Universidad de Alcalá

PhD. Program in Information and Communications Technologies

Efficient management of road intersections for automated vehicles – The FRFP system applied to the various types of intersections and roundabouts

PhD. Thesis presented by

Basilio Filocamo

2020

PhD. Program in Information and Communications Technologies

Efficient management of road intersections for automated vehicles – The FRFP system applied to the various types of intersections and roundabouts

PhD. Thesis presented by

Basilio Filocamo

Advisors

Dr. Miguel Ángel Sotelo Vázquez
Dr. Javier Alonso Ruiz

Alcalá de Henares, 01th of April, 2020

To all the people who made this work possible.

Acknowledgments

After several years of graduation and many years of work, it was exciting to be able to return to work for the University of Alcalá. I thank Dr. Miguel Ángel Sotelo Vázquez and Dr. Javier Alonso Ruiz for the opportunity they have given me to collaborate with a very professional group. I also thank them for the advice, for the ideas and for the guide in general that they provided me without which I would certainly not have achieved the same results obtained.
The work done was the umpteenth challenge of my professional career that made me grow professionally and that will allow me to make a profitable use of it in the future.

I would also like to thank all the members of the INVETT group from the University of Alcalá for their support. I would also like to thank Dr. Massimo Villari of the University of Messina for giving me the opportunity to collaborate with his work group during this doctoral period.

I would also like to thank my whole family. My parents, who have always believed in me by supporting me in my choices, have made my human and professional journey possible. My children and my wife who have always supported me in the hope that they can be proud of my work.
I hope that this work done with so much passion and many sacrifices can be useful to the scientific community.

Synthesis

In recent years, we have witnessed an increasingly rapid development of road vehicle technology. This is enabling increasingly safe driving, for example through pedestrian protection systems, automatic braking systems and so on. Surely the main objective in the use of technology in road systems is, and must be, safety. Important use of technology in vehicles is already leading to automatic driving without a driver. Here again one of the main aspects is safety with the goal of zero accidents. Clearly, there are other aspects as well, always leaving safety first, such as travel times, fuel consumption, emissions and therefore air quality depending on the characteristics of the place in question. One of the most critical aspects of automatic driving without a driver is the management of road junctions. In fact, intersections are the points with the greatest problems related to safety and on which all the variables mentioned above depend closely.

This thesis introduces and describes a new algorithm for the management of road intersections (FRFP - first to reach the end of the intersection first to pass) which aims at a safe management with optimized crossing times and therefore with related reductions in fuel and emissions. Several state-of-the-art systems have been analysed and compared. The idea was to look for a natural method of optimisation when crossing a junction. I started by trying to understand what we humans do instinctively when we are driving and at a crossroads without any precedence. In most cases we try to understand if we can reach the end of the intersection before the vehicle involved in the intersection can reach the beginning of the intersection. In particular, we evaluate whether by applying an acceleration or maintaining our speed we can reach the end of intersection first without collisions. Otherwise we will give priority to the other vehicle involved. If we think we can do it, perhaps accelerating a little, we proceed to the crossing, vice versa we are induced to brake or slow down by passing the competing vehicle. By making this reasoning with much more precise estimates to an algorithm, the system can be very efficient, so the FRFP (first to reach the end of the intersection first to pass) system is born.

Another problem with the introduction of driverless vehicles is the installation of Autonomous Intersection Managers (AIM) who would be responsible for processing the information and then control the behaviour of vehicles. In this case, then, an intersection manager evaluates and makes decisions for all the vehicles involved. The high number of intersections would make it difficult to install AIM. This problem is solved by applying the algorithm developed through Vehicle-to-Vehicle (V2V) communication between the vehicles involved. The V2C communication protocol for Vehicle-to-Cloud communications was examined to optimize the algorithm even at long distances from the intersection and for possible route optimizations based on traffic and presence of pedestrians, emissions in inhabited areas where certain limits should not be exceeded also depending on weather conditions.

