance before the middle of the storage zone. As a result, the CAV decelerates before and the other CAVs behind are resorted to stopping earlier (see Figure 3.2). Let's call the "**point of synchronization**", the position from which CAV₂ respects the safe distance with CAV₁. The choice of the point of synchronization is not neutral. There are two contradicting behaviors that the designer needs to face:

• Smooth CAVs behaviors at the conflict zone: The earlier is the point of synchronization, the faster CAVs are in the conflict zone, but, the shorter is the distance used for scheduling in the conflict zone.

²The virtual platoon presented in this chapter will be extended in the next chapter to a safe reservation protocol



Figure 3.2: Virtual platoon test of three robots, where the point of synchronization is at the middle of the storage zone (White paper): The two follower robots move slowly so far to keep a safe distance

• Efficient use of the storage zone: The further is the point of synchronization, the better is the use of the storage zone for scheduling and for receiving CAVs from the upstream intersection, but, the CAVs move slower in the conflict zone.

In other words, moving the synchronization point backward reduces the scheduling opportunities and requires additional distance to clear the upstream intersection. A discussion about the opportunistic position of the synchronization point was given in [75, 76]. The position of this point has to be thoroughly calculated according to the performance objectives of the intersection. This work will be conducted in the next chapter. This chapter intends to fulfill our primary objective: smoothing the intersection traffic by keeping the intersection geometry unchanged, i.e., without increasing the length of the storage area. The second challenge that this chapter intends to overcome is to increase the scheduling opportunity. To meet both challenges, it is obvious that the synchronization point must be as close as possible to the conflict zone. In order to do not change the current geometry of the intersection, the constraint of respecting the safety distance in a virtual platoon is considered at the stop line, namely at the end of the storage zone.

The position choice of the synchronization allows also to fully harness the intersection scheduling opportunities. As one can observe in Figure 3.2, because of the point of synchronization, both robots that move in the same lane are separated too early. This big distance between them makes the platoon with the leader one³ inefficient. With a longer distance before the point of synchronization, CAVs will have a larger time to form real groups that will cross the intersection together. So, there is a bigger opportunity to modify the initial order of arrival (FIFS).

³Forming real platoon in the same conflict zone is one of the priority rule of DCP (see section 2.4.3)

3.2.2/ ADAPTIVE CRUISE CONTROL

With a continuous objective function, the equation (3.1) under the constraints (3.2), (3.3) and (3.14) can easily be solved, using Pontryagin [153] maximum principle to derive the optimal control of the second CAV trajectory [154, 155, 156]. However, the system dynamic equation (3.7) ignores the time delay issue. In the majority of the well-known car following models, such as models of GHR [157], of Gipps [158], of Wiedemann 99[159, 160] and of Intelligent Driver Model (IDM) [161], a time delay is added to consider the reaction time of the human drivers. As for human drivers, a time delay needs to be considered for CAVs. This time delay is first caused by the sensor's frame rates, cycle time and accuracy [162]. Second, the data processing algorithms require time [163, 164, 165]. Third, the normal communication issues are also a non-negligible source of delay [166, 167, 168]. Finally, the mechanical components of the vehicles to perform the control take time. As a result, the dynamic of the CAV given by (3.1), must formally be written as follows:

$$\dot{X}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} X(t) + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} U(t-d),$$
(3.7)

with d > 0 is the time delay.

With d > 0, the solution of the optimization problem is subject to approximations through numerical solutions [169], discrete-time optimal control techniques [170], with a sampling time or nominal control based on analytical approximations [171]. In this chapter we considered the last option, using RT-CVC [77]. The objective function of the RT-CVC is to be instantaneously the closest to the safety distance from the leader. This allows to fully use the storage zone for receiving CAVs and for scheduling them. Formally, the problem solved by RT-CVC is written as follows:

$$\begin{array}{ll}
\min_{u_{i+1}(t)} & p_i(t) - p_{i+1}(t) \\
\text{s.t.} & p_i(t) - \underbrace{\frac{v_i(t)^2}{2\underline{u}_i}}_{h_i(t)} - \underbrace{p_{i+1}(t) - v_{i+1}(t)\tau - u_{i+1}\frac{\tau^2}{2}}_{p_{i+1}(t+\tau)} + \underbrace{\frac{(v_{i+1}(t) + u_{i+1}\tau)^2}{2\underline{u}_{i+1}}}_{h_{i+1}(t+\tau)} \ge S, \quad (3.8)$$

$$u_{i+1}(t) \ge \underline{u_{i+1}}$$

with $h_i(t)$ and S > l designate the distance that the CAV needs to come to a complete stop and the constant relative distance that the second CAV needs to maintain for safety. Figure 3.3 illustrates the principle of RT-CVC. The solution to the problem 3.8 is given [172] and tested for following real cars during the thesis (see Figure 3.4) and in a real



Figure 3.3: RT-CVC cruise control principle for safely considering the time delay τ

intersection in ITS world congress [43, 173]. It was also compared to the enhanced IDM [74] in [75, 77]. It shows that RT-CVC is safer under the following conditions:





- <u>ui</u> ≤ <u>ui</u>: This condition aims to consider the difference of braking capability between the leader and the follower.
- $\tau = 1.2 \cdot d$: This allows to consider errors to the time discretization of the computation process.

In addition to the safety advantage, RT-CVC allows to explore the storage zone to the maximum, since the CAV moves at the limit of the safety constraint.

3.2.3/ COOPERATIVE CRUISE CONTROL

The cooperative cruise control for virtual platoon splits the intersection space into three sets of zones: entrance, conflict and exit zones. For safety reasons, each CAV_i at a rank $n_i(t)$ adjusts its speed in storage and conflict zones according to:

- Real leader (rl): the precedent CAV that moves on the same storage zone,
- Virtual leaders (vl): the set of conflicting CAVs vl_j , at a rank $n_j(t) < n_i(t)$,



Figure 3.5: Virtual platoon: (a) Intersection, (b) virtual platoon, (C) all considered obstacles

- Next leader (nl): the precedent conflicting CAV that will join the same exit zone,
- Stop line (*sl*): the boundary between the storage zone and the conflict zone. It can be any synchronization point chosen previously.

Figure 3.5 gives an example of the obstacles considered by CAV₄. In this figure the index of CAVs correspond to their rank. One can observe that rl is CAV₂ ($rl = \{2\}$), CAV₁ and CAV₃ form the set of vl ($vl = \{1, 3\}$) and nl consists of CAV₀ ($nl = \{0\}$). sl is the point of synchronization. It is the virtual obstacle that prevent collision in the case where CAV_4 is not able to adjust its speed according to CAV₀, CAV₁ and CAV₃.

We consider five signed accelerations $u_i^{rl}(t)$, $u_i^{vl}(t)$, $u_i^{sl}(t)$, $u_i^{nl}(t)$ and $u_i^d(t)$ that are computed instantaneously according to the speeds of both CAV_i (CAV₄ in figure 3.5) and obstacles and their relative distance s_i^* . Table 3.1 gives the considered parameters. $u_i^{nl}(t)$ is computed for the next leader that has already left the intersection (CAV₀ in figure 3.5). $u_i^d(t)$ is used to respect the maximal desired speed of CAV_i ($\overline{v_i}$). Since there are several virtual leaders, then $u_i^{vl}(t)$ is computed as follows:

$$u_i^{vl}(t) = \min_{vl_i \in vl} \left(u_i^j(t) \right)$$
(3.9)

with

$$u_i^j(t) = \begin{cases} u_i^{sl}(t) & \text{if } s_i^j < 0\\ u_{vl_i}^j(t) & \text{otherwise} \end{cases}$$
(3.10)

 $u_{yl}^{j}(t)$ is computed according to the relative distance and the speed of CAV_j. The instan-

Variables	Meaning
*	the obstacle
\underline{u}_{i}^{*}	the minimal desired acceleration of CAV_i for a given obstacle $*$
$\underline{u_i}$	the absolute minimal desired deceleration of CAV _i
$\underline{u_*}$	the minimal expected deceleration of the obstacle
$\overline{u_i}$	the maximal acceleration of CAV _i
$v_i(t)$	the current speed of CAV _i
$\overline{v_i}$	the maximal desired speed of CAV _i
S_i^*	the relative distance between CAV_i and the obstacle $*$ minus a safety
·	distance
τ	the time delay of the system multiplied by 1.2

Table 3.1: Parameters of the acceleration function

taneous acceleration that results from all conflicts is obtained by the following equality:

$$u_{i}^{c}(t) = \max\left(u_{i}^{sl}, \min\left(u_{i}^{vl}(t), u_{i}^{nl}(t)\right)\right)$$
(3.11)

because CAV_i must stop before the conflict zone (synchronization point) if it is not able to follow its virtual leaders and its next leader. Finally, the CAV must avoid collision with its real leader. Hence, we have:

$$u_{i}(t) = \min(u_{i}^{rl}, u_{i}^{c}, u_{i}^{d}(t), \overline{u_{i}}).$$
(3.12)

We draw the reader attention to the fact that equations (3.9), (3.10), (3.11) and (3.12) are open to any adaptive cruise control originaly used by the CAV. The control strategy can be linear to the errors [175], based on sliding mode techniques [176], on the well-known IDM [74] or even on Deep Reinfocement Learning techniques [177]. In this thesis, we use RT-CVC as stated in the previous subsection. The accelerations are computed as follows:

$$u_{i}^{*}(t) = \frac{\underline{u_{i}^{*}} \cdot \tau - 2 \cdot v_{i}(t) - 2 \cdot \underline{u_{i}^{*}} \cdot \sqrt{X_{i}^{*}(t)}}{2 \cdot \tau}$$
(3.13)

with in equation (3.13), we have:

$$X_{i}^{*}(t) = \begin{cases} \frac{\underline{u}_{i} \cdot \tau^{2} + 4 \cdot \tau(v_{i}(t) - 2 \cdot \overline{v_{i}}) + 4 \cdot \frac{\overline{v_{i}^{2}}}{\underline{u}_{i}}}{4 \cdot \underline{u}_{i}} & * = d \\ \\ \frac{\underline{u}_{i}^{sl} \cdot \tau^{2} + 4 \cdot v_{i}(t) \cdot \tau - 8 \cdot s_{i}^{sl}(t)}{4 \cdot \underline{u}_{i}^{sl}} & * = sl \\ \\ \frac{\underline{u}_{i}^{sl} \cdot \underline{u}_{*} \cdot \tau^{2} + 4 (\underline{u}_{*} \cdot v_{i}(t) \cdot \tau + v_{*}(t)^{2}) - 8 \cdot \underline{u}_{*} \cdot s_{i}^{*}(t)}{4 \cdot \underline{u}_{*}^{s} \cdot \underline{u}_{*}} & * \neq sl, d \end{cases}$$
(3.14)

In order to illustrate RT-CVC, simulations are performed between two conflicting CAVs,



Figure 3.6: Simulated scenario of virtual platoon with 2 conflicting CAVs: CAV2 follows CAV1

i.e., CAV₁ and CAV₂. Parameters of the simulations are given in table 3.2 and Figure 3.6. The simulations aim to show that \underline{u}_{i}^{sl} acts as a cursor for the synchronization point. This is, somehow, similar to $\delta_{i,j}$ ($p_{i+1}(t)$). The simulation results are given in figure 3.7. Figure 3.7-A gives the position of CAV₁ and CAV₂ under four different values of \underline{u}_{i}^{sl} . The circles show where CAV2 begins to follow CAV₁. One can note that \underline{u}_{i}^{sl} modifies the position where CAV₂ begins to follow CAV₁. The reason is found in Figure 3.7-B that gives the acceleration values for the four scenarios of \underline{u}_{2}^{sl} . As the \underline{u}_{i}^{sl} is closer to zero, as CAV₂ begins the deceleration sconer to be able to come to a complete stop before the stop line with respect to \underline{u}_{2}^{sl} . So, in the fourth scenarios, CAV₂ respects \underline{u}_{2}^{sl} . Because we intend to benefit from the total storage zone for the scheduling, in the reminder of the whole thesis we set:

$$\underline{u}_{i}^{sl} = \underline{u}_{i}^{vl} = \underline{u}_{i}^{nl} = \underline{u}_{i}^{rl} = \underline{u}_{i}$$
(3.15)

Another important point to highlight, is the intersection safety. This simulation shows that the formed virtual platoon is a collision free. First, let us focus on figure 3.7-A and B. CAV₁ exits the potential zone of collision, at t = 6.6s whereas CAV₂ enters later (t = 7.7s and $v_2(7.7) = 15.0m/s$ for $\underline{u}_2^{sl} = -1.5m/s^2$, t = 7.6s and $v_2(7.6) = 14.5m/s$ for $\underline{u}_2^{sl} = -2m/s^2$, t = 7.5sand $v_2(7.5) = 13.5m/s$ for $\underline{u}_2^{sl} = -3m/s^2$ and t = 7.0s and $v_2(7.0) = 14.9m/s$ for $\underline{u}_2^{sl} = -4m/s^2$). Figures 3.7-C and D give the results of the simulation of CAV₁ launching a deceleration of $-6m/s^2$ from t = 5.2, in order to come to a complete stop at the position 96.72m. The CAV₁ remains in the potential zone of collision. Figures 3.7-C shows that the four CAVs were able to come to a complete stop at a position of 87.74m. In other words, they don't enter the potential zone of collision. Figures 3.7-D shows that the four CAVs respect the

Variables	values
CAV length	4 <i>m</i>
CAV width	2.2m but the width of the potential zone of collision is considered $3m$
$v_1(t)$	15 <i>m</i> / <i>s</i>
$v_1(t)$	computed
$\overline{v_i}$	$15m/s$ for $i \in \{1, 2\}$
$d_{1,enter}$	92 <i>m</i>
$d_{1,exit}$	99 <i>m</i>
$d_{2,enter}$	95 <i>m</i>
$d_{2,exit}$	104 <i>m</i>
<u><i>u</i></u> ₁	$-6m/s^2$ is the value that CAV ₂ considers for computing $u_2^1(t)$
$\underline{u_2^{vl}}$	$-4m/s^2$
$\underline{u_2^{sl}}$	$-1.5m/s^2$ for pos_1 and accel_1, $-2m/s^2$ for pos_2 and accel_2, $-3m/s^2$
	for pos_3 and accel_3 and $-4m/s^2$ for pos_4 and accel_4
$s_2^{sl}(t)$	80 <i>m</i>
τ	500 <i>ms</i>

Table 3.2: Simulation data of figures 3.7-A and B

 \underline{u}_i . However, the complete stop happens in the conflict zone. This point is important to consider while scheduling CAVs.

3.2.4/ ARCHITECTURE

 u_{rl} can be computed by using the frontal sensor of the CAV whereas for computing a_c , the CAV relies on data coming from wireless communication. According to the discussion given in section 2.4.4, CAVs need a consistent list of presence. To this end, two strategies are used together in this thesis. They are based on communication with the intersection manager that can be an ITS Roadside Unit (RSU). The first one is the default deny. A CAV is resorted to coming to a complete stop before the stop line if it is unable to communicate with the RSU. The second strategy is based on an Ordered Presence List (OPL) that is continuously updated and broadcasted by the RSU.

OPL is a list where CAVs are first ordered according to the FIFS policy. A new CAV in the conflict zone sends to RSU its ID, position, speed and next direction. The RSU adds it at the end of the OPL. When the new CAV receives the OPL, it analyses all precedent CAVs to determine the set of conflicting CAVs that are *vl* and *nl*. It can then start the negotiation with them to modify its rank in the list, while modifying *vl* accordingly. During the whole trajectory of the CAV, the CAV permanently sends its new position and speed to RSU. When the CAV exit the exit zone (a sufficient distance to allow the followers to stop safely $\overline{v} \cdot \tau - \frac{\overline{v}^2}{2\underline{u}}$ with $\overline{v} = \max_i \overline{v_i}$ and $\underline{u} = \min_i \underline{u_i}$), it cuts the communication with the RSU.

Two negotiation architectures are considered:



Figure 3.7: Results of the simulated scenario of virtual platoon with 2 conflicting CAVs: A-position curves and B-acceleration curves

- Mixed architecture: The CAV directly negotiates with the other CAVs its new rank, without modifying OPL sent by the RSU. The new negotiation results on new sets of *vl* for all concerned CAVs.
- Centralized architecture: According to given rules, the intersection manager modifies the rank of OPL. In such a case, once CAVs receive the new OPL they rebuild their sets of *vl*.

The mixed architecture is used later. The CAVs negotiate together their rank and the rank is sent to the intersection manager, which will update the OPL. The new version of the OPL is considered as an acknowledgment of the results of the negotiation. So, CAVs update their access list, when they receive the new version of OPL from the intersection manager.