The algorithm has been tested in different types of intersections (2 lanes intersections, ramp-ups, 8 lanes intersections) and on roundabouts. The results were very interesting, showing a significant

improvement in average crossing speeds, fuel consumption and emissions compared to state-of-the-art algorithms.

Key words: automated vehicle; intersection management; roundabout; FRFP

Abstract

In the last decade, the automatic driving systems for vehicles circulating on public roads are becoming more and more a reality. There is always a stronger interest from both research centres and car manufacturers. One of the most critical aspects is the management of the intersection; who will have to go first and in what ways? This is the question we want to answer through this research. Clearly the goal is to manage the intersection safely, making it possible to reduce road congestion, travel time, and emissions and fuel consumption as much as possible. The research is conducted by comparing a new management system with the systems already known in the state of the art for different types of intersections. The new system proposed by us is called FRFP (first to reach the end of the intersection first to pass). In particular, vehicles will increase or decrease their speed in collaboration with each other by making the right decision. The vehicle that can potentially reach the intersection exit first. Even if the work done was born without the need to take into account possible communication errors, the problem was, however, overcome through the concept of blockchain. An error in communication or hacking of information between vehicles could lead to possible collisions. In order to ensure correct information between vehicles, it was decided to apply blockchain algorithms. Assuming a V2V communication, the system provides for unanimous agreement between the vehicles involved. In full agreement, therefore, the vehicles will cross the intersection as established by the FRFP algorithm. Otherwise, the vehicles will reset all decisions and will safely restart from the point of intersection. This system, which guarantees its total safety, avoids any communication or processing errors that may occur.

Keywords: automated vehicle; intersection management; roundabout; FRFP

Contents

Acknowledgments ... xi
Synthesis ... xii
Abstract ... 1
Contents ... 2
List of Figures ... 4
List of Tables ... 6
List Acronyms ... 7
Chapter 1 ... 8
Introduction and Related Work ... 8
Related Work ... 11
1.1 Types of Communication ... 11
1.2 State of art algorithms ... 13
Chapter 2 ... 23
SUMO (Simulation of Urban Mobility) ... 23
Basic concepts ... 23
Creation of road networks ... 24
Generation of the road population ... 26
Simulation ... 28
Chapter 3 ... 30
Developed algorithm: FRFP ... 30
Chapter 4 ... 43
Implementation of the FRFP system ... 43
Creation of road networks ... 43
Generation of the road population ... 45
Developed algorithm ... 46
Chapter 5 ... 55
Result ... 55
Intersection type 1 (2 lanes): ... 55
Intersection type 2 (on-ramp): ... 58
Intersection type 3 (8 lanes): ... 60

Intersection type 4 (roundabout): 65
Chapter 6 67
Main Contributions 67
Future Work 68
Chapter 7 69
Bibliography..69

List of Figures

Figure 1 Evolution of the number of road fatalities in the EU and targets for 2001-2020 9
Figure 2 Diagram of Intersection System 13
Figure 3 Case of Confirm Reservation 14
Figure 4 Case of Reject of Reservation 14
Figure 5 Cooperative Vehicle Intersection Control (CVIC) Trajectory Overlaps 15
Figure 6 Vehicle Synchronized at Begin Intersection 16
Figure 7 Ballroom Vehicle Crossing 17
Figure 8 Detect Zipper 18
Figure 9 FAFP Algorithm Traffic Scene 19
Figure 10 Consensus formation on movement priorities for conflicting trajectories 20
Figure 11 Speed synchronization at intersection 21
Figure 12 Schermata grafica NETEDIT Tool 24
Figure 13 Script Python to generate .rou.xml file 27
Figure 14 Example of file.rou.xml 27
Figure 15 Example of how to use the TRACI library 29
Figure 16 Intersection management with only two vehicles 30
Figure 17 Intersection management 2 lanes intersection 32
Figure 18 Block Diagram for priority calculation. 34
Figure 19 Regulation State case 1 36
Figure 20 Balance State case 2 38
Figure 21 On-ramp intersection 38
Figure 22 All possible vehicle trajectories 39
Figure 23 Trajectories without collisions with the cars in the right direction 39
Figure 24 Trajectories without collisions with the cars in the down_right direction 40
Figure 25 Block Diagram 41
Figure 26 Roundabout System 42
Figure 27 Intersection to two lanes (Netedit) 43
Figure 28 Intersection on ramp (Netedit) 44
Figure 29 Intersection to 8 lanes (Netedit) 44
Figure 30 Roundabout system (Netedit) 45
Figure 31 Script Phyton to generate file.rou.xml 46
Figure 32 List_priority function 47
Figure 33 Time Calculation to reach end intersection 48
Figure 34 Priority List in Intersection 49
Figure 35Function of Modulation speed 50
Figure 36 Congestion 51
Figure 37 Freeze Pre intersection 52
Figure 38 Freeze Post intersection 53
Figure 39 Roundabout Vs Intersection 54