3.2.5/ DISCUSSION

In this section, we went deeply into virtual platoon parameters. According to our objective to keep the chance that CAVs negotiate their rank during their movement into the storage zone, we state the following parameters:

• The synchronization point: The ego-CAV needs to keep a safe space with the conflicting CAV only before the latter leaves the conflict zone.

- Advanced cruise control: The ego-CAV moves behind its real leader, being the nearest to the safe distance. This allows to stay in the platoon with its real leader for forming a group together.
- Cooperative cruise control: Four obstacles are considered for the safety of the intersection, real leader, virtual leader, next leader and the stop line. The stop line is used when the CAV isn't able to follow the conflicting leader.
- Mixed architecture: There is an intersection manager that broadcast frequently an updated OPL, where each CAV can find its rank. Accordingly, CAVs negotiate together their new rank and send it together to the intersection manager that updates the OPL.

In addition to these four parameters, we recall that, in the proposed approach, we integrated the time delay in the used cruise control as well as the difference of the braking capability between CAVs, for any emergency reasons, by choosing the RT-CVC algorithm for the cruise control.

3.3/ SIMULATOR

Now that all the parameters of the virtual platoon are fixed, it is possible to run simulations. The simulation aims to test the different policies and to evaluate them. A graphical interface was designed to visualize collisions if any. In the reminder, we will describe the following points:

- Simulated environment
- Main classes and functions
- Chosen metrics

This description allows the reader to better understand the scope of the results presented in the remainder of the thesis.

3.3.1/ SIMULATED ENVIRONMENT

The simulator is programmed using Java language through Eclipse. The visualization is based on JavaFX, with a frame rate of 50 frames per second, for showing a 2D top view of the intersection. The simulated intersection is a usual simple four-branch intersection presented in figure 3.8, where each 5 pixels represent 1m. As a result, the length of the storage zone is 80m and the radium for turning left and right are respectively 15m and



Figure 3.8: Simulated intersection geometry in pixels

12m. The length of each side of the conflict zone square is, 27m. The width of each lane is 3m.

The simulation considers homogeneous traffic that consists of only CAVs. All CAVs have the same characteristics. Their length is 4.4*m* and width is 1.8*m*. They all use RT-CVC for the cooperative cruise control. Their control parameters are $\underline{u}_i = -4$, $\underline{u} = -4m/s^2$, $\underline{u}_* = -6m/s^2$, $\tau = 0.5s$ and $\overline{v} = 13.88m/s$, which is 50km/h. These parameters allows a maximum average flow of 0.5 CAVs per second. The chosen parameters aim to be as close as possible to the ordinary vehicles in urban area. So, each CAV is a passenger car unit (pcu) [178]. This allows the use of the passenger car equivalent assumptions to consider other kinds of vehicles.

The visualization of CAVs uses a picture of a car that can have four colors: green, orange, red and yellow. They respectively mean that $vl = \emptyset$, $vl \neq \emptyset$, $u_i < \underline{u}$ and if $u_i < 0$ in the conflict zone. Near each CAV, its current speed is displayed. All these data allow visualizing dangerous cases while testing policies. Two real-time curves are integrated into the simulation display. The first one gives the instantaneous average speed of all present CAVs. The second one gives the number of CAVs that exit the intersection, from the beginning of the simulation. The instantaneous average speed is computed as follows :

$$s(t) = \frac{\sum_{i}^{N} v_i}{N},\tag{3.16}$$

with N is the number of CAVs in storage and conflict zones of the intersection.

Figure 3.9 gives the displayed interface for the user when the simulation is running.



Figure 3.9: Screenshot of the simulator

3.3.2/ CLASSES AND FUNCTIONS

The simulated system is a multi-agent system, consisting of car agents, a driving automation agent and the intersection manager agent. Our code associates to each agent a class. Each agent is presented as follows:

Car agent

Car agent has its own ID, origin and destination. It computes the itinerary of the ego-CAV in the intersection, the absolute and relative coordinates and its speed. This agent updates the set of vl, nl, rl at each new frame, that is 50 fps. Accordingly, it asks the driving automation agent to compute the acceleration value. This agent computes also the parameters needed for negotiating its rank in OPL, such as the time spent, the expected exit time and so on. This is done through communication with the conflicting cars. The car agent keeps also contact with the RSU (intersection manager) either to request a right of way, to be added in OPL, or to update the OPL for changing its rank. The OPL is updated by sending the new position and speed with the new rank if modified.

Car agents are generated by the main program. The generation function respects a flow rate of each lane given in pcu/s. The car generation follows Poisson's distribution according to the flow rate [179]. This leads to a Markov model of the headway time [180]. The origin-destination is computed according to the probabilities given by the user, to turn right, go straight and turn left.

Driving automation

This agent is a set of functions of the adaptive cruise control and the desired speed. It was separated from the car agent, in order to consider several kinds of driver-agent through car-following models. Initially, several tests were performed using IDM and PID controller. In this document, only RT-CVC is used for simulations. This class considers the different kinds of obstacles *rl*, *vl*, *nl*, *sl*. For each CAV in *vl* it computes the acceleration according



Figure 3.10: Fundamental diagram of the traffic (left-side) and the used flow-speed diagram (right-side)

to its relative position and speed. With all received data from the car agent, this driving automation agent computes the "best" acceleration.

Intersection manager

The intersection manager receives messages from cars and builds the OPL. It behaves like a geo networking filter to accept or refuse messages from cars. More precisely, car position and direction in the messages will either make the intersection manager accept or refuse the car. The last option will drop the car from the OPL. By default, the OPL is built according to FIFS.

In addition to the virtual platoon, the architecture of the simulator allows to simulate the stop and go protocol. To this end, only CAVs that are not in conflict is added to OPL according to a given set of rules. Remember that we used a default deny policy and *sl* as the point of synchronization. Hence, a CAV that is not in the OPL just comes to a complete stop the furthest in the storage zone. If it is the leader in the lane, it will be immobilized near the stop line, sending request messages to the intersection manager to get an OPL where it is included (Green).

3.3.3/ METRICS

There are several ways to evaluate the efficiency of a given policy and to compare it. In this thesis, we chose a flow-speed diagram. There are three reasons for this choice:

 The virtual platoon makes the CAVs that are on different roads act as they are moving on the same road. One of the famous ways for modeling the road behavior is the use of the fundamental diagram [81], where congested and smooth traffic can be clearly identified. Moreover, the capacity of the road, that is the maximum average traffic flow, can be computed.

Simulations	1	$\frac{2}{0.18 \mu nc/s}$	3	$4 \\ 0.25 unc/s$	5 0.28µnc/s	6
Flow sharing	0.1 <i>upc</i> /s	0.18 <i>upc</i> /s	0.2 <i>upc</i> /s	0.25 <i>upc</i> /s	0.28 <i>upc</i> /s	0.5 <i>upc</i> /s
intersection	0.4 <i>upc</i> /s	0.72 <i>upc</i> /s	0.8 <i>upc</i> /s	1 <i>upc</i> /s	1.12 <i>upc</i> /s	1.2 <i>upc</i> /s

Table 3.3: Simulated flow rates

- Many simulations in the literature show an improvement in terms of KPI (Key Performance Indicator) for some traffic flows and given intersection geometry. However, it is difficult to link the KPI together, from the traffic engineering standpoint.
- Using several simulation points to address the flow-speed diagram avoids KPI measures with a few simulations (sequence of arrivals of CAVs, speed, geometry...). This small number of simulations can be in favor of a given policy.

The fundamental traffic diagram expresses the traffic flow q (pcu/s) according to the traffic density k (pcu/m) for a given road [181]. Two states of the traffic can be clearly identified from the diagram, as shown in figure 3.10. This is due to the speed that is inversely proportional to the density. The traffic theory puts a relation between traffic flow, traffic density and average speed *S*, as follows:

$$q = \frac{S}{k} \tag{3.17}$$

The fundamental diagram is used for many purposes in traffic engineering. First, it allows the estimation of the road capacity and the travel times through the equation (3.17). It is also used for the traffic forecasting, mainly through the speed-flow diagram. In this thesis, we use the flow-speed diagram instead of the speed-flow diagram and flow density diagram. This permits an easy interpretation and comparison of curves. We can easily observe the capacity of the road as well as the speed improvement. It shows also when the traffic is congested.

The depicted diagram is computed according to the result of simulations. Simulation points are presented, as well as the coefficient of determination R^2 of the fitting curve. Each point represents 60s of simulations. The points show the different observed results. A point can be the result of a sequence of CAVs in favor or against the policy. For instance, during the simulation minute, there is no spawned CAV who turns left. The fitting curve hides these kinds of advantages or disadvantages, evaluating the policy in general. Each policy is simulated with the following traffic flows given in table 3.3 during 10min, six times. The capacity of a lane is 0.5m/s. In each lane, there are 10% of CAVs who turn left, 80% who go straight and 10% who turn right.

3.3.4/ SIMLATION PURPOSE

In this section, we introduced the implemented intersection simulator. This simulator aims to compare the different policies in terms of traffic efficiency. In this simulator, we separated CAVs and the intersection manager. However, it is important to note that there are no simulated communication delays or packet loss. These parameters are integrated in the value of $\tau = 0.5s$ that represents the global delay due to sensors, data fusion, control algorithms, communication.... In practice, the value of τ needs to be respected to get similar results than the ones will be presented hereafter. Otherwise, τ must be thoroughly evaluated according to the existing materials.

To the best of my knowledge, the used metrics are original for such an application. Contrary to traffic lights where the designer tries to reach a given capacity through cycle time and phases, CIM is a fully autonomous flow of CAVs. The metric suite better than the usual KPI, for the studied case. The simulated intersection is an elementary one that includes the turning movement. This has the advantage to consider the complexity raised by such an intersection. Some of these complexities (The three movements in the same lane) are avoided, while using intersections with more lanes.

3.4/ POLICY

In this thesis, we have simulated many policies. However, only a few were interesting, in terms of safety and efficiency. In the reminder, we will present the results and the associated algorithms to the following ones:

- FIFS
- FRO
- VP DCP

To the best of my knowledge, FRO and VP DCP were introduced neither to virtual platoon protocol nor to the reservation one, before our first paper [71]. In practice, they raise the issue of adjusting the cruise control to dynamic scheduling. More precisely, a CAV can have a new obstacle during its movement in the storage zone. As discussed in section 3.2.4, the new OPL is updated only if all conflicting CAVs acknowledge the new order for VP-DCP, whereas a centralized architecture is used for FRO.



Figure 3.11: Simulation of FIFS policy under the virtual platoon protocol

3.4.1/ FIFS

By default, the simulator gives the results of FIFS under the virtual platoon protocol. The OPL is built dynamically. At each new CAV message, the CAV is ranked at the end of the list. So, FIFS ranks CAVs according to their arrival date to the conflict zone, at the last position. The most priority CAV is the one that arrived first, and so on. Each CAV forms a set of *vl* and *nl* by reading the OPL from their rank position to the first CAV. A CAV in *nl* is ignored only if it is totally removed from the OPL, whereas a CAV in *vl* is dropped from the set if it has already freed the potential zone of collision. With this simple behavior, FIFS policy is respected.

The result of FIFS simulations is given in figure 3.11. One can note from the points presented in the figure, that only a few simulations are between the extreme smooth part and the extreme congested part of the graph. The majority of points are in the congested part of the traffic. This is due to the small intersection capacity. Its maximal cumulative throughput is averagely near to 0.81 upc/s. The half of simulation runs are over the capacity of the intersection under the FIFS policy (see table 3.3).

3.4.2/ FRO

FRO is an extension of FIFS by allowing CAVs to modify their rank according to their estimated exit time. The used architecture of the negotiation is centralized. Simply speaking, the intersection manager modifies the ranks of CAVs. Getting a safe FRO, is not an easy task. FOS faces deadlock, livelock and collision risks. Three strategies were used to avoid the deadlock and livelock:

- A step-by-step approach: The OPL is frequently ordered from the bottom to the top. Each CAV is ranked according to the immediate precedent CAV in the OPL. A swap happened, only if the exit time of the CAV is smaller than the precedent, and so on. This allows to avoid a livelock situation, with permanent swaps.
- Exit time constraint: The exit time of a given CAV is always bigger than its real leader *rl*. The computed exit time is at least bigger or equal to the exit time of CAV_{*rl*} plus 2*s*. This allows to prevent the intersection from the deadlock presented in figure 2.11-B, where the follower is ranked before the leader that is moving in the same storage zone.
- Exit time evaluation: The exit time is evaluated according to the conflict zone, rather than using the potential zones of collisions. This is similar to the approach used in [59]. This allows preventing the deadlock presented in figure 2.11-A. This deadlock is owed to a circular wait where at least three CAVs are involved. Even if the use of OPL prevents circular wait, the dynamic updates of the OPL that is raked according to exit time of the potential zone of collisions will slow down CAVs by making several swaps in the same group of CAVs.

Regardless of the exit time of the *rl*, the exit time is computed as follows:

$$t_{i,exit} = \min\left(\underbrace{\frac{d_{i,exit} - p_i(t)}{v_i(t)}}_{(1)}, \underbrace{\sqrt{\frac{2(d_{i,exit} - p_i(t))}{\overline{u_i}}}_{(2)} + \tau}_{(2)}\right)$$
(3.18)

The left term of the minimum in equation (3.18), is used when the CAV is moving $(v_i(t) >> 0)$, whereas the right one is used when the state of the CAV is near to a complete $stop(v_i(t) \rightarrow 0)$. For avoiding collisions in FSO, if the immediate precedent CAV (CAV_j) is in conflict, the swap is allowed only if it is able to come to a complete stop before the stop line. The swap happens only under the following condition:

$$p_j(t) < 80 - \left(v_j(t)\tau + \frac{v_j(t)^2}{2\underline{u_i}}\right) + s,$$
 (3.19)

with $p_j(t) = 0$ at the beginning of the conflict zone and *s* is a safety distance that equals 2m in the simulator.

The results of the simulations of FRO are given in figure 3.12. One can observe that the FRO improves the intersection throughput compared to FIFS. The simulation points are



Figure 3.12: Simulation of FRO policy under the virtual platoon protocol

more homogeneously distributed. The average throughput capacity of the intersection is close to 0.96upc/s. The scatter plot around the curve is wide, due to the fact that the efficiency of the algorithm strongly depends on the destinations of CAVs. On the simulator interface, as the flow increases, the user can notice that the traffic is smooth for only one part of the traffic, whereas the remainder is at a standstill. The main disadvantages of the FRO are as follows:

- It continues the evacuation of the lanes that are not loaded, as long as the space between two exiting vehicles does not exceed the time needed to the evacuation of the first CAV of the overloaded lanes at a standstill (See equation 3.18₋(2)).
- It is difficult to implement when the next intersection is saturated. In this paper there is no next intersection, so the exit time does not include the presence of *nl*.

The first disadvantage is solved using DCP. This protocol deserves to be studied and extended to the virtual platoon protocol. The second one is studied using DCP in [122].

3.4.3/ VP-DCP

This section analyzes DCP and extends it to the virtual platoon. First, let us simulate DCP, using the stop and go protocol. The architecture is centralized. The intersection manager receives the requests from all CAVs and the OPL is formed only by CAVs that

aren't in conflict. A CAV in the OPL is not dropped unless it exits the conflict zone. There is neither *nl* nor *vl*, since all CAVs in the OPL aren't in conflict. A CAV is included in the OPL by using the following rules:

- **First CAV:** If the CAV is the first one at the intersection (empty OPL), it is added into the OPL (allowed to pass through the intersection) and gets the green.
- Follower: If there is a CAV_j on the same lane close behind a CAV_i in the OPL (p_i(t) − p_j(t) ≤ d_f), is considered as a follower. The follower is included by default in the OPL (it receives the green of the precedent CAV) if there is not any conflict with all already CAVs in the OPL.
- **Parallel (mapped):** Any other CAV is included, if both following conditions are satisfied:
 - There is not any conflict with all already CAVs in the OPL
 - The CAV is nearer to the conflict zone than the furthest authorized CAV (e.g. follower).
- Efficiency and fairness: If the last CAV left the conflict zone, the green is given to the nearest CAV. If there are two CAVs with the same temporal distance, the green is given the oldest one.

One of the crux issue is the definition of d_f to admit a CAV as a follower. To this end we used the average headway time between two following CAVs wich is near to 2*s*, for which we added the CAV's length and a safety distance *s*. However, as the speed of the CAV is close to 0, as the value of the distance become shorter. Hence, we considered the following value of d_f :

$$d_f = \min\left(2 \cdot v_j(t) + 4.4 + 2, 15\right),\tag{3.20}$$

with j is the index of the follower CAV.