Figure 40 Crossing 2 lanes: inserted 558 vehicles equivalent to about 1956 vehicles/h for Algorithm Implemented [FRFP]........56
Figure 41 Crossing 2 lanes: inserted 2000 vehicles equivalent to about 3442 vehicles/h for Algorithm Implemented [FRFP]........57
Figure 42 On-ramp crossing: inserted 465 vehicles equivalent to about 1773 vehicles/h for Algorithm Implemented [FRFP]........58
Figure 43 Crossing 8 lanes: inserted 468 vehicles equivalent to about 1640 vehicles/h for Algorithm Implemented [FRFP]........61
Figure 44 Crossing 8 lanes: inserted 587 vehicles equivalent to about 1950 vehicles/h for Algorithm Implemented [FRFP]........62
Figure 45 Crossing 8 lanes: inserted 685 vehicles equivalent to about 2380 vehicles/h for Algorithm Implemented [FRFP]........63
Figure 46 Crossing 8 lanes: inserted 808 vehicles equivalent to about 2535 vehicles/h for Algorithm Implemented [FRFP]........64
Figure 47 Crossing 8 lanes Vs Roundabout: inserted 685 vehicles equivalent to about 2378 vehicles/h for Algorithm Implemented [FRFP]........65

List of Tables

Table 1 Intersection 2 lanes 1950 Vehicles/Hour 56
Table 2 Intersection 2 lanes 3450 Vehicles/Hour 57
Table 3 Intersection on-ramp 1773 Vehicles/Hour....59
Table 4 Intersection 8 lanes 1640 Vehicles/Hour 60
Table 5 Intersection 8 lanes 1950 Vehicles/Hour 61
Table 6 Intersection 8 lanes 2380 Vehicles/Hour 62
Table 7 Intersection 8 lanes 2535 Vehicles/Hour 63
Table 8 Intersection 8 lanes Vs Roundabout 2378 Vehicles/Hour 66

List Acronyms

AIM	Autonomous Intersection Management
AVs	Autonomous Vehicles
BRIP	Ballroom Intersection Protocol
CAVs	Connected Autonomous Vehicles
CVIC	Cooperative Vehicle Intersection Control
DC-SICL	Distributed Coordinated Signal-free Intersection Control Logic
DLR	German Aerospace Centre
FCFS	First Come, First Served
FAFP	First Arrive, First Pass
FRFP	First to Reach the end of the intersection First to Pass
HQEP	Higher QoS Earlier Pass
HWFP	Lane Weight Based Policy
MILP	Mixed Integer Linear Programming
MINLPs	Mixed-Integer Non-Linear Programs
MLAP	Multi Lane Authorization Policy
NETEDIT	Network Editor
PAP	Platoon based Authorization Policy
QoS-CITS	One Self-Designed Traffic Simulator
SIAP	Synchronized Intersection Arrival Pattern
SUMO	Simulation of Urban Mobility
TraCI	Traffic Control Interface
TCP	Transmission Control Protocol
V2V	Vehicle-to-Vehicle
V2I	Vehicle to Infrastructure
V2C	Vehicle to Cloud
XML	eXtensible Markup Language

Chapter 1

Introduction and Related Work

In the last few years, more and more companies (Google, Nissan, Tesla, etc.) and university research centres have been dealing with automatic driving systems. In particular we are analysing solutions for cooperation between vehicles, especially in the context of urban mobility. Undoubtedly, the intersections represent one of the most complex and important scenarios to be managed for automatic vehicles. In fact, the level of interactions between vehicles, in these contexts, is very high.

According to the European Commission's statement in the field of Mobility and Transport, while road safety in the European Union has improved considerably in recent decades (and EU roads are the safest in the world), the number of deaths and injuries is still too high. This is why the EU has adopted the "Zero Fatality Target" approach and a safe system to prevent fatal or serious accidents on Europe's roads.

The focus is on working together on road safety with Member States' authorities to develop national initiatives, set targets and address all factors affecting accidents (infrastructure, vehicle safety, driver behaviour and emergency response).

In May 2018, as part of the "Europe on the move" package, the European Commission presented a new approach to EU road safety policy together with a medium-term strategic action plan.
Worldwide, the number of road accident victims continues to grow. According to the World Health Organisation's report on global road safety entitled 'Global status report on road safety', in 2016 alone, there were 1.35 million road accident victims. This means that more people worldwide die as a result of road accidents than from HIV/AIDS, tuberculosis or diarrhoeal diseases. In addition, worldwide, road accidents are now the most common cause of death for children and young people between the ages of 5 and 29.

Compared to the global context, the situation in Europe is relatively positive, thanks to decisive action at EU, national, regional and local level. Between 2001 and 2010, the number of road deaths in the EU fell by 43 %, and by a further 21 % between 2010 and 2018.

In 2018, however, a further 25.100 people were killed and some 135.000 seriously injured on EU roads. From a human and social point of view, this is an unnecessary and unacceptable price to pay for mobility. A recent study estimated that, if only the economic aspect is taken into account, road accidents in the EU cost around EUR 280 billion a year, equivalent to around 2 % of GDP.

A very important factor to consider is that at EU level in recent years there has been stagnation in the progress made in reducing road fatality rates. It is highly unlikely that the EU will achieve the current medium-term objective of halving the number of road accident victims.