The simulations of DCP under the stop and go protocol give the result presented in figure 3.13. One can observe that there are a few points near the capacity. This means that some of them are on the left side of the graph, causing congestion. More precisely, DCP under the stop and go protocol is sensitive to the CAV's destination when the traffic flow is high (more than 0.23upc/s per lane). However, even with the stop and go protocol that makes CAV resort to a complete stop, DCP has a slightly bigger capacity than FRO with virtual platoon protocol. The average throughput capacity of the simulated intersection equals to 0.98upc/s. This encourages us to extend DCP to the virtual platoon protocol.

There are two golden rules of DCP to increase the traffic throughput, which are as follows:

• Follower: Two successive CAVs from the same lane spent less time than two conflicting ones



Figure 3.13: Simulation of DCP under the stop and go protocol

• **Parallel/mapping:** It is better to increase the number of CAVs that can pass in parallel

Both rules need to be included in the DCP-VP (DCP under the virtual platoon). Remember that there is no red or green in the virtual platoon. In order to explore both rules, the CAV needs to be ranked the closest either to its leader on the same lane or to a CAV that can simultaneously pass through the conflict zone.

Let us call the CAV of the first option a real leader and the CAV of the second option as a mapping leader. An example of the ranking problem is presented in figure 3.14. The green CAV needs to be added to the OPL. There are three possible ranks, i.e. behind the blue CAV, the amber one or the red one. If all CAVs go straight, the amber CAV is the mapping leader *ml*, whereas the blue one is the real leader. In such a case, whatever the position of the green CAV before the red CAV, will have the same result.

Conditions and new rank finding

However, if both leader CAVs, i.e., blue and red, are far from the green one, it is not efficient to modify the sequence. To this end, it is important to define a relative distance from which the green CAV can consider the other as a leader to join. d_f defined in equation (3.20) is considered as the maximum allowed distance for both real leader (blue CAV) and mapping leader (Amber CAV). In addition to the relative distance, a CAV can be considered as a leader, either a real leader or a mapping leader, only if it is in the storage zone.



Figure 3.14: Scheduling problem in DCP-VP

The algorithm for searching a leader for changing the rank begin from the current rank of the ego-CAV $_j$ and go progressively to the beginning of the OPL. The searching stops at the rank of the real leader. The first found leader, while searching, either mapping or real one is the one chosen. This leader $_i$ needs to fulfill both conditions :

- CAV_i is in the storage zone
- $p_i(t) p_j(t) \le d_f$

Simply speaking, even if the green CAV is close to the blue one, it will select the amber one if it fulfills both conditions. This aims to make the CAV gain a new position progressively.

Negotiation of the new rank

The CAV negotiates with conflicting CAVs that are located in the OPL between the rank of the chosen leader and the current rank of the CAV. The negotiation begin with the last one and go to the rank of the chosen leader. If one conflicting CAV refuses, the negotiation stops and the last gained rank is kept. For safety reasons and for avoiding livelock, the conflicting CAV refuses, if at least one of these conditions aren't fulfilled:

- · It has a conflict with the chosen leader
- · It doesn't need to decelerate to admit the precedent CAV
- It is in the storage zone.



Figure 3.15: Simulations of DCP-VP

The first rule with the second one aims to avoid a frequent swap that makes both CAVs slow down. The third rule avoids collisions.

Other improvements

Another rule is added to rank CAVs in the OPL. This rule gives a higher priority to CAVs without conflict. If the CAV that is ranked before is not conflicting, and it is further from the conflict zone, the ranks are swapped. This task is performed by the intersection manager, before sending the updated OPL. The main objective is to avoid useless negotiations with CAVs that can cross the intersection without braking.

Simulations

The simulation results of the DCPVP are given in figure 3.15. One can note from the figure that there are fewer points on the congested part. Some points are very high (high throughput) near the capacity. This is due to the schedule that allows four movements crossing the intersection simultaneously (turning right). The average maximum throughput is 1.48upc/s. This is significantly much higher than the previous results.

3.4.4/ COMPARISON

In order to compare the four studied intersection management, figure 3.16 presents all obtained fitting curves. One can observe that DCPVP significantly enhance traffic management from two points of view:



Figure 3.16: Comparison between the four simulated intersection management

- · Average speed: The CAVs go faster for any value of traffic flows
- throughput: The intersection allows a high capacity.

The second interesting performance in terms of the capacity is DCP with stop and go protocol. However, CAVs move slower in general and their capacity is too close to FRO. Moreover, in DCP simulation, there are few points near the capacity. This is due to the fact that CAVs in DCP is resorted to a complete stop, whereas in FRO they have the advantage of forming virtual platoons. Finally, FIFS is the worse one.

3.5/ CONCLUSION

In this chapter, we tackled the issue of scheduling virtual platoon. First, we defined the way the CAVs adjust their speeds. To this end, we defined the synchronization point, the cooperative cruise control as well as the communication architecture. In order to compare different policies, we built a simulator and defined a metric that allows to evaluate policies from the traffic engineering standpoint.

Three policies are compared, using the virtual platoon protocol, FIFS, FRO and DCPVP. We also simulate DCP with the stop and go protocol. The simulation shows that both scheduling and cruise control have a significant impact on the intersection performance. A notable gain in terms of speed and capacity is achieved through DCPVP. If we compare DCPVP to FIFS that has the same cruise control, DCPVP increases the throughput by 81%. The comparison between DCPVP and DCP with the stop and go protocol, shows that the speed adjustment improves the capacity by 50%. These results motivate us to continue our studies. Indeed, in this chapter, we put the synchronization point at the end of the storage zone. The following chapter aims to minimize the time between conflicting vehicles.

4

OPTIMAL VIRTUAL PLATOON

4.1/ INTRODUCTION

CAVs introduce the following new optimization opportunities, in intersection management:

- Sequence formation/Scheduling: CAVs are able to communicate together in order to determine which CAV crosses the intersection first, which one the second and so on,
- *Trajectory planing*: The accelerations of CAV can be optimized according to the precedent CAVs.

Cooperative Intersection Management (CIM) aims at exploring both opportunities for enhancing the traffic situation. In CIM, each CAV contributes to the decision-making by negotiating the "right-of-way" and by adjusting its speed accordingly.

In the last chapter, we considered the virtual platoon protocol, while trying to optimize the sequence formation through CAVs negotiation. We showed that VP-DCP improves significantly the intersection throughput. However, this work needs to be improved in terms of trajectory planning. Indeed, on the one hand, a significant performance was gained by modifying the protocol from the stop and go to the virtual platoon. On the other hand, it remains a possible improvement of the cooperative cruise control by reconsidering the synchronization point.

Recall that in the previous chapter, CAVs need to keep a safe distance at the beginning of the conflict zone. This aims to efficiently use the storage zone for the sequence formation through the negotiation of their rank. This chapter investigates if there is a better synchronization point to minimize the time between two conflicting CAVs. To this end, it studies an elementary intersection of two robots. This leads to a mixed protocol (virtual platoon/reservation). The results of the study will be generalized to the studied intersection of CAVs.


Figure 4.1: Key zones of intersection.

This chapter is organized as follows. The second section introduces the problem. The third one focuses on the safety time between two intersecting movements of robots. This introduces an optimal control point. Simulations using an optimal schedule are performed. The third section uses the optimal control point for the studied intersection of CAVs. Simulations and comparisons are presented, before the conclusion.

4.2/ PROBLEM STATEMENT

Because we intend to address the CIM safety problem in a general case (industrial, domestics and urban environments), in the remainder we use the term robot instead of CAV. As defined in the first chapter, the intersections commonly have a storage zone, a conflict zone and an exit zone. The conflict zone covers the set of conflict points. These points are the result of the intersection with a non-zero degree angle of two lines representing the vehicle movement. Each conflict point defines a potential collision area in which two conflicting vehicles cannot access it simultaneously. The shape of this area depends on the space occupied by the vehicles and the curvature of the lines that intersect. Each potential collision area forms an elementary intersection. An example is given in figure 4.1, starting from a usual intersection to the elementary intersection. Let us now consider the elementary intersection as showed in figure 4.1. The average speed of each robot



Figure 4.2: Two kinds of headway h: (a) with the precedent vehicle that moves on the same lane called h_f , (b) with a conflicting vehicle called h_c , $h_c > h_f$.

crossing the elementary intersection strongly depends on the safety constraint as follows:

$$V = \frac{D}{T_{min} + T_s},\tag{4.1}$$

where *D* is the length of the trajectory from the beginning of the storage zone to the exit of the conflict zone, T_{min} is the traveling time of the robot at its maximal speed (\overline{v}) and T_S is the time required to avoid collisions with other precedent robots that are either moving in the same lane or in the conflicting lane. Since *D* and T_{min} are constant, T_S is in the core of the optimization of CIM. T_S is the result of the following addition:

$$T_s = h + L \tag{4.2}$$

where h is due to safety distance kept with the precedent robot and L is the time lost to respect the distance.

When the precedent robot is moving in the same lane, T_s is widely studied in the platooning literature [182]. A safe headway time *h* is maintained all the time to avoid collision with the precedent robot. *L* results from the latency of the control system to recover *h*. When the robot and the precedent one form intersecting movements, *L* strongly depends on the speed profile of the robot. For instance, if the robot comes to a complete stop before the potential collision area for letting the precedent robot go, a significant time to clear the conflict zone is needed later, resulting in a big *L* [54] and [183] (see figure 4.2).

Many papers [??] assume that the trajectory optimization makes *L* tend towards 0. More precisely, thanks to the trajectory optimization, the second robot anticipates the exit time



Figure 4.3: Elementary conflict between two robots: The remained distance here indicates $D - p_1(t) - l_c - l_1 - s_1$

of the precedent one, in order to immediately cross the conflict point at the desired speed. However, such an assumption is not safe when it comes to real robots. The second robot cannot exit immediately after the first one because it must be able to react if the first one comes to a complete stop or does not respect exactly its exit time. Hence, a buffer time [39] and a clock synchronization approaches [68] are studied later to overcome the unpredictable robot trajectory. This contributes to increase L.

In order to study the problem, we consider in the following two conflicting robots R0 and R1 presented in figure 4.3. One can observe that the conflict movement raises the following confusion:

- The collision risk is raised only momentary (only when R0 occupies the conflict point figure 4.3-A-B-C). When R0 leaves the potential collision area, R1 can get into the potential collision area immediately without collision risks.
- However, the safe distance that is momentary kept by R1 to avoid collision with R0 when it occupies the potential collision area will delay the exit time of R1 (L > 0). This prevents R1 to enter exactly when R0 leaves.

More precisely, a safe speed profile allows R1 to be able to come to a complete stop if R0 does not respect its exit time and stays at the potential collision area. Hence, the optimization of R1 performance resorts in the minimization of λ presented in figure 4.3. We draw the reader attention that h_c and λ are different, but the minimization of λ is

easier to formulate mathematically. A correct evaluation of λ and its minimization, not only improves R1 exit time but also significantly improves the intersection performance through a suitable sequence (which robot goes first and which is the second and so on), as it will be discussed later.

4.3/ OPTIMAL CONTROL POINT

This subsection gives the formulation of the optimal control of the trajectory of R1 when it intersects with R0. To this end, this section defines:

- The optimal control point (see section 4.3.1): the speed and the distance that R1 must take in order to cross safely the intersection with a minimum λ. More precisely, we set the optimal red point in figure 4.3-C that allows R1 to exist the soonest.
- The formulation of the problem of the optimal speed profile according to a given objective function (see section 4.3.2). Here, besides safety concerns, we consider the energy minimization.

Notation	Meaning
l_c	The length of the potential collision area
D_i	The position of the complete exit of R <i>i</i> from the potential collision area
τ	Reaction time of the system
l_i	The length of Ri
Si	The desired distance with obstacle when Ri comes to a complete stop
$p_i(t)$	The distance traveled at t by Ri from its entrance to the storage zone
$v_i(t)$	The speed of Ri at t.
$\overline{v_i}$	The maximum speed of R <i>i</i> .
\underline{u}_i	The acceleration lower bound
$\overline{u_i}$	The acceleration upper bound of Ri.
$u_i(t)$	The acceleration Ri at time t.
t _{i,enter}	The time when R_i enters into the potential collision area.
t _{i,exit}	The time when R <i>i</i> leaves the potential collision area.

Table 4.1:	Elementary	intersection	notations
14010 1.1.	Lionnonitary		notationo

Notations used hereafter are presented in Table 4.1. We draw the reader's attention to the following folds:

• *s* is extended to consider both the positioning system accuracy and the approximations (see footnote 1).

- τ is the upper bound of the amount of time needed to the robot to respond to the stimulus, including the communication, detection and computation time as well the time needed to apply the control (acceleration).
- R*i* state at time *t* is determined by its position and speed, $X_i(t) = [p_i(t), v_i(t)]^T$ whereas the acceleration $u_i(t)$ is its control input.

4.3.1/ PRELIMINARIES OF OPTIMAL AND SAFE TRAJECTORY

In order to allow R1 to come to a complete stop when R0 is in the potential collision area, the minimum safe distance that R1 must keep to the exit is the following:

$$D - p_1(t) \ge \underbrace{v_1(t) \cdot \tau}_{d_{\tau}} - \underbrace{\frac{v_1(t)^2}{2 \cdot \underline{u}_1}}_{d_u} + \underbrace{l_c + l_1 + s_1}_{d_z}^{1}$$
(4.3)

with $t \in [t_{0,enter}, t_{0,exit}]$. In (4.3), d_{τ} considers the travelled distance during the reaction time, whereas d_u is the braking distance. d_z includes the physical ground occupation of the robot as well as the length of the potential collision area. λ is the time needed to travel the remained distance to the exit located at D. λ is determined by R1 state at $t = t_{0,exit}$.

By considering that R1 accelerates at the maximum when R0 releases the potential collision area, λ is a function of $X_1(t_{0,exit})$. Under the assumption of the equality in (4.3), the minimization of λ is obtained by searching the roots of its derivative [184]. This gives the analytical expression of the optimal control point $X_i^* = [p_i^*, v_i^*]^T$ as follows:

$$X_{i}^{*} = \begin{bmatrix} D - v_{i}^{*} \cdot \tau + \frac{v_{i}^{*2}}{2 \cdot \underline{u}_{i}} - d_{z} \\ \min\left(\overline{v_{i}}, \underline{u}_{i} \frac{\sqrt{\overline{u_{i}} \cdot \underline{u}_{i} \cdot \tau^{2} - 2(\underline{u}_{i} - \overline{u}_{i})d_{z}} - \overline{u_{i}} \cdot \tau}{\underline{u}_{i} - \overline{u_{i}}} \right) \end{bmatrix}$$
(4.4)

Figure 4.4 illustrates λ variations according to the speed. Each speed value is linked to a position, by solving the equality in the equation (4.3). The safe distance is presented by the dashed line in Fig. 4.4 and the values can be read on the right axis. The curve shows the optimal state. When R1 moves too fast, it must take a big distance to the potential collision area, which requires more time to exit. On the opposite side, when the robot has a small speed, it is too near to the potential collision area, but it needs more time to exit because of its low speed. The balance between safe distance and speed is presented in (4.4). Hence, in order to minimize λ (*L* as well), R1 must be in the state X_1^* at $t = t_{0,exit}$. In the following, the optimal state given in (4.4) is called the control point that R1 must reach thanks to DMPC. It remains to define a sufficient condition that allows a safe behavior of

¹There is no explicit solution to the differential equation $A = x(t) + x(t + \tau) - \frac{\dot{x}(t)^2}{2\cdot \underline{u}}$ with A a constant in \mathbb{R} . Hence, equation (4.3) is an approximation. s includes $\frac{1}{2}\overline{u_1} \cdot \tau$



Figure 4.4: λ variation according to the $v_1(t_{0,exit})$

the robot before getting a safe control point and to study the feasibility.

Safe robot behavior

When $t \in [t_{0,enter}, t_{0,exit}]$, R0 is in the potential collision area and R1 should be able to achieve a complete stop without getting into the potential collision area. Formally, (4.3) must be respected, before the exit of R0. The following lemma gives the safety bound:

lemma¹ (Safety constraints). For a given point $X(t^*)$ that belong to the the following constraint set $C_{\underline{u}}$:

$$X(t) = \begin{bmatrix} p(t) \\ v(t) \end{bmatrix} \in \mathcal{C} := p(t) \ge v(t) \cdot \tau - \frac{v_1(t)^2}{2 \cdot \underline{u}} + A,$$
(4.5)

with $A \in \mathbb{R}$, all points of the trajectory from $t^* - T$ to t that leads to $X(t^*)$, respecting both constraints:

$$u(t) \ge \underline{u} \tag{4.6}$$

and

$$v(t) \ge 0 \tag{4.7}$$

belongs to $C_{\underline{u}}$, with $T > \tau$ and $u(t) = \frac{dv(t)}{dt}$.

Proof. In order to prove the lemma we prove first the equality in (4.6). Let $a = \underline{u}$ be the acceleration leading to $x(t^*)$ from $x(t^* - T)$. If the equality holds,

Hence, we have:

$$p(t^{*}) = v(t^{*} - T) \cdot \tau + \frac{1}{2} \underline{u} \cdot (T - \tau)^{2} + v(t^{*} - T) \cdot (T - \tau) + p(t^{*} - T)$$
(4.8)

and

$$v(t^*) = \underline{u} \cdot (T - \tau) + v(t^* - T)$$
(4.9)

Since $v(t^*)$ and $v(t^*)$ respect (4.5), then:

$$-v(t^* - T) \cdot \tau - \frac{1}{2}\underline{u} \cdot (T - \tau)^2 -$$

$$v(t^* - T) \cdot (T - \tau) - p(t^* - T)$$

$$\geq (\underline{u} \cdot (T - \tau) + v(t^* - T)) \cdot \tau$$

$$- \frac{\underline{u} \cdot (T - \tau) + v(t^* - T))^2}{2 \cdot \underline{u}} + A$$

After a simplification, we have:

$$-p(t^* - T) \ge \underbrace{\left(\underline{u} \cdot (T - \tau) + v(t^* - T)\right)}_{v(t^*)} \cdot \tau$$
$$+ v(t^* - T) \cdot \tau - \frac{v(t^* - T)^2}{2 \cdot u} + A$$

with $v(t^*) \ge 0$ and $\tau \ge 0$ the equality in (4.6) is proved. Second, in order to prove the inequality $(u(t) < \underline{u})$, it is sufficient to prove that $C_{u_1} \subset C_{u_2}$ if $u_1 \ge u_2$. To this end, consider $X_1(t) \in C_{u_1}$. Hence, we have from (4.5):

$$u_1 \le \frac{v_1(t)^2}{2 \cdot (p_1(t) + v_1(t) \cdot \tau + A)}$$

Since $u_2 \le u_1$, then $X_1(t) \in C_{u_2}$, which proves the inequality.

In other words, lemma 1 tells us that to keep the trajectory of R1 safe to a safe control point $X_1(t_{0,exit})$, it is sufficient that the *initial point is safe*, the trajectory *respects the acceleration lower bound* $(u_1(t) \ge \underline{u}_2)$ and its *speed is kept positive* $(v_1(t) \ge 0)$.

Feasibility of the control point

The feasibility of the control point depends on the following conditions:

- Cond1: $t_{0,exit}$ allows enough time to R1 to reach simultaneously both optimal distance p^* and speed v^* given in (4.4).
- Cond2: $p_1(t) p^*$ allows enough space to reach the control point.



Figure 4.5: Unfeasible optimal point (in red): A-Cond1 is not satisfied, B-Cond2 is not satisfied. The remained distance here indicates $D - p_1(t) - l_c - l_1 - s_1$.

Figure 4.5 illustrates the situations when each one of these conditions is not satisfied. In simple words, Cond1 is not satisfied when the robot R1 is too far from the control point (see figure 4.5-A), whereas Cond2 isn't fulfilled if the robot R1 is too close (see figure 4.5-B). Hence, the feasibility of the control point is written as follows:

$$D_{\text{Cond2}} \le p_1^* - p_1(t) \le D_{\text{Cond1}}$$
 (4.10)

with both $D_{\text{Cond}i}$ being analytical functions of $t_{0,exit}$, $v_1(t)$, v_1^* and of the physical limitations of the robot: \underline{u}_i , \overline{u}_i and \overline{v}_i . These conditions can be either checked formally through the optimal trajectory control, or directly as a feasibility output of the quadratic programming used to solve the DMPC presented in the next section.

4.3.2/ TRAJECTORY CONTROL DESIGN

In the following we call a safe state of R1 a state that respects (4.3) when $t \le t_{0,exit}$. A trajectory is safe if all states of the trajectory are safe. To keep the trajectory of R1 safe to a *safe control point*, it is sufficient that the *initial point is safe*, the trajectory *respects the acceleration lower bound* ($u_1(t) \ge u_2$) and its *speed is kept positive* ($v_1(t) \ge 0$).

When $t \le t_{0,exit}$, R1 considers two control values called: u_s and u_{opt} . They are computed at each time step τ . u(t) is defined as follows:

$$u_1(t) = \min(u_s(t), u_{opt}(t)) \text{ with } t \le t_{O,exit}$$

$$(4.11)$$



Figure 4.6: Illustration of both feasibility conditions to reach $v^* = 5.04m/s$ from distance to p^* according to $t_{0,exit}$: $\tau = 0.5s$ and $v_1(0) = 10m/s$

When $t > t_{0,exit}$ (After R0 confirms that the potential collision area is cleared), R1 accelerates at the maximum $\overline{u_1}$ to reach the maximum speed $\overline{v_i}$.

 u_s is applied when at least one of both conditions Cond1 and Cond2 is not respected. u_s simply makes the robot R1 come to a complete stop before the potential collision area. This is because the feasible point is either before (\neg Cond1) or after (\neg Cond2) the optimal control point. u_{opt} is applied when both conditions are fulfilled. Otherwise, the robot set $u_{opt} = +\infty$ and does not compute it. In order to compute u_{opt} , we use in this section the Model Predictive Control (MPC). Other approaches can be used, such PID controlled [185] or optimal trajectory control based on Pontryagin's maximum principle [155].

The dynamic of each robot R*i* is modeled as a discrete time system for which the time step $\Delta t = \tau$. The robot state at each time step *k* is described as follows²:

$$X_{i}(k) = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix} \cdot X_{i}(k-2) + \begin{bmatrix} \frac{1}{2}\tau^{2} \\ \tau \end{bmatrix} u_{i}(k-1)$$
(4.12)

Under the following constraints:

$$0 \le v_i(k) \le \overline{v_i} \tag{4.13}$$

for the speed and

$$\underline{u}_i \le u_i(k) \le \overline{u_i} \tag{4.14}$$

for the acceleration. The studied objective function is the minimization of the energy,

 $^{^{2}}k - 2$ is due to the system delay

derived from[186]. It is defined as follows:

$$J = \frac{1}{2} \sum_{i=0}^{N-1} u_1(t+i \cdot \tau)$$
(4.15)

with $N \cdot \tau$, is the time horizon of optimization, such that $N \cdot \tau \ge t_{0,exit}$.

In order to obtain the optimal $u_1(t + i \cdot \tau)$, we use quadratic programming [187]. To this end we have the following problem to solve:

$$\begin{array}{ll} \underset{\mathcal{X}}{\text{minimize}} & \frac{1}{2} \mathcal{X}^{T} \mathcal{Q} \mathcal{X} \\ \text{subject to} & A \mathcal{X} = b \\ & G \mathcal{X} \leq h \end{array}$$
(4.16)

with $X^T = [p_1(t), v_1(t), u_1(t), \dots, p_1(t+N\tau), v_1(t+N\tau), u_1(t+N\tau)]$. *Q* is a positive semi-definite diagonal matrix in order to obtain *J* as defined in (4.15). *A* and *b* are defined to have the current state of the robot at *t*, the optimal control point at $t + N\tau$ and to respect the robot dynamic (4.12). Finally, *G* and *h* are built according to speed and acceleration constraints described in (4.13) and (4.14), respectively.

For safety reasons, slack variables are not used because the safety constraint (4.3) might not be respected with them. The studied Cond1 and Cond2 avoid unfeasible quadratic programming problems. If one of both conditions is not respected u_s keeps the R1's trajectory safe.

4.3.3/ NUMERICAL EXAMPLE OF OPTIMAL TRAJECTORY CONTROL

First, we consider an elementary intersection with two robots. R1 parameters are given in Table 4.2. The optimal control point $X^{*T} = [D - 15.76m, 5.04m/s]^T$ is obtained from (4.4). R1 is able to totally clear the potential collision area after $\lambda = 2.35s$ at 8.74m/s. Figure 4.6 gives the feasibility area of the control point according to the remained time and distance to get X^* . At each $t_{0,exit}$, Cond1 gives the upper bound of the distance to p^* and Cond2 the lower bound. Hence, all points between the two lines, allow R1 with $v_1(0) = 10m/s$ to reach the control point.

In order to illustrate the control process, let us consider that $t_{0,exit} = 6s$. Feasible distances to p^* are in the range of [28.5*m*, 64.5*m*]. Figure 4.7 compares the speed profiles of R1 with five initial distances to p^* . In this figure, we assume that R1 accesses to the potential collision area at $p_1(t) = 73.76m$ and totally frees it at $p_1(t) = 80.76m$. More details about the five scenarios are given in Table 4.3.

Solid lines in figure 4.7 show the speed profile of R1 when R0 exit at $t_{0,exit} = 6s$. In

Table 4.2: R1 parameters

Parameters	l_c	τ	l_1	<i>s</i> ₁
Values	3 <i>m</i>	0.5 <i>s</i>	4 <i>m</i>	2 <i>m</i>
Parameters	$v_1(t=0)$	$\overline{v_1}$	\underline{u}_1	$\overline{u_1}$
values	10m/s	15m/s	$-3m/s^2$	$2m/s^2$



Figure 4.7: R1 position, speed and acceleration according to initial conditions

scenarios SC1 and SC2, R1 does not fulfill Cond1 and Cond2, respectively. R1 exclusively uses u_s . In SC2, SC3 and SC4, both constraints are respected. R1 uses u_{opt} . One can

Scenario	SC1	SC2	SC3	SC4	SC5
Initial position	0 <i>m</i>	0.5 <i>m</i>	18.5 <i>m</i>	37 <i>m</i>	37.5 <i>m</i>
Initial distance to the control point	65 <i>m</i>	64.5 <i>m</i>	42 <i>m</i>	29 <i>m</i>	29.5m

Table 4.3: Five studied scenarios

Table 4.4: Intersection of 6 robots

Scenario	R0	R1	R2	R3
Lane	L0	L1	L0	L1
Arrival times	0 <i>s</i>	0S	2 <i>s</i>	2 <i>s</i>

observe that if the control point is feasible, the robot frees faster the the potential collision area. Besides, for different initial states of R1, after $t_{0,exit} = 6s$, R1 has the same speed profile in SC2, SC3 and SC4.

Dashed lines in figure 4.7 show the speed profile of R1 when R0 does not send an exit acknowledgement at $t_{0,exit} = 6s$ and still occupy the potential collision area for indeterminable time (e.g. R0 breakdown). In all scenarios, R1 is able to come to a complete stop before the potential collision area and a side collision is avoided.

4.3.4/ SEQUENCE OPTIMIZATION

In this subsection, we consider several robots that cross the potential collision area. We present the following approach to minimize the maximum exit time of a set of robots $\min(C_{\max}(t_{i,exit}))$. To this end, we will use the dynamic programming presented in [85].

The dynamic programming, solves the problem from a set of smaller problem and so on. The sub-problem state is defined as follows:

$$v = \{n_1, \cdots, n_j, \cdots, n_l, k\}$$
 (4.17)

The state gives the number of robots coming from the lane *j* that have already left the intersection, with *l* the number of lanes. *k* in 4.17 designates the lane of the last robot that left the intersection. Hence, $n = \sum_{j \in [0,l]} n_j$ gives the number of all the robots that left the intersection.

Consider that each state is linked through an arrow *a* to all states which have an additional robot that left the intersections in a given lane. Each arrow is valuated according to either λ or headway. We obtain a valued digraph $G = (V, A, \Phi)$, with *V* the set of vertices, *A* the set of arrows and Φ the valuations. There is a one-to-one mapping between all possible



Figure 4.8: Intersection of six robots: Position, speed and acceleration

combinations of exit sequences and paths from $\{0, \dots, 0, \dots, 0\}$ to $\{N_1, \dots, N_j, \dots, N_l\}$, where N_j is the total number of the robots in the lane *j*.

The optimal sequence that minimizes the maximum exit time of robots is computed by searching the shortest in the graph. The computation of the shortest path is bounded by the following number of steps:

$$2 \cdot l \cdot (\max_{j \in [0,l]} (n_j) + 1)^l$$
(4.18)

With two conflicting lanes, it is possible to solve the optimization problem, with at most $O(n^2)$ step and memory size.

In order to illustrate the approach, let us consider a scenario with four robots: R*i* with $i \in \{0, \dots, 3\}$. Except their initial speed that is equal to $\overline{v_i}$, i.e. $v_i(t_{init}) = 15m/s$, the four robots have the parameters presented in Table 4.2. Even robots (R0, R2) are coming from lane L0 and odd robots (R1, R3) are coming from L1. Their arrival time t_{int} is presented in table 4.4.

At 15m/s, the safe headway time between two robots that move in the same lane is two seconds h = 2s [188]. The length of the storage zone is 98m (distance of the potential collision area from the origin) in both lanes. D = 105m for all robots. Considering $\lambda' = 2.83s$, $\lambda = 2.35s$ and h = 2s, the shortest path algorithm gives the optimal sequence presented in figure 4.8-C. $\lambda' = 2.83s$ is due to the fact that if R0 or R1 are scheduled at the second rank, the initial state of both does not fulfill the first condition Cond1. Figure 4.8-A and figure 4.8-B give the trajectory of robots according to the shortest path.

The vertices in the figure 4.8-C are colored according to the robot number. The black

values on the left side of vertices give the time when the robots are ready to free the potential collision area if there is no conflict. For example, for the vertex $v = \{2, 1, 0\}$, 9*s* indicates that R2 is ready to free the potential collision area at t = 9s. The red values over the vertices are the minimum exit time computed according to the shortest path from $\{0, 0\}$ to the corresponding vertex. The arrows are valued according to λ if a conflicting robot is scheduled after. For instance, the arrow linking $\{1, 1, 0\}$ (i.e. R0) to $\{1, 2, 1\}$ (i.e. R3) has a value of 2.35*s*. The arrow valuation equals 2*s* if both robot are coming from the same lane, as between $\{1, 1, 0\}$ and $\{2, 1, 0\}$. One can note from figure 4.8-C that there are two possible optimal sequences: R0 >R2 >R1 >R3 and R1 >R3 >R0 >R2.

Both optimal sequences show that it is interesting to group together robots that are not in conflict. This is similar to DCP rules. Thus, since the robots have a trajectory allowing them to stop before the potential collision area, it is possible to extend distributed heuristics like the ones proposed in [54], [71] and [188] to set groups of robots that cross together the intersections. These approaches are similar to the presented DCPVP in the precedent chapter. However, in order to improve the intersection performance, the optimal control point needs to be added.

4.4/ OPTIMIZED COOPERATIVE CRUISE CONTROL

The studied intersection of robots focus on only one potential conflict point. Moreover, the used optimization objective in the quadratic programming makes CAVs begin deceleration too early in the storage zone (see figure 4.8-B). This may prevent from forming groups and, above all, this slows down the upstream intersection. In order to solve both problems, we modify the synchronization point introduced in section 3.2.1. The main idea is to use a synchronization point that allows the CAV to reach the optimal control point when the conflicting CAV exit.

Recall that the CAVs use RT-CVC for the longitudinal control. This control makes the CAV be as close as possible to the safety distance stated in (3.8) and also in (4.3). Simply speaking, the control aims to replace inequality with equality. According to the synchronization point, the CAV is brought to a given state when the precedent conflicting one frees the potential collision area. Among several states, there is one that results in the optimal control point. Hence, in the following, we study the optimal synchronization point.

4.4.1/ OPTIMAL SYNCHRONIZATION POINT

First, the optimal control point according to the simulated CAVs data is the following:

$$X_i^* = \begin{bmatrix} D - 16.93m \\ 6.21m/s \end{bmatrix}$$
(4.19)

This allows to free the conflict zone, after exactly 2.52s. This value includes $\tau = 0.5s$. In order to obtain these values, we only used the equation (4.4).

To reach the optimal control point, there are several possible approaches. However, recall that we aim to push the CAV to the maximum of the storage zone, using an optimal synchronization point. According to Cond1 and Cond2, there are two possible behaviors of CAVs:

- Cond1: Move freely then decelerate to reach the optimal control point (SC1).
- Cond2: Move freely, come to a complete stop at the optimal synchronization point then accelerate to meet the optimal control point (SC2).