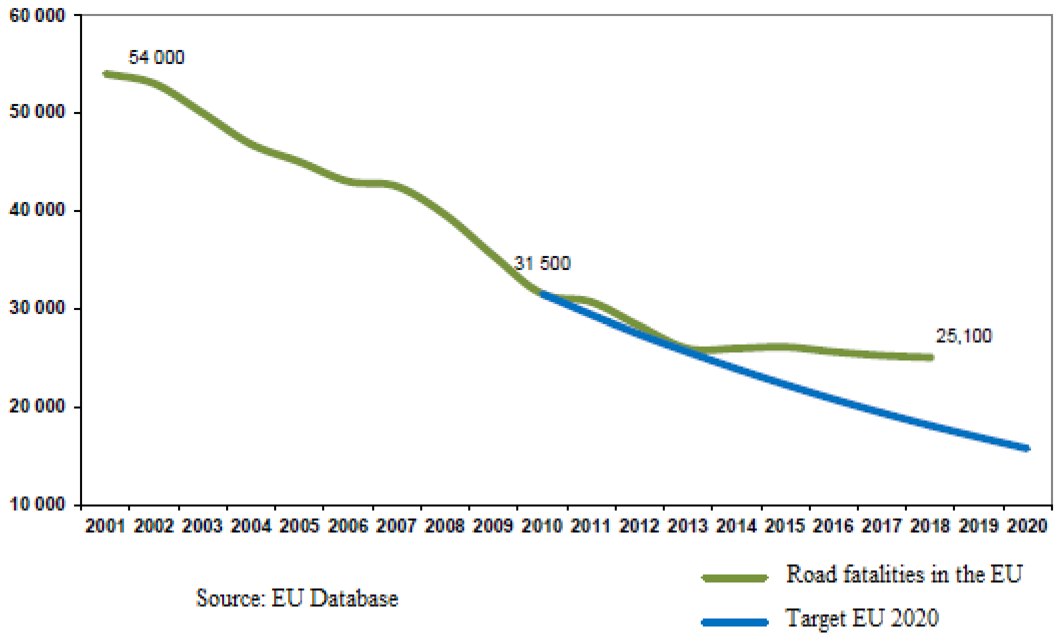


Figure 1 Evolution of the number of road fatalities in the EU and targets for 2001-2020[1]

The European Union has reaffirmed its ambitious long-term goal of approaching zero casualties by 2050 ("Vision Zero"). In March 2017, for the first time, EU Transport Ministers set a target for the reduction of serious injuries, i.e. halving the number of serious injuries in the EU by 2030 compared to 2020 figures.

The Communication "Europe on the move" sets out a new approach to achieving these targets.
A key point addressed by the European Commission is to address new trends, such as the growing phenomenon of distractions due to mobile devices. Some technological advances, especially in terms of connectivity and automation, will open up new opportunities for road safety in the future, reducing the burden of human error.

An important objective to achieve the "Zero Collision" target is certainly to use collaboration between vehicles together with the new technologies developed in recent years for road safety.
Road infrastructure throughout the EU has deteriorated due to poor maintenance. Maintenance budgets have often suffered significant cuts and have not gone hand in hand with the expansion of infrastructure and the ageing of major links. This has led to a deterioration in the state of the roads in many EU countries and has increased not only the risk of accidents, but also congestion and noise, with consequences for pollution and fuel consumption.

In order to achieve the ambitious "Vision Zero" objective, vehicles will have to work together in dialogue and take joint decisions in order to optimise their road route in total safety. In particular,

[1] Source: EU Road Safety Framework 2021-2030 - Next steps towards the "Vision Zero" target (COMMISSION EUROPEAN, Brussels, 19.6.2019, SWD(2019) 283 draft)

they will have to pay particular attention to their crossing at road junctions, avoiding collisions in the first place and pursuing the reduction of crossing times, emissions and fuel consumption. Equally important is certainly the interaction with the system of pedestrians and objects in general that occupy the road routes.

A bad management of the intersections certainly gives rise to possible road congestion, especially if there is a high presence of vehicles. The vehicles will therefore have to work together to modulate their speed to make the above mentioned possible.

The same scenario can be characterized by different conditions, for example, an intersection with low presence of vehicles is certainly less critical in management than a strongly congested crossroads. In the latter case, the decisions made are of fundamental importance to avoid a critical increase in congestion.

This work examines the development of a new intersection management system by comparing it with other state-of-the-art systems. Different types of intersections are analysed, including roundabouts, in different vehicle flow conditions.

The approach used allows the management of collaboration between vehicles by means of vehicle-to-vehicle communication (V2V). This type of communication offers several advantages compared to other systems that can also be used in our system. For example, management based on the presence of the Autonomous Intersection Manager (AIM) has the great disadvantage of requiring the installation of intersection managers at each intersection thus making the initial phase of real implementation of these systems at the urban level very complicated.

In a less critical and more extensive approach, the proposed system can provide a vehicle-to-cloud-edge communication (V2C) for a coarser collaboration already at long distances. This can also be used for intelligent routing of vehicles based on the emission limits imposed according to the real traffic and weather conditions of the areas concerned. The use of this type of communication can also be used for optimization in the choice of routing according to different parameters such as, for example, the reduction of road congestion and the facilitation of pedestrian crossings. Furthermore, the possibility of collaborating over long distances allows us to limit abrupt variations in speed with a consequent reduction in consumption, CO2 emissions and vehicle wear. Our work is focused on the management-collaboration of vehicles near the intersection (50-100 meters) but it is also proposed to give an important starting point on the use of the same system in V2C communication.