For the first scenario (SC1), the synchronization point is obviously at the entrance of the potential collision area sl_1^* . In such a scenario, the CAV is in the limit of the Cond1. The other synchronization point is behind the optimal control point sl_2^* . Its position must allow the CAV when accelerating to reach the optimal control point. Its distance from the optimal control point can be easily obtained as follows:

$$sI_2^* = p_i^* - \frac{\left(v_i^*\right)^2}{2\overline{u_i}}$$
 (4.20)

With this in mind³, when the CAV comes to a complete stop, it must be able to reach the optimal control point according to the expected exit time of the precedent conflicting CAV. Hence, a new synchronization point sl_2^* is considered to follow the virtual leader *i*, instead of using u_sl at the stop line.

Contrasting with SC1, where CAV is at the border of Cond1, in SC2 the CAV still waits at the optimal synchronization point until the precedent CAV is approaching to the exit for launching the acceleration. With a given time of exit of the virtual leader *i*, the CAV *j* lunches the $\overline{u_i}$ acceleration at time *t*, if:

$$t^* \ge t_{exit,i} - \frac{v_j^*}{u_j} \tag{4.21}$$

CAV *j* keeps accelerating at the maximum while, it is able to come to a complete stop at

³In the simulation $sl_2^* = D - 26.58m$

the edge of the potential collision area.

4.4.2/ ACCELERATION COMPUTATION

Recall that the studied curve is only possible for CAVs that are able to come to a complete stop at sl_2^* before t^* . In order to allow all CAVs to meet the optimal control point, we consider the following rules for each CAV *j* that flows a virtual leader CAV *i*:

- while CAV *i* doesn't clear the potential collision, keep u_j^{sl^{*}1} as the upper bound of the acceleration.
- Is $u_i^{sl_2^*} \ge \underline{u_j}$
 - If yes : move freely,
 - Otherwise : compute a cruise control u_{opt} to reach the optimal control point at $t_{exit,i}$

To solve the problem of the computation of the cruise control we used the minimization of the integration of quadratic acceleration during the remained time [59]. The solution to the equation is an acceleration function of the form:

$$u_{opt}(t) = at + b \tag{4.22}$$

with *a* and *b* are constants to be found according to the initial conditions of speed and position. In order to avoid writing clutters, we invite the reader to get details about the values of *a* and *b* in [186], in pages 2410 and 2411. The solution either turns to SC2 (*b* negative), and all other intermediate solutions from SC2 to SC1. Otherwise, the solution is unfeasible (*b* positive and $v_i(t) = \overline{v_i}$).

In order to summarize the CAV's behavior into simple words, the CAV uses sl_2^* instead of the stop line. Once begin decelerating, it tries to follow the virtual leader to exit if possible at the optimal control point.

4.4.3/ PROCESS

In the studied intersection, one CAV *j* needs to adjust its speed with several other virtual CAVs ($i \in vl$). For safety reasons, we compute the acceleration as follows :

$$u_j^{vl}(t) = \min_{vl:\in Vl} \left(u_j^i(t) \right) \tag{4.23}$$

and

$$u_i^c(t) = \max\left(u_i^{sl_1^*}, \min\left(u_j^{vl}(t), u_j^{nl}(t)\right)\right)$$
(4.24)

As one can note, equations (4.23) and (4.24) are similar to the ones given in (3.9) and (3.11), respectively. The expected exit time is already computed for the FRO policy (3.18). Both FIFS and DCPVP keep the same rules.

4.5/ SIMULATION

In this section, we compare the introduction of the optimal control point to the following policies:

- FIFS
- DCPVP

In order to distinguish the policy with and without the optimal control point, we added OP for the policy where an optimal control point is added. the Other comparisons were conducted during the thesis. However, for obvious reasons, the optimal control point wasn't efficient. For instance, for FRO, since the CAV that is ready first goes, the conflicting follower CAVs just come to a complete stop between the (sl_2^*) and the stop line. So, the optimal control point is lost and the acceleration later is no more optimal. It is even, less interesting since it is further than the first simulated version of FRO.

The results of the OPDCPVP invites us to introduce an apporach inspired from the wellknown Particle Swarm Optimization (PSO) [189] for improving the results. So the DCPVP is extended by considering the weakness of both OPDCPVP and DCPVP, using mobile synchronization point.

4.5.1/ OPFIFS

The results of the optimal point included to FIFS are given in figure 4.9. In this figure, one can note that there are three sets of points. The first one is in the smooth part of the traffic. The second set is at the capacity of the intersection, with a wide range of speed, being near to the maximum speed. The last set is in the congested part of the traffic. The fitting curve shows that the average capacity of the intersection is near to 1upc/s.

In order to study the difference with the initial FIFS, figure 4.10 compares both fitting curves. From this figure, the reader can easily observe the improvement in terms of capacity and speed. Indeed, the fitting curve of OPFIFS is upper to the one of FIFS for all values of speed. Recall that initial the average capacity of FIFS is 0.81upc/s. The optimal control point improves the capacity by 23%. This capacity increasing makes OPFIFS more efficient than the well known DCP and the classical FRO. Recall that they result on an average capacity of 0.98upc/s and of 0.96upc/s, for DCP and FRO, respectively.



Figure 4.9: Comparison between the fitting curves of FIFS and OPFIFS



Figure 4.10: Simulation results of OPDCPVP: DCPVP with optimal control point

4.5.2/ OPDCPVP

The same rules of DCPVP are kept for the OPDCPVP. Simulations of OPDCPVP are presented in figure 4.11. One can observe that the distribution of the point around the fitting curve is different according to the state. When the traffic is smooth the points are homogeneously distributed around the fitting curve. The same observation can be done for the congested part. However, as the traffic becomes near to the capacity, there are fewer points. Mainly, the traffic performance is sensitive to destination of the simulated CAVs and their headway times. The intersection capacity of the OPDCPVP is averagely around 1.44upc/s.



Figure 4.11: Simulation results of OPFIFS: FIFS with optimal control point

Figure 4.12 compares OPDCPVP to DCPVP, through their fitting curves. One can clearly observe that OPDCPVP improves slightly the average speed of CAVs for the smooth traffic. However, the capacity of the DCPVP is lower. The gained speed is lost at the congested traffic. The main reason is that the opportunity for ranking negotiation is a little smaller than the one of DCPVP. Recall that, on the one hand, CAVs refuse to change their rank if they are accelerating. On the other hand, in OPDCPVP CAVs may begin the acceleration before sl_2^* which is located before the stop line.

From this comparison, we can derive the following observations:

4.5. SIMULATION

- The optimal control point is interesting when the traffic is low: There is no need of changing the ranks of CAVs.
- The optimal control becomes inefficient as the traffic demand is becoming high: CAVs need a better schedule.



Figure 4.12: Comparison of the fitting curves of the DCPVP and the OPDCPVP

4.5.3/ PSODCPVP

The last simulations show that the benefit of the optimal control point is limited to smooth traffic. At this stage of the CIM improvement, the dilemma between scheduling and cruise control is clearly stated. In order to illustrate the observed phenomena, let us consider the example given in figure 4.13.

Dilemma:Speed adjustment/Scheduling

let us consider a simple intersection with five CAVs: 3 in lane 1 and 2 in lane 2. In this example we compare two kinds of CIMs:

• Speed adjustment (CIM_{Sa}) for avoiding collision with the precedent conflicting CAV ranked according to FIFS, as showed in Figure 4.13(a).

• A sequence formation (CIM_{So}) with green and red signalization, as showed in Figure 4.13(b).

Each arc between two CAVs in the Figure 4.13 shows which CAV goes first and which one crosses the intersection after. For safety reasons, when a CAV is behind another one that moves either on the same lane or on a conflicting lane, there is a headway time. Let h_f (blue arc in Figure 4.13) and h_c (amber arc in Figure 4.13) be the headway times between two successive CAVs that move respectively on the same lane and on conflicting lanes (see Figure 4.2). We consider that there are two kinds of h_c : h_{cSa} and h_{cSo} . The first one results from speed adjustment (CIM_{Sa}) and the second one from the red color (CIM_{So}). We have the following relation:

$$h_f \le h_{cSa} \le h_{cSo} \tag{4.25}$$

Without loss of generality, let us consider that $h_f = 2s$, $h_{cSa} = 3s$ and $h_{cSo} = 6s$. If we focus on the exit time of the five present CAVs, both CIMs obtains the same result which is 12s. But, by removing the last vehicle, CIM_{Sa} is more efficient whereas if a sixth CAV arrives at Lane 2, then CIM_{So} is more efficient.



Figure 4.13: Example: (a) CIM based on speed adjustment and FCFS, (b) CIM based on sequence optimization.

A similar problem is raised by the optimal control point in OPDCPVP. Because they begin acceleration for exiting at the optimal headway time, CAVs refuse the negotiation. In the following, we present the solution tested in this thesis.

Mobile synchronization point

In order to solve the problem we suggested a mobile synchronization point. Each CAV *i* generates its own sl_i randomly, with $56m \le sl_i \le 80m$. The lower bound of sl_i allows the CAV to reach its maximum speed when crossing the conflict zone, whereas the upper

bound is the stop line. The mobile synchronization point needs to be adapted to the traffic state. If the traffic demand is low and there is no opportunistic change of the rank, theoretically, the lower bound is a good solution. Otherwise, when the traffic demand is high, it is better to form groups of CAVs.

In order to make sl_i fits the traffic state, we used a distributed approach inspired from the well known PSO. From the OPL, the CAV negotiates its rank with the CAVs in vl. According to its current sl_i , it considers only CAVs that are behind sl_i to negotiate the rank. It compares the completion time of the new rank $C_{max}^{sl_i}$ to the completion of the precedent one C_{max}^{old} , considering only CAVs in vl. The fitness of the sl_i is computed as follows.

$$f(sl_i) = C_{max}^{old} - C_{max}^{sl_i}$$
(4.26)

If the result of the fitness of the current sl_i is positive, it is kept as a local optimum $\underline{sl_i}$. Whatever the result, after the negotiation, the CAV sends the computed $f(sl_i)$ and sl_i to the intersection manager. Among the all received fitness values during the step time k, it keeps the best value:

$$sl^*(k) = argmax_i\left(f(sl_i(k))\right). \tag{4.27}$$

In other words, it keeps the sl_i^* that gives the maximum gain of the completion time. The intersection manager computes the $sl_{IM}(k)$ that it will send later to the CAVs as follows:

$$sl_I M(k) = \alpha sl^*(k) + (1 - \alpha) sl_{IM}(k - 1)$$
 (4.28)

with $\alpha = 0.001$ in the simulation ⁴. A CAV *i* computes the new value of sl_i for the step k + 1 using the well-known particle's position evolution of the PSO:

$$sl_i(k+1) = sl_i(k) + \mathcal{V}_i(k),$$
 (4.29)

with $\mathcal{V}_i(k)$ is the particle velocity at the step *k*. $\mathcal{V}_i(1)$ is generated randomly when the CAV *i* is created. $-0.5 \leq \mathcal{V}_i(1) \leq 0.5$. For all other simulation steps, $\mathcal{V}_i(k)$ is randomly computed as follows:

$$\mathcal{V}_{i}(k) = \beta \mathcal{V}(k-1) + \phi \left(sl_{i}^{*} - sl_{i}(k) \right) + \psi \left(sl_{IM}(k) - sl_{i}(k) \right),$$
(4.30)

with β , ϕ and ψ are empirically set equal to $\frac{1}{3}$.

Simulation

The simulation results of the proposed distributed PSO is presented in figure 4.14. One can note from the figure that the majority of points are in the smooth part of the traffic. The points on the congested part of the traffic do not have an average speed less than

⁴The value of α is computed empirically after several simulation runs



8km/h. The average capacity of the intersection is close to 1.53 pcu/s.



The PSODCPVP is compared to both DCPVP and OPDCPVP, in figure 4.15. There is a small gain in terms of capacity (3% compared to DCPVP and 6% compared to OPDCP). However, even with this small gain, the PSODCPVP shows that there is a potential gain to deal with the issue of a mobile adaptive synchronization point.

4.5.4/ CONCLUSION

This chapter proposes an approach to address the crux problem of the safety constraint between two intersecting movements in cooperative intersection management. By providing a safe optimal control point, both trajectory and sequence are optimized in an elementary intersection. Optimal control theory and shortest path algorithm are used for providing smooth traffic.

The optimal control point is used for a more general intersection. With FIFS, the optimal control point allows a significant improvement of the intersection speed and throughput. However, the optimal control point doesn't suit other policies. With DCPVP, the optimal control point only slightly improves the speed of the intersection, when the traffic demand is low. In order to overcome this drawback, we introduced a mobile synchronization point based on distributed PSO. This improves both intersection throughput and speed. However, the improvement is not as significant as the one achieved in the previous chapter.



Figure 4.15: Comparison of the fitting curves of the tree approaches of the DCPVP

Ш

CONCLUSION

5

GENERAL CONCLUSION AND PROSPECTS

Cooperative Intersection Management uses the communication ability of vehicles to obtain a right of way. The right of way can be a simple semaphore green or red. In this case, many approaches are proposed to decide the best set of vehicles that are allowed to cross simultaneously the intersection. One of the most efficient ways to improve the intersection efficiency is to allow vehicles to form groups that cross simultaneously the intersection. These groups are formed by the Distributed Clearing Policy. More complicated right of way relies on cooperative cruise control. Vehicles adjust their speed to avoid collision with all other conflicting vehicles. There are mainly two cooperative cruise control protocols in the literature. The first one is based on the reservation. The vehicles reserve the time they occupy the potential zone of collision. However, this protocol raises problems of safety and feasibility. Another noticeable protocol relies on the ordered presence list, on which the rank determines the precedent vehicles that should be considered to avoid the collision. This aims to form a virtual platoon, where conflicting vehicles are considered as they are moving in front of the ego-vehicle on the same lane. However, the main used scheduling approach for this protocol is the well-known First In First Served policy.

This virtual platoon protocol needs to be extended by introducing new scheduling policies. In this thesis, we introduced the First Ready Out policy and the Distributed Clearing Policy. To this end, we defined the synchronization point and negotiation rules. A simulator was built to show the vehicle interaction and to evaluate the performance. Comparisons with the first in first served show a huge gain in terms of average speed and intersection capacity when using the well-known distributed clearing policy.

The virtual platoon was also extended, using the expected exit time of vehicles. This allows the computation of an optimal control point that needs to be respected by the vehicle in order to minimize the headway. This point allows vehicles to free the soonest potential zone of collision. The optimal control point improves significantly the first in first served policy. However, when it comes to other scheduling policies, the improvement is questionable. One possible reason is that the new synchronization point prevents vehicles to negotiate their rank near the stop line. To overcome this limitation, a mobile synchronization point between conflicting vehicles is added. The synchronization point is adjusted to the traffic state using distributed particle swarm optimization. Compared to the former distributed clearing policy, the approach improves slightly both the speed and throughput of the virtual platoon.

Two interesting perspectives can be drawn from this work:

- The treated system raises the issue of the control of hybrid multi-agent systems, with hard real-time constraints. Our contribution showed us the complexity of the raised problem, where both continuous control and discrete scheduling need to be simultaneously optimized to improve the performance of the system. Similar problems are raised in the transportation system, such as vehicle's overtaking, road merging, drone interaction... One of the possible research topics is to find a methodological approach that treats these problems. In our thesis, each problem (discrete, continuous) is optimized separately, and then approximate methods are used to take advantage of each optimization. This may contribute to developing new distributed heuristics adapted for the interaction of mobile entities.
- The used approach is based on the patterns of optimal solutions of scheduling (group formation) and cruise control (optimal control point). It is interesting to use other approaches for controlling the intersection, mainly the ones based on machine learning techniques. For instance, deep reinforcement learning can be used to train the vehicles to have a better behavior according to the state of the other conflicting vehicles. Multiagent deep reinforcement learning is a very active research topic [190]. Intersection management is a good application, in order to effectively contribute to this topic.