The objective of the work has been focused on determining an algorithm that could have a higher performance than those characterizing state of the art systems. A first publication ([1] B. Filocamo, A. Galletta, M. Fazio, J. A. Ruiz, M. Á. Sotelo and M. Villari, "An Innovative Osmotic Computing Framework for Self Adapting City Traffic in Autonomous Vehicle Environment," *2018 IEEE Symposium on Computers and Communications (ISCC)*, Natal, 2018, pp. 01267-01270.) defines the basis of the algorithm and lays the foundation for the application by means of V2V and V2EC

communication protocol, the latter protocol allows us to optimize the management of the intersection and more generally the routing already at long distances.
The objective led to the development of a new management algorithm that we called FRFP ([2] Filocamo, B.; Ruiz, J.A.; Sotelo, M.A. Efficient Management of Road Intersections for Automated Vehicles—The FRFP System Applied to the Various Types of Intersections and Roundabouts. *Appl. Sci.* **2020**, *10*, 316.) but also introduced the use of the blockchain, never used until now in this field, to ensure the reliability of the information exchanged between vehicles. From the use of the blockchain as a guarantee system for the reliability of communications between vehicles, thanks to the work done, is published the article: A. Buzachis, B. Filocamo, M. Fazio, J. A. Ruiz, M. Á. Sotelo and M. Villari, "Distributed Priority Based Management of Road Intersections Using Blockchain," *2019 IEEE Symposium on Computers and Communications (ISCC)*, Barcelona, Spain, 2019, pp. 1159-1164 [3]

Related Work

The management of road intersections has become in recent years a fundamental theme in the management of automatic vehicles. Our research starts from an accurate analysis of the state of the art on this topic and then investigates possible solutions not yet explored and that can improve the efficiency of the intersection management algorithms with autonomous vehicles (AVs).

1.1 Types of Communication

Most road accidents occur because drivers of vehicles do not clearly perceive the actions of other vehicles and/or do not respond promptly to their surroundings. Thus, drivers of vehicles are unable to take an action that complements the other vehicles ones.

If vehicles are able to clearly communicate their actions to other vehicles (CAVs), collisions can be avoided or mitigated. In order to achieve this, it is important that vehicles can communicate with each other directly or through common interlocutors. The objective of vehicle-to-vehicle communication is, in general, to prevent accidents by allowing vehicles in transit to send each other data such as position, speed and acceleration. Depending on how the technology is implemented, the driver of the vehicle can simply receive a warning in case of risk or the vehicle itself, in the case of automatically driven vehicles, can adopt a speed modulation to prevent an accident. Therefore, the main objective of the development of communication technologies between vehicles is to improve road safety and reduce the number of accidents.

Vehicle-to-Vehicle Communication (V2V)

Vehicle-to-vehicle (V2V) communication is a direct wireless transmission between vehicles, allowing two vehicles to "talk" or "communicate" with each other.

An intelligent transport system will use vehicle-to-vehicle communication data to improve traffic management, allowing vehicles to also communicate with road infrastructure, such as traffic lights and signage. This technology could become mandatory in the future and help put driverless vehicles on the roads of the world.

The implementation of V2V communication and an intelligent transport system certainly needs to agree on rules and procedures for data confidentiality. When vehicles equipped with V2V communication connect to each other, they exchange information including position, speed, direction of travel, etc., in order to ensure that the vehicles are in a safe and secure environment.

The advantages of this type of Vehicle-to-Vehicle (V2V) communication are: Immediate application compared to communication systems that provide an operator of intersection (AIM); The physical installation of intersection management systems would entail considerable implementation time and high costs. The limitations of this type of communication, however, also found in other types of communication are related to the threats of hacking. These communications are vulnerable to hacking attacks that can alter the information sent to vehicles. The consequences of such attacks can be well understood. The security aspect of the information sent/received is a fundamental aspect that must be analysed and managed.