BIBLIOGRAPHY

- [1] FG Tyack. Street traffic signals, with particular reference to vehicle actuation. *Journal of the Institution of Electrical Engineers*, 82(494):125–154, 1938.
- [2] KL Bng and LE Nilsson. Optimal control of isolated traffic signals. *IFAC Proceedings Volumes*, 9(4):173–184, 1976.
- [3] E.A. Sofronova and A.I. Diveev. Traffic flows optimal control problem with full information*. In 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), pages 1–6, 2020.
- [4] Girija H Kulkarni and Poorva G Waingankar. Fuzzy logic based traffic light controller. In 2007 International Conference on Industrial and Information Systems, pages 107–110. IEEE, 2007.
- [5] Seyed Sajad Mousavi, Michael Schukat, and Enda Howley. Traffic light control using deep policy-gradient and value-function-based reinforcement learning. *IET Intelligent Transport Systems*, 11(7):417–423, 2017.
- [6] Yaying Zhang, Wei Qiang, and Zhengyu Yang. A new traffic signal control method based on hybrid colored petri net in isolated intersections. *International Journal of Intelligent Transportation Systems Research*, 15(2):98–107, 2017.
- [7] Fo Vo Webster. Traffic signal settings. *Road Research Lab Tech Papers /UK*/, (39), 1958.
- [8] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Yibing Wang. Review of road traffic control strategies. *Proceedings of the IEEE*, 91(12):2043–2067, 2003.
- [9] Alan J Miller. A computer control system for traffic networks. 1963.
- [10] John DC Little. The synchronization of traffic signals by mixed-integer linear programming. *Operations Research*, 14(4):568–594, 1966.
- [11] Jean-Jacques Henry, Jean Loup Farges, and Jean Tuffal. The prodyn real time traffic algorithm. In *Control in Transportation Systems*, pages 305–310. Elsevier, 1984.

- [12] Nathan H Gartner. OPAC: A demand-responsive strategy for traffic signal control. Number 906. 1983.
- [13] Florence Boillot, JM Blosseville, JB Lesort, V Motyka, M Papageorgiou, and S Sellam. Optimal signal control of urban traffic networks. In *Road Traffic Monitoring*, 1992 (IEE Conf. Pub. 355), page 75. IET, 1992.
- [14] Suvrajeet Sen and K Larry Head. Controlled optimization of phases at an intersection. *Transportation science*, 31(1):5–17, 1997.
- [15] Maxime Tréca, Julian Garbiso, Dominique Barth, and Mahdi Zargayouna. Fast bootstrapping for reinforcement learning-based traffic signal control systems using queueing theory. In 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall), pages 1–5, 2020.
- [16] Denos C Gazis. *Traffic theory*, volume 50. Springer Science & Business Media, 2006.
- [17] Aleksandar Stevanovic, Cameron Kergaye, and Peter T Martin. Scoot and scats: A closer look into their operations. In 88th Annual Meeting of the Transportation Research Board. Washington DC, 2009.
- [18] Deepeka Garg, Maria Chli, and George Vogiatzis. A deep reinforcement learning agent for traffic intersection control optimization. In 2019 ieee intelligent transportation systems conference (itsc), pages 4222–4229. IEEE, 2019.
- [19] Ali Louati, Hassen Louati, and Zhaojian Li. Deep learning and case-based reasoning for predictive and adaptive traffic emergency management. *The Journal of Supercomputing*, 77(5):4389–4418, 2021.
- [20] Yi Zhang. An adaptive pre-signal setting to provide bus priority under a coordinated traffic-responsive network. In 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), pages 1–6. IEEE, 2020.
- [21] Cecilia Pasquale, Simona Sacone, Silvia Siri, and Antonella Ferrara. Traffic control for freeway networks with sustainability-related objectives: Review and future challenges. *Annual Reviews in Control*, 48:312–324, 2019.
- [22] Mouna Karoui, Gerard Chalhoub, and Antonio Freitas. An efficient path planning glosa-based approach over large scale and realistic traffic scenario. *Internet Technology Letters*, page e194, 2020.
- [23] Abdeljalil Abbas-Turki, Florent Perronnet, Jocelyn Buisson, Abdellah El-Moudni, Mourad Ahmane, and Renan Zéo. Cooperative intersections for emerging mobility systems. In 15th meeting of the Euro Working Group on transportation, 2012.

- [24] Cristofer Englund, Lei Chen, Alexey Vinel, and Shih Yang Lin. Future applications of vanets. In *Vehicular ad hoc Networks*, pages 525–544. Springer, 2015.
- [25] JD Chovan, L Tijerina, JH Everson, JA Pierowicz, and DL Hendricks. Examination of intersection, left turn across path crashes and potential ivhs countermeasures. Technical report, 1994.
- [26] Max Bareiss, John Scanlon, Rini Sherony, and Hampton C Gabler. Crash and injury prevention estimates for intersection driver assistance systems in left turn across path/opposite direction crashes in the united states. *Traffic injury prevention*, 20(sup1):S133–S138, 2019.
- [27] Wenrui Qu, Qiao Sun, Qun Zhao, Tao Tao, and Yi Qi. Statistical analysis of safety performance of displaced left-turn intersections: case studies in san marcos, texas. *International journal of environmental research and public health*, 17(18):6446, 2020.
- [28] Hari Krishna Gaddam and K Ramachandra Rao. Speed-density functional relationship for heterogeneous traffic data: a statistical and theoretical investigation. *Journal of modern transportation*, 27(1):61–74, 2019.
- [29] Chang-qiao Shao, Jian Rong, and Xiao-ming Liu. Study on the saturation flow rate and its influence factors at signalized intersections in china. *Procedia-Social and Behavioral Sciences*, 16:504–514, 2011.
- [30] Anna Granà, Tullio Giuffrè, Elżbieta Macioszek, and Francesco Acuto. Estimation of passenger car equivalents for two-lane and turbo roundabouts using aimsun. *Frontiers in Built Environment*, 6:86, 2020.
- [31] Satyajit Mondal and Ankit Gupta. Non-linear evaluation model to analyze saturation flow under weak-lane-disciplined mixed traffic stream. *Transportation Research Record*, page 0361198121998370, 2021.
- [32] Zhen Yang, Yiheng Feng, and Henry X Liu. A cooperative driving framework for urban arterials in mixed traffic conditions. *Transportation research part C: emerging technologies*, 124:102918, 2021.
- [33] Lei Chen and Cristofer Englund. Cooperative intersection management: A survey. *IEEE transactions on intelligent transportation systems*, 17(2):570–586, 2015.
- [34] Shunsuke Aoki and Ragunathan Rajkumar. V2v-based synchronous intersection protocols for mixed traffic of human-driven and self-driving vehicles. In 2019 IEEE 25th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), pages 1–11. IEEE, 2019.

- [35] Elnaz Namazi, Jingyue Li, and Chaoru Lu. Intelligent intersection management systems considering autonomous vehicles: A systematic literature review. *IEEE Access*, 7:91946–91965, 2019.
- [36] Jia Wu, Abdeljalil Abbas-Turki, and Abdellah El Moudni. Cooperative driving: an ant colony system for autonomous intersection management. *Applied Intelligence*, 37(2):207–222, 2012.
- [37] Matthias Grünewald, Carsten Rust, and Ulf Witkowski. Using mini robots for prototyping intersection management of vehicles. In *Proceedings of the 3rd international symposium on autonomous minirobots for research and edutainment (AMiRE 2005)*, pages 287–292. Springer, 2006.
- [38] Arnaud de La Fortelle. Analysis of reservation algorithms for cooperative planning at intersections. In *13th International IEEE conference on intelligent transportation systems*, pages 445–449. IEEE, 2010.
- [39] Michael Quinlan, Tsz-Chiu Au, Jesse Zhu, Nicolae Stiurca, and Peter Stone. Bringing simulation to life: A mixed reality autonomous intersection. In 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 6083–6088. IEEE, 2010.
- [40] Chien-Liang Fok, Maykel Hanna, Seth Gee, Tsz-Chiu Au, Peter Stone, Christine Julien, and Sriram Vishwanath. A platform for evaluating autonomous intersection management policies. In 2012 IEEE/ACM Third International Conference on Cyber-Physical Systems, pages 87–96, 2012.
- [41] Florent Perronnet, Abdeljalil Abbas-Turki, Jocelyn Buisson, Abdellah El Moudni, Renan Zéo, 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, 2012.
- [42] Florent Perronnet, Abdeljalil Abbas-Turki, and Abdellah El Moudni. A sequencedbased protocol to manage autonomous vehicles at isolated intersections. In 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), pages 1811–1816, 2013.
- [43] A Lombard, F Perronnet, A Abbas-Turki, A El Moudni, and R Bouyekhf. V2x for vehicle speed synchronization at intersections. In *Proc. 22nd Intelligent Transportation Systems World Congr.*, 2015.
- [44] Mohammad Khayatian, Mohammadreza Mehrabian, and Aviral Shrivastava. Rim: Robust intersection management for connected autonomous vehicles. In 2018 IEEE Real-Time Systems Symposium (RTSS), pages 35–44, 2018.

- [45] Mohammad Khayatian, Mohammadreza Mehrabian, Harshith Allamsetti, Kai-Wei Liu, Po-Yu Huang, Chung-Wei Lin, and Aviral Shrivastava. Cooperative driving of connected autonomous vehicles using responsibility-sensitive safety (rss) rules. In Proceedings of the ACM/IEEE 12th International Conference on Cyber-Physical Systems, pages 11–20, 2021.
- [46] R Hult, M Zanon, S Gros, and P Falcone. Optimal coordination of three cars approaching an intersection. *Online: https://youtu. be/nYSXvnaNRK4*, 2017.
- [47] Robert Hult, Mario Zanon, Gianluca Frison, Sébastien Gros, and Paolo Falcone. Experimental validation of a semi-distributed sequential quadratic programming method for optimal coordination of automated vehicles at intersections. *Optimal Control Applications and Methods*, 41(4):1068–1096, 2020.
- [48] Ismail H Zohdy and Hesham Rakha. Game theory algorithm for intersection-based cooperative adaptive cruise control (cacc) systems. In 2012 15th International IEEE Conference on Intelligent Transportation Systems, pages 1097–1102. IEEE, 2012.
- [49] R Naumann, R Rasche, Jürgen Tacken, and C Tahedi. Validation and simulation of a decentralized intersection collision avoidance algorithm. In *Proceedings of Conference on Intelligent Transportation Systems*, pages 818–823. IEEE, 1997.
- [50] Michel Ferreira and Pedro M d'Orey. On the impact of virtual traffic lights on carbon emissions mitigation. IEEE Transactions on Intelligent Transportation Systems, 13(1):284–295, 2011.
- [51] Florent Perronnet, Abdeljalil Abbas-Turki, and Abdellah El Moudni. A sequencedbased protocol to manage autonomous vehicles at isolated intersections. In 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), pages 1811–1816. IEEE, 2013.
- [52] Ozan K Tonguz. Red light, green light—no light: Tomorrow's communicative cars could take turns at intersections. *IEEE Spectrum*, 55(10):24–29, 2018.
- [53] Cristina Olaverri-Monreal, Pedro Gomes, Michelle Krüger Silvéria, and Michel Ferreira. In-vehicle virtual traffic lights: a graphical user interface. In *7th Iberian Conference on Information Systems and Technologies (CISTI 2012)*, pages 1–6. IEEE, 2012.
- [54] Mourad Ahmane, Abdeljalil Abbas-Turki, Florent Perronnet, Jia Wu, Abdellah El Moudni, Jocelyn Buisson, and Renan Zeo. Modeling and controlling an isolated urban intersection based on cooperative vehicles. *Transportation Research Part C: Emerging Technologies*, 28:44–62, 2013.

- [55] Rusheng Zhang, Frank Schmutz, Kyle Gerard, Aurélicn Pomini, Louis Basseto, Sami Ben Hassen, Akihiro Ishikawa, Inci Ozgunes, and Ozan Tonguz. Virtual traffic lights: System design and implementation. In 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall), pages 1–5. IEEE, 2018.
- [56] Kurt Dresner and Peter Stone. Multiagent traffic management: A reservation-based intersection control mechanism. In *Autonomous Agents and Multiagent Systems, International Joint Conference on*, volume 3, pages 530–537. IEEE Computer Society, 2004.
- [57] Tsz-Chiu Au and Peter Stone. Motion planning algorithms for autonomous intersection management. In *Workshops at the Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.
- [58] Yuanyuan Wu, Haipeng Chen, and Feng Zhu. Dcl-aim: Decentralized coordination learning of autonomous intersection management for connected and automated vehicles. *Transportation Research Part C: Emerging Technologies*, 103:246–260, 2019.
- [59] Andreas A Malikopoulos and Liuhui Zhao. Optimal path planning for connected and automated vehicles at urban intersections. In 2019 IEEE 58th Conference on Decision and Control (CDC), pages 1261–1266. IEEE, 2019.
- [60] Zhihong Yao, Haoran Jiang, Yang Cheng, Yangsheng Jiang, and Bin Ran. Integrated schedule and trajectory optimization for connected automated vehicles in a conflict zone. *IEEE Transactions on Intelligent Transportation Systems*, 2020.
- [61] Gabriel R Campos, Paolo Falcone, and Jonas Sjöberg. Traffic safety at intersections: a priority based approach for cooperative collision avoidance. In *FASTzero'15 Proceedings*, pages 9–15, Gothenburg, 2015. Chalmers University Technology, Gothenburg, Sweden.
- [62] Alexander Katriniok, Peter Kleibaum, and Martina Joševski. Distributed model predictive control for intersection automation using a parallelized optimization approach. *IFAC PapersOnLine*, 50(1):5940–5946, 2017.
- [63] Amir Mirheli, Mehrdad Tajalli, Leila Hajibabai, and Ali Hajbabaie. A consensusbased distributed trajectory control in a signal-free intersection. *Transportation Re*search Part C: Emerging Technologies, 100:161–176, 2019.
- [64] Maximilian Kloock, Patrick Scheffe, Sascha Marquardt, Janis Maczijewski, Bassam Alrifaee, and Stefan Kowalewski. Distributed model predictive intersection control of multiple vehicles. In 2019 IEEE Intelligent Transportation Systems Conference (ITSC), pages 1735–1740, 2019.

- [65] Zhixia Li, Madhav V Chitturi, Lang Yu, Andrea R Bill, and David A Noyce. Sustainability effects of next-generation intersection control for autonomous vehicles. *Transport*, 30(3):342–352, 2015.
- [66] Xiangyu Meng and Christos G Cassandras. Optimal control of autonomous vehicles for non-stop signalized intersection crossing. In 2018 IEEE Conference on Decision and Control (CDC), pages 6988–6993. IEEE, 2018.
- [67] S Alireza Fayazi and Ardalan Vahidi. Vehicle-in-the-loop (vil) verification of a smart city intersection control scheme for autonomous vehicles. In 2017 IEEE Conference on Control Technology and Applications (CCTA), pages 1575–1580. IEEE, 2017.
- [68] Mohammad Khayatian, Yingyan Lou, Mohammadreza Mehrabian, and Aviral Shirvastava. Crossroads+ a time-aware approach for intersection management of connected autonomous vehicles. ACM Transactions on Cyber-Physical Systems, 4(2):1–28, 2019.
- [69] Jia Wu, Florent Perronnet, and Abdeljalil Abbas-Turki. Cooperative vehicle-actuator system: a sequence-based framework of cooperative intersections management. IET Intelligent Transport Systems, 8(4):352–360, 2014.
- [70] Alejandro Ivan Morales Medina, Nathan Van de Wouw, and Henk Nijmeijer. Automation of a t-intersection using virtual platoons of cooperative autonomous vehicles. In 2015 IEEE 18th International Conference on Intelligent Transportation Systems, pages 1696–1701. IEEE, 2015.
- [71] Wendan Du, Abdeljalil Abbas-Turki, Abderrafiaa Koukam, Stéphane Galland, and Franck Gechter. On the v2x speed synchronization at intersections: Rule based system for extended virtual platooning. *Procedia Computer Science*, 141:255–262, 2018.
- [72] Gentry Lee. A generalization of linear car-following theory. *Operations research*, 14(4):595–606, 1966.
- [73] PG Gipps. Computer program multim for simulating output from vehicle detectors on a multi-lane signal-controlled road. Technical report, 1976.
- [74] Arne Kesting, Martin Treiber, and Dirk Helbing. Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 368(1928):4585–4605, 2010.
- [75] Florent Perronnet. *Régulation coopérative des intersections: protocoles et politiques.* PhD thesis, Université de Technologie de Belfort-Montbeliard, 2015.
BIBLIOGRAPHY

- [76] Xuguang Hao. *Contribution à l'intersection coopérative: commandes longitudinale et latérale*. PhD thesis, Bourgogne Franche-Comté, 2017.
- [77] Fatma Outay, Stéphane Galland, Nicolas Gaud, and Abdeljalil Abbas-Turki. Simulation of connected driving in hazardous weather conditions: general and extensible multiagent architecture and models. *Engineering applications of artificial intelligence*, 104:104412, 2021.
- [78] Alejandro Ivan Morales Medina, Nathan van de Wouw, and Henk Nijmeijer. Cooperative intersection control based on virtual platooning. *IEEE Transactions on Intelligent Transportation Systems*, 19(6):1727–1740, 2017.
- [79] Biao Xu, Shengbo Eben Li, Yougang Bian, Shen Li, Xuegang Jeff Ban, Jianqiang Wang, and Keqiang Li. Distributed conflict-free cooperation for multiple connected vehicles at unsignalized intersections. *Transportation Research Part C: Emerging Technologies*, 93:322–334, 2018.
- [80] Yougang Bian, Shengbo Eben Li, Wei Ren, Jianqiang Wang, Keqiang Li, and Henry X Liu. Cooperation of multiple connected vehicles at unsignalized intersections: Distributed observation, optimization, and control. *IEEE Transactions on Industrial Electronics*, 67(12):10744–10754, 2019.
- [81] Carlos F Daganzo and Nikolas Geroliminis. An analytical approximation for the macroscopic fundamental diagram of urban traffic. *Transportation Research Part B: Methodological*, 42(9):771–781, 2008.
- [82] Huile Xu, Yi Zhang, Christos G Cassandras, Li Li, and Shuo Feng. A bi-level cooperative driving strategy allowing lane changes. *Transportation research part C: emerging technologies*, 120:102773, 2020.
- [83] Fatemeh Baratian-Ghorghi, Huaguo Zhou, and Isaac Wasilefsky. Impacts of red light photo enforcement cameras on clearance lost time at signalized intersections. In *Transportation Research Board 94th Annual Meeting*, number 15-2245, 2015.
- [84] Thomas Urbanik, Alison Tanaka, Bailey Lozner, Eric Lindstrom, Kevin Lee, Shaun Quayle, Scott Beaird, Shing Tsoi, Paul Ryus, Doug Gettman, et al. Signal timing manual, volume 1. Transportation Research Board Washington, DC, 2015.
- [85] Jia Wu, Abdeljalil Abbas-Turki, Aurelien Correia, and Abdellah El Moudni. Discrete intersection signal control. In *2007 IEEE international conference on service operations and logistics, and informatics*, pages 1–6. IEEE, 2007.
- [86] Byungkyu Park, Jongsun Won, and Ilsoo Yun. Application of microscopic simulation model calibration and validation procedure: Case study of coordinated actuated signal system. *Transportation Research Record*, 1978(1):113–122, 2006.

- [87] Douglas Brent West et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.
- [88] Edsger W Dijkstra et al. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.
- [89] Donald B Johnson. A note on dijkstra's shortest path algorithm. *Journal of the ACM* (*JACM*), 20(3):385–388, 1973.
- [90] Joao Carlos Namorado Climaco and Ernesto Queiros Vieira Martins. A bicriterion shortest path algorithm. *European Journal of Operational Research*, 11(4):399– 404, 1982.
- [91] Pierre A Humblet. Another adaptive distributed shortest path algorithm. *IEEE transactions on communications*, 39(6):995–1003, 1991.
- [92] Jia Wu, Abdeljalil Abbas-Turki, and Abdellah El Moudni. Régulation du trafic aux intersections: Prise en compte des véhicules prioritaires. *Revue e-STA, France*, 6, 2009.
- [93] Jia Wu, Abdeljalil Abbas-Turki, and Abdellah EL Moudni. Intersection traffic control by a novel scheduling model. In 2009 IEEE/INFORMS International Conference on Service Operations, Logistics and Informatics, pages 329–334. IEEE, 2009.
- [94] Fei Yan, Mahjoub Dridi, and Abdellah El Moudni. Autonomous vehicle sequencing algorithm at isolated intersections. In 2009 12th International IEEE Conference on Intelligent Transportation Systems, pages 1–6, 2009.
- [95] Bin Mohamad Nor Mohamad Hafizulazwan. Optimal scheduling of connected and automated vehicles at urban intersections via milp. In *61*, pages 160–165., 2018.
- [96] Nick McKeown. The islip scheduling algorithm for input-queued switches. *IEEE/ACM transactions on networking*, 7(2):188–201, 1999.
- [97] Tianyu Wang. *Parallel machine scheduling with precedence constraints*. PhD thesis, École centrale de Nantes, 2018.
- [98] Houssine Chetto and Maryline Chetto. Some results of the earliest deadline scheduling algorithm. *IEEE Transactions on software engineering*, 15(10):1261, 1989.
- [99] Beno¹t Chachuat. Mixed-integer linear programming (milp): Model formulation. *Mc-Master University Department of Chemical Engineering. Accessed July*, 17, 2019.
- [100] Saeid Soleimaniamiri and Xiaopeng Li. Scheduling of heterogeneous connected automated vehicles at a general conflict area. *Transp. Res. Board 98th Annu. MeetingTransportation Res. Board*, 2019.

- [101] Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, Carnegie-Mellon Univ Pittsburgh Pa Management Sciences Research Group, 1976.
- [102] Marco Dorigo and Luca Maria Gambardella. Ant colonies for the travelling salesman problem. *biosystems*, 43(2):73–81, 1997.
- [103] Tri-Hai Nguyen and Jason J Jung. Ant colony optimization-based traffic routing with intersection negotiation for connected vehicles. *Applied Soft Computing*, 112:107828, 2021.
- [104] Luis Cruz-Piris, Miguel A Lopez-Carmona, and Ivan Marsa-Maestre. Automated optimization of intersections using a genetic algorithm. *IEEE Access*, 7:15452–15468, 2019.
- [105] Jinjian Li, Mahjoub Dridi, and Abdellah El-Moudni. A cooperative traffic control for the vehicles in the intersection based on the genetic algorithm. In 2016 4th IEEE International Colloquium on Information Science and Technology (CiSt), pages 627– 632. IEEE, 2016.
- [106] Qiang Lu and Kyoung-Dae Kim. A genetic algorithm approach for expedited crossing of emergency vehicles in connected and autonomous intersection traffic. *Journal of Advanced Transportation*, 2017, 2017.
- [107] Fei Yan, Mahjoub Dridi, and Abdellah El Moudni. Autonomous vehicle sequencing problem for a multi-intersection network: A genetic algorithm approach. In 2013 International Conference on Advanced Logistics and Transport, pages 215–220. IEEE, 2013.
- [108] Laurence A Wolsey and George L Nemhauser. *Integer and combinatorial optimization*, volume 55. John Wiley & Sons, 1999.
- [109] Chung Laung Liu. Elements of discrete mathematics. 1985.
- [110] Tsz-Chiu Au, Neda Shahidi, and Peter Stone. Enforcing liveness in autonomous traffic management. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 25, 2011.
- [111] Jean Gregoire, Silvère Bonnabel, and Arnaud de La Fortelle. Priority-based coordination of robots. 2014.
- [112] Arnaud de La Fortelle. Analysis of reservation algorithms for cooperative planning at intersections. In *13th International IEEE Conference on Intelligent Transportation Systems*, pages 445–449, 2010.

- [113] Alexandre Lombard, Florent Perronnet, Abdeljalil Abbas-Turki, and Abdellah El Moudni. Decentralized management of intersections of automated guided vehicles. *IFAC-PapersOnLine*, 49(12):497–502, 2016.
- [114] Joseph Douglas Horton. A polynomial-time algorithm to find the shortest cycle basis of a graph. *SIAM Journal on Computing*, 16(2):358–366, 1987.
- [115] Kailong Zhang, Dafang Zhang, Arnaud de La Fortelle, Xiao Wu, and Jean Gregoire. State-driven priority scheduling mechanisms for driverless vehicles approaching intersections. *IEEE Transactions on Intelligent Transportation Systems*, 16(5):2487– 2500, 2015.
- [116] Dustin Carlino, Stephen D. Boyles, and Peter Stone. Auction-based autonomous intersection management. In 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), pages 529–534, 2013.
- [117] Michael W Levin and Stephen D Boyles. Intersection auctions and reservationbased control in dynamic traffic assignment. *Transportation Research Record*, 2497(1):35–44, 2015.
- [118] Aurélien Corréïa. *Modélisation de conflits dans l'algèbre des dioïdes: application à la régulation de trafic dans les carrefours*. PhD thesis, Besançon, 2007.
- [119] Jia Wu, Fei Yan, and Jianxing Liu. Effectiveness proving and control of platoonbased vehicular cyber-physical systems. *IEEE Access*, 6:21140–21151, 2018.
- [120] Rahmi Akcelik. Traffic signals: capacity and timing analysis. 1981.
- [121] Nagui M Rouphail, Mohammad Anwar, Daniel B Fambro, Paul Sloup, and Cesar E Perez. Validation of generalized delay model for vehicle-actuated traffic signals. *Transportation research record*, 1572(1):105–111, 1997.
- [122] Florent Perronnet, Jocelyn Buisson, Alexandre Lombard, Abdeljalil Abbas-Turki, Mourad Ahmane, and Abdellah El Moudni. Deadlock prevention of self-driving vehicles in a network of intersections. *IEEE Transactions on Intelligent Transportation Systems*, 20(11):4219–4233, 2019.
- [123] Cortés J Tallapragada P. Coordinated intersection traffic management. IFAC-PapersOnLine, 48(22):233–239, 2015.
- [124] Shuo Feng, Ziyou Song, Zhaojian Li, Yi Zhang, and Li Li. Robust platoon control in mixed traffic flow based on tube model predictive control. *IEEE Transactions on Intelligent Vehicles*, pages 1–1, 2021.

- [125] Huile Xu, Shuo Feng, Yi Zhang, and Li Li. A grouping-based cooperative driving strategy for cavs merging problems. *IEEE Transactions on Vehicular Technology*, 68(6):6125–6136, 2019.
- [126] A M Ishtiaque Mahbub and Andreas A. Malikopoulos. A platoon formation framework in a mixed traffic environment. *IEEE Control Systems Letters*, 6:1370–1375, 2022.
- [127] Sharmila Devi Kumaravel, Andreas A. Malikopoulos, and Ramakalyan Ayyagari. Decentralized cooperative merging of platoons of connected and automated vehicles at highway on-ramps. In 2021 American Control Conference (ACC), pages 2055–2060, 2021.
- [128] Ismail H Zohdy, Raj Kishore Kamalanathsharma, and Hesham Rakha. Intersection management for autonomous vehicles using icacc. In *2012 15th international IEEE conference on intelligent transportation systems*, pages 1109–1114. IEEE, 2012.
- [129] Ismail H Zohdy and Hesham A Rakha. Intersection management via vehicle connectivity: The intersection cooperative adaptive cruise control system concept. *Journal of Intelligent Transportation Systems*, 20(1):17–32, 2016.
- [130] Hesham A Rakha, Ismail Zohdy, Raj K Kamalanathsharma, et al. Agent-based game theory modeling for driverless vehicles at intersections. 2013.
- [131] Martin J Osborne et al. *An introduction to game theory*, volume 3. Oxford university press New York, 2004.
- [132] Zsolt Szalay, Zoltán Hamar, and Peter Simon. A multi-layer autonomous vehicle and simulation validation ecosystem axis: Zalazone. In *International Conference on Intelligent Autonomous Systems*, pages 954–963. Springer, 2018.
- [133] Shuja Ansari, Jawad Ahmad, Syed Aziz Shah, Ali Kashif Bashir, Tuleen Boutaleb, and Sinan Sinanovic. Chaos-based privacy preserving vehicle safety protocol for 5g connected autonomous vehicle networks. *Transactions on Emerging Telecommunications Technologies*, 31(5):e3966, 2020.
- [134] Rafael Molina-Masegosa and Javier Gozalvez. Lte-v for sidelink 5g v2x vehicular communications: A new 5g technology for short-range vehicle-to-everything communications. *IEEE Vehicular Technology Magazine*, 12(4):30–39, 2017.
- [135] Adrian Abunei, Ciprian-Romeo Comşa, and Ion Bogdan. Implementation of etsi its-g5 based inter-vehicle communication embedded system. In 2017 International Symposium on Signals, Circuits and Systems (ISSCS), pages 1–4. IEEE, 2017.

- [136] Ioannis Mavromatis, Andrea Tassi, and Robert J Piechocki. Operating its-g5 dsrc over unlicensed bands: A city-scale performance evaluation. In 2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), pages 1–7. IEEE, 2019.
- [137] Sungbum Lee, Jong-Hyouk Lee, and Byoungsoo Koh. Threat analysis for an invehicle telematics control unit. *International Journal of Internet Technology and Secured Transactions*, 8(4):653–663, 2018.
- [138] Alissa Knight. *Hacking Connected Cars: Tactics, Techniques, and Procedures.* John Wiley & Sons, 2020.
- [139] Makoto Tsubokura, Toshio Kobayashi, Takuji Nakashima, Takahide Nouzawa, Takaki Nakamura, Huilai Zhang, Keiji Onishi, and Nobuyuki Oshima. Computational visualization of unsteady flow around vehicles using high performance computing. *Computers & Fluids*, 38(5):981–990, 2009.
- [140] Yifang Ma, Zhenyu Wang, Hong Yang, and Lin Yang. Artificial intelligence applications in the development of autonomous vehicles: a survey. IEEE/CAA Journal of Automatica Sinica, 7(2):315–329, 2020.
- [141] John S Baras, Xiaobo Tan, and Pedram Hovareshti. Decentralized control of autonomous vehicles. In 42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475), volume 2, pages 1532–1537. IEEE, 2003.
- [142] Norbert Neuendorf and Torsten Bruns. The vehicle platoon controller in the decentralised, autonomous intersection management of vehicles. In *Proceedings of the IEEE International Conference on Mechatronics, 2004. ICM'04.*, pages 375–380. leee, 2004.
- [143] Mark VanMiddlesworth, Kurt Dresner, and Peter Stone. Replacing the stop sign: Unmanaged intersection control for autonomous vehicles. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 3*, pages 1413–1416, 2008.
- [144] Pol Alemany Prats. Performance study of the ieee 802.11 p and etsi geonetworking protocols. Master's thesis, Universitat Politècnica de Catalunya, 2017.
- [145] Alireza Talebpour and Hani S Mahmassani. Influence of connected and autonomous vehicles on traffic flow stability and throughput. *Transportation Research Part C: Emerging Technologies*, 71:143–163, 2016.
- [146] Falco Creemers, Alejandro Ivan Morales Medina, Erjen Lefeber, and Nathan van de Wouw. Design of a supervisory controller for cooperative intersection control using model predictive control. *IFAC-PapersOnLine*, 51(33):74–79, 2018.

- [147] Jason R. Marden, GÜrdal Arslan, and Jeff S. Shamma. Cooperative control and potential games. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 39(6):1393–1407, 2009.
- [148] Assad Al Alam, Ather Gattami, and Karl Henrik Johansson. An experimental study on the fuel reduction potential of heavy duty vehicle platooning. In 13th international IEEE conference on intelligent transportation systems, pages 306–311. IEEE, 2010.
- [149] Carl Bergenhem, Erik Hedin, and Daniel Skarin. Vehicle-to-vehicle communication for a platooning system. *Procedia-Social and Behavioral Sciences*, 48:1222–1233, 2012.
- [150] Stanley W. Smith, Yeojun Kim, Jacopo Guanetti, Ruolin Li, Roya Firoozi, Bruce Wootton, Alexander A. Kurzhanskiy, Francesco Borrelli, Roberto Horowitz, and Murat Arcak. Improving urban traffic throughput with vehicle platooning: Theory and experiments. *IEEE Access*, 8:141208–141223, 2020.
- [151] Aurélien Corréïa, Abdeljalil Abbas-Turki, Rachid Bouyekhf, and Abdellah El Moudni. A dioid model for invariant resource sharing problems. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 39(4):770–781, 2009.
- [152] Meng Zhang, Abdeljalil Abbas-Turki, Alexandre Lombard, Abderrafiaa Koukam, and Kang-Hyun Jo. Autonomous vehicle with communicative driving for pedestrian crossing: Trajectory optimization. In 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), pages 1–6. IEEE, 2020.
- [153] Richard E Kopp. Pontryagin maximum principle. In *Mathematics in Science and Engineering*, volume 5, pages 255–279. Elsevier, 1962.
- [154] Sergei M Aseev and Arkady V Kryazhimskiy. The pontryagin maximum principle and transversality conditions for a class of optimal control problems with infinite time horizons. *SIAM Journal on Control and Optimization*, 43(3):1094–1119, 2004.
- [155] Rafal Kamocki. Pontryagin maximum principle for fractional ordinary optimal control problems. *Mathematical Methods in the Applied Sciences*, 37(11):1668–1686, 2014.
- [156] Nico Tauchnitz. The pontryagin maximum principle for nonlinear optimal control problems with infinite horizon. *Journal of Optimization Theory and Applications*, 167(1):27–48, 2015.
- [157] Denos C Gazis, Robert Herman, and Richard W Rothery. Nonlinear follow-theleader models of traffic flow. *Operations research*, 9(4):545–567, 1961.

- [158] Peter G Gipps. A behavioural car-following model for computer simulation. *Transportation Research Part B: Methodological*, 15(2):105–111, 1981.
- [159] AG PTV. Ptv vissim 10 user manual. PTV AG: Karlsruhe, Germany, 2018.
- [160] Umair Durrani and Chris Lee. Calibration and validation of psychophysical carfollowing model using driver's action points and perception thresholds. *Journal of Transportation Engineering, Part A: Systems*, 145(9):04019039, 2019.
- [161] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2):1805, 2000.
- [162] Fabian de Ponte Müller. Survey on ranging sensors and cooperative techniques for relative positioning of vehicles. Sensors, 17(2):271, 2017.
- [163] Shyam Narayan Patel and V. Prakash. Autonomous camera based eye controlled wheelchair system using raspberry-pi. In 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), pages 1–6, 2015.
- [164] Alexander Warsewa, Michael Böhm, Flavio Guerra, Julia L Wagner, Tobias Haist, Cristina Tarín, and Oliver Sawodny. Self-tuning state estimation for adaptive truss structures using strain gauges and camera-based position measurements. *Mechanical Systems and Signal Processing*, 143:106822, 2020.
- [165] Yaofu Huang, Zengshan Tian, and Qing Jiang. A radar and monocular camerabased fusion approach for pedestrian detection. In *The 2nd International Conference on Computing and Data Science*, pages 1–7, 2021.
- [166] Andrey Belogaev, Alexey Elokhin, Artem Krasilov, Evgeny Khorov, and Ian F Akyildiz. Cost-effective v2x task offloading in mec-assisted intelligent transportation systems. *IEEE Access*, 8:169010–169023, 2020.
- [167] Ibrahim Shaer, Anwar Haque, and Abdallah Shami. Multi-component v2x applications placement in edge computing environment. In ICC 2020-2020 IEEE International Conference on Communications (ICC), pages 1–6. IEEE, 2020.
- [168] Abdallah Moubayed, Abdallah Shami, Parisa Heidari, Adel Larabi, and Richard Brunner. Edge-enabled v2x service placement for intelligent transportation systems. *IEEE Transactions on Mobile Computing*, 20(4):1380–1392, 2020.
- [169] John T Betts, Stephen Campbell, and Claire Digirolamo. Examination of solving optimal control problems with delays using gpops-. Numerical Algebra, Control & Optimization, 11(2):283, 2021.

- [170] Rasheed Al-Salih and Martin Bohner. Quadratic programming problems on time scales. *Appl. Comput. Math*, 19(2):205–219, 2020.
- [171] Sen Chen, Zhixiang Chen, and Zhiliang Zhao. An error-based active disturbance rejection control with memory structure. *Measurement and Control*, 54(5-6):724– 736, 2021.
- [172] Xuguang Hao, Abdeljalil Abbas-Turki, Florent Perronnet, and Rachid Bouyekhf. V2ibased velocity synchronization at intersection. *Mathematical Methods & Computational Techniques in Science & Engineering (MMCTE)*, pages 67–72, 2014.
- [173] Alexandre Lombard. *Contribution à la régulation coopérative des espaces conflictuels entre véhicules autonomes.* PhD thesis, Bourgogne Franche-Comté, 2017.
- [174] Alexandre Lombard, Jocelyn Buisson, Abdeljalil Abbas-Turki, Stéphane Galland, and Abderrafiaa Koukam. Curvature-based geometric approach for the lateral control of autonomous cars. *Journal of the Franklin Institute*, 357(14):9378–9398, 2020.
- [175] Tae Soo No, Kil-To Chong, and Do-Hwan Roh. A lyapunov function approach to longitudinal control of vehicles in a platoon. In VTC2000-Spring. 2000 IEEE 51st Vehicular Technology Conference Proceedings (Cat. No. 00CH37026), volume 1, pages 336–340. IEEE, 2000.
- [176] Lydie Nouveliere et al. Experimental vehicle longitudinal control using a second order sliding mode technique. *Control Engineering Practice*, 15(8):943–954, 2007.
- [177] Sampo Kuutti, Richard Bowden, Harita Joshi, Robert de Temple, and Saber Fallah. End-to-end reinforcement learning for autonomous longitudinal control using advantage actor critic with temporal context. In 2019 IEEE Intelligent Transportation Systems Conference (ITSC), pages 2456–2462. IEEE, 2019.
- [178] Transport for London. Traffic modelling guidelines, 2010.
- [179] Donald Ervin Knuth. Seminumerical algorithms. *The art of computer programming*, 2, 1997.
- [180] Xiqun Chen, Li Li, and Yi Zhang. A markov model for headway/spacing distribution of road traffic. IEEE Transactions on Intelligent Transportation Systems, 11(4):773– 785, 2010.
- [181] Nikolas Geroliminis and Jie Sun. Properties of a well-defined macroscopic fundamental diagram for urban traffic. *Transportation Research Part B: Methodological*, 45(3):605–617, 2011.
- [182] Xiaotong Sun and Yafeng Yin. Behaviorally stable vehicle platooning for energy savings. *Transportation Research Part C: Emerging Technologies*, 99:37–52, 2019.

- [183] Jia Wu, Fei Yan, and Jianxing Liu. Effectiveness proving and control of platoonbased vehicular cyber-physical systems. *IEEE Access*, 6:21140–21151, 2018.
- [184] R Russell Rhinehart. *Engineering Optimization: Applications, Methods and Analysis.* John Wiley & Sons, 2018.
- [185] Aidan O'dwyer. *Handbook of PI and PID controller tuning rules*. World Scientific, 2009.
- [186] Andreas A Malikopoulos, Seongah Hong, B Brian Park, Joyoung Lee, and Seunghan Ryu. Optimal control for speed harmonization of automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 20(7):2405–2417, 2018.
- [187] Sung J Chung and Katta G Murty. Polynomially bounded ellipsoid algorithms for convex quadratic programming. In *Nonlinear Programming 4*, pages 439–485. Elsevier, 1981.
- [188] Huile Xu, Shuo Feng, Yi Zhang, and Li Li. A grouping-based cooperative driving strategy for cavs merging problems. *IEEE Transactions on Vehicular Technology*, 68(6):6125–6136, 2019.
- [189] Jacinto Carrasco, Salvador García, MM Rueda, Swagatam Das, and Francisco Herrera. Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review. Swarm and Evolutionary Computation, 54:100665, 2020.
- [190] Iou-Jen Liu, Unnat Jain, Raymond A Yeh, and Alexander Schwing. Cooperative exploration for multi-agent deep reinforcement learning. In *International Conference on Machine Learning*, pages 6826–6836. PMLR, 2021.

LIST OF FIGURES

2.1	 A- A top view of a four-way intersection, B- Illustration of the three functional areas and the terminology used to analyze the intersection: 4 approaches, 12 movements and 20 conflict points. 	7
2.2	Illustration of the control loop component through a four-way intersection.	8
2.3	Example of phases' definition of a 4 way-intersection.	10
2.4	Cycle time analysis	11
2.5	Cycle time analysis	12
2.6	Illustration of the three main protocols to share the intersection. A-Stop and go, B- Reservation, C- Virtual platoon : Conflicting CAVs are considered as virtual obstacles	18
27	Example of adaptive phase with Stop and Ge protocol (B) compared to	10
2.1	static phase of traffic lights (B)	19
2.8	Potential collision area	20
2.9	Example of virtual platoon, where the CAV considers the real obstacle (rl) virtual obstacles vl , includind the next leader nl and the stop line sl . nl is the last CAV that moving in the distension lane [71]	22
2.10	Lost time computation when the traffic light switches to green	23
2.11	FIFS deadlock examples: A-FIFS deadlock is the result of the reservation of tiles A, B and C, by CAVs 1, 2 and 3 (The number on CAVs are only ID) B-FIFS deadlock is due to communication issues in Virtual Platoon that raises the problem of order inconsistency (The number on vehicles indicates their ranks according to the time they were discovered: 2 waits	
	for 1 who is behind 3)	26
3.1	Virtual platoon formed by two CAVs that does not move in the same storage zone: CAV ₂ adjusts in R_4 its speed according to CAV ₁ that moves in R_1	36
3.2	Virtual platoon test of three robots, where the point of synchronization is at the middle of the storage zone (White paper): The two follower robots move slowly so far to keep a safe distance	39
 2.5 2.6 2.7 2.8 2.9 2.10 2.11 3.1 3.2 	Cycle time analysis	12 18 19 20 22 23 26 36 36

3.3	RT-CVC cruise control principle for safely considering the time delay $ au$	41
3.4	RT-CVC cruise control combined with lateral control [174] to follow a target vehicle (0-50km/h), using radar	41
3.5	Virtual platoon: (a) Intersection, (b) virtual platoon, (C) all considered ob- stacles	42
3.6	Simulated scenario of virtual platoon with 2 conflicting CAVs: CAV2 follows CAV1	44
3.7	Results of the simulated scenario of virtual platoon with 2 conflicting CAVs: A-position curves and B-acceleration curves	46
3.8	Simulated intersection geometry in pixels	48
3.9	Screenshot of the simulator	49
3.10	Fundamental diagram of the traffic (left-side) and the used flow-speed dia- gram (right-side)	50
3.11	Simulation of FIFS policy under the virtual platoon protocol	53
3.12	Simulation of FRO policy under the virtual platoon protocol	55
3.13	Simulation of DCP under the stop and go protocol	57
3.14	Scheduling problem in DCP-VP	58
3.15	Simulations of DCP-VP	59
3.16	Comparison between the four simulated intersection management	60
4.1	Key zones of intersection.	64
4.2	Two kinds of headway <i>h</i> : (a) with the precedent vehicle that moves on the same lane called h_f , (b) with a conflicting vehicle called h_c , $h_c > h_f$	65
4.3	Elementary conflict between two robots: The remained distance here indi- cates $D - p_1(t) - l_c - l_1 - s_1$	66
4.4	λ variation according to the $v_1(t_{0,exit})$	69
4.5	Unfeasible optimal point (in red): A-Cond1 is not satisfied, B-Cond2 is not satisfied. The remained distance here indicates $D - p_1(t) - l_c - l_1 - s_1$	71
4.6	Illustration of both feasibility conditions to reach $v^* = 5.04m/s$ from distance to p^* according to $t_{0,exit}$: $\tau = 0.5s$ and $v_1(0) = 10m/s$	72
4.7	R1 position, speed and acceleration according to initial conditions	74
4.8	Intersection of six robots: Position, speed and acceleration	76

4.9	Comparison between the fitting curves of FIFS and OPFIFS	81
4.10	Simulation results of OPDCPVP: DCPVP with optimal control point	81
4.11	Simulation results of OPFIFS: FIFS with optimal control point	82
4.12	Comparison of the fitting curves of the DCPVP and the OPDCPVP \ldots	83
4.13	Example: (a) CIM based on speed adjustment and FCFS, (b) CIM based on sequence optimization.	84
4.14	Simulation results of PSODCPVP: DCPVP with mobile point of synchro- nization based on distributed PSO	86
4.15	Comparison of the fitting curves of the tree approaches of the DCPVP	87

LIST OF TABLES

2.1	Real tests of CIM	17
3.1	Parameters of the acceleration function	43
3.2	Simulation data of figures 3.7-A and B	45
3.3	Simulated flow rates	51
4.1	Elementary intersection notations	67
4.2	R1 parameters	74
4.3	Five studied scenarios	75
4.4	Intersection of 6 robots	75

LIST OF DEFINITIONS

1	Definition: Cooperative Intersection management	16
2	Definition: Policy	26

PERSOListe de mes publications

AUTHOR'S PUBLICATIONS

[1] W. Du, A. Koukam, A. Abbas-Turki, and F. Gechter "Traffic Jam Reduction Based on Vehicle Platoons and Cooperative Intersection Management". In Futurmob 2018.

[2] W. Du, A. Abbas-Turki, A. Koukam, S. Galland, and F. Gechter, "On the v2x speed synchronization at intersections: Rule based system for extended virtual platooning,"Procedia Computer Science, vol. 141,pp. 255–262, 2018. (The paper obtained the best paper award in the conference)

[3] W. Du, A. Abbas-Turki, A. Lombard, A. Koukam and F. Gechter, "Safe and optimal Intersection of Autonomous and Connected Robots", The International Workshop on Intelligent Systems. IEEE, 2020.

[4] W. Du, A. Abbas-Turki, A. Koukam and F. Gechter "On the Safety Constraints Between Intersecting Movements of Autonomous and Connected Robots". The 2020 International Conference on Control, Automation and Diagnosis. IEEE, 2020.

Document generated with LATEX and: the LATEX style for PhD Thesis created by S. Galland — https://github.com/gallandarakhneorg/tex-templates the tex-upmethodology package suite — http://www.arakhne.org/tex-upmethodology/ **Title:** Traffic congestion reduction based on vehicle platoons and intelligent crossroads interactions **Keywords:** Cooperative driving, Virtual platoons, Connected and autonomous vehicles

Abstract:

Intersections are at the core of urban congestion. For more than a decade, new approaches based on autonomous and connected driving have been proposed. They aim to improve the performance of traffic control at intersections, by harnessing connectivity and driving automation (longitudinal control). These approaches have in common the fact that vehicles can negotiate together their right of way to use the conflicting space. However, they are different in terms of the way they share the space and optimization techniques. The challenge is to define the sequence of access of vehicles to the common space (which one goes first, which is the second, and so on) and the speed profile of vehicles to avoid, if possible, unnecessary stops. The literature shows that it is difficult to optimally solve both problems simultaneously in a dynamic context under strong real-time constraints. To solve the problem with respect to reality, the thesis explores the negotiation protocol as well as the policy that meets the safety requirements and respects the hard real-time constraints of the system. From the safety standpoint, vehicles access conflicting spaces by forming virtual platoons. In this way, they can maintain a sufficient safety gap to be ready to react safely in the case of danger. Regarding the realtime constraints, a rule-based system was chosen to form the sequences. In order to improve the performance of the intersection, two properties were

exploited. The rules allow vehicles that follow each other (property 1) or those that can pass in parallel (property 2) to form groups. The group crosses the intersection together. A distributed right-of-way negotiation algorithm is proposed and compared to other policies of the literature. The simulation shows a significant gain in terms of intersection capacity. To further improve the performance of the proposed cooperative traffic control at intersections, the thesis focused on the longitudinal control issue. It defines an optimal output state achievable, using optimal control theory. Control based on guadratic programming shows the interest of the approach on an elementary intersection. On the one hand, the optimal output state minimizes the headway times between two conflicting vehicles. This improves the throughput of the intersection. On the other hand, it allows the modification of the sequences during the longitudinal control to improve the sequence dynamically according to the new incoming vehicles. The new approach was extended to a complex intersection. Several optimal output state-based sequence formation policies were simulated. The simulation shows that the policy based on distributed particle swarm optimization significantly improves the performance of the intersection in terms of capacity and speed. Distributed particle swarm optimization allows the formed group of platoons to be adapted to the dynamic traffic demand patterns.

Titre : Contribution à la fluidification du trafic basée sur la formation de pelotons de véhicules autonomes et les carrefours connectés et intelligents.

Mots-clés : Fluidification du trafic aux intersections, Pelotons virtuels, Véhicule autonome et connecté

Résumé :

Les intersections sont au centre des congestions urbaines. Depuis plus d'une décennie, de nouvelles approches basées sur la conduite autonome et connectée ont été proposées. Elles visent à améliorer les performances de la régulation du trafic aux intersections, en exploitant la connectivité et l'autonomie de la conduite (commande longitudinale). Ces approches doivent à la fois définir l'ordre des accès des véhicules aux espaces communs (quel véhicule passe en premier, lequel est le second et ainsi de suite) tout en calculant une commande longitudinale pour éviter, si possible, des arrêts inutiles. Il en sort de la littérature la difficulté de résoudre d'une manière Université Bourgogne France de l'Observatoire d'une frantere geux Droblemes simultanement dans université dynamique sous des contraintes temps-25000 Besançon, France

réel dures. Dans cette thèse nous avons supposé que les véhicules forment des pelotons virtuels, en suivant les véhicules des autres voies. Pour décider de l'ordre de passage, le choix s'est porté sur un système à base de règle. Afin d'améliorer les performances de l'intersection, les propriétés des solutions optimales ont été exploitées pour définir des algorithmes distribués de négociation des droits de passage. La simulation montre que la politique basée sur les algorithmes par essaims particulaires distribués améliore considérablement les performances de l'intersection en termes de capacité et de vitesse. Les essaims particulaires distribués sont conçus dans cette thèse pour adapter la formation des pelotons au contexte du trafic.