Vehicle-to- Infrastructure Communication (V2I)

Vehicle-to-infrastructure (V2I) communication is the wireless exchange of data between vehicles and road infrastructure. V2I communication is typically wireless and bidirectional between vehicles and the road system in general. Particularly important for the management of automatic vehicles is the communication between the vehicles and the intersection manager (AIM). The disadvantage of this type of communication is that it is necessary to install intersection managers for each intersection. The high number of intersections in the world makes this system difficult to implement in a short time. In this case there is a decision-making system that can be centralized, probably even more risky for all aspects of security against hacker attacks.

Vehicle-to-Cloud Communication (V2C)

Another communication system is the Vehicle-to-Cloud system. This type of communication has the great advantage of having no limits in terms of communication distances. Therefore, it can be used already at long distances to choose an optimal route to avoid congestion or to make pre-regulations of the speed in order to synchronize the crossing of crossings in safety already at long distances. Therefore, a communication that facilitates in the best way the synchronization of vehicles optimizing fuel consumption and efficiency.

This communication system is very interesting for several aspects. It is interesting for the collection of emissions data, for insurance policies based on driving style and actual use of vehicles, for fleet owners who have to track their vehicles. This communication could also be used to provide services to drivers tailored to their exact location and habits.

1.2 State of art algorithms

1.2.1 FCFS for intersection manager

K. Dresner and P. Stone in [4] *and* [5] present an intersection management reservation-based approach. The proposed system is based on the coordination of the intersection by the presence of AIM (Autonomous Intersection Management). The reservation policy is based on the FCFS system (First Come, First Served). This system results in some situations not very effective. The vehicle near the intersection could have a low speed compared to a more distant vehicle that, therefore, could potentially be the first to overcome the intersection with a minimum variation of the parameters of the vehicles involved. In this case the FCFS management could be more expensive in terms of time and efficiency (consumption, CO2 emissions, vehicle wear).

In this system the intersection manager (AIM) simulates the crossing of vehicles that require authorization to pass through the intersection and if no conflicts are detected, the AIM issues a reservation to the vehicles concerned. Vehicles are not allowed to access the intersection if they do not obtain authorization from the intersection manager. The procedure therefore provides, as shown in *Figure 2* Diagram of Intersection System, that the single car makes a request to the intersection manager who then processes the various requests and confirms or rejects the request.

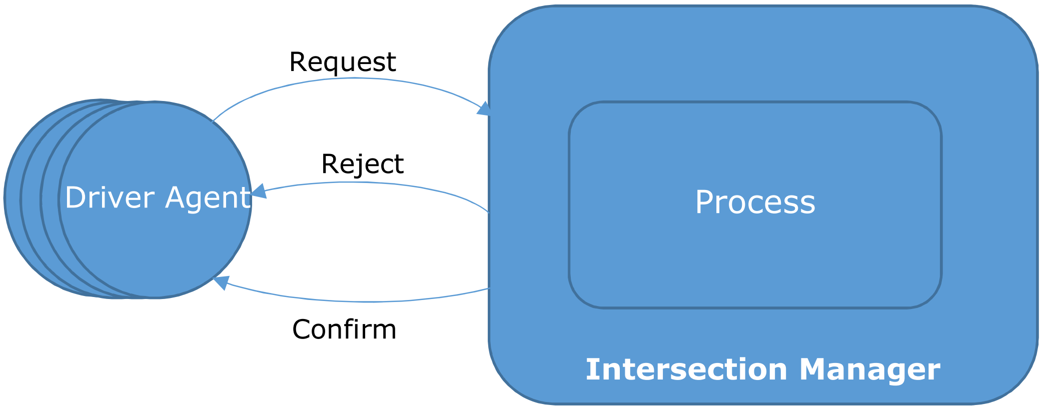


Figure 2 Diagram of Intersection System

If the request is made by the individual vehicle (*Figure 3 Case of Confirm Reservation*) or if the trajectories of several vehicles requesting the crossing present no conflict, the cars will receive the reservation confirmation for crossing the intersection.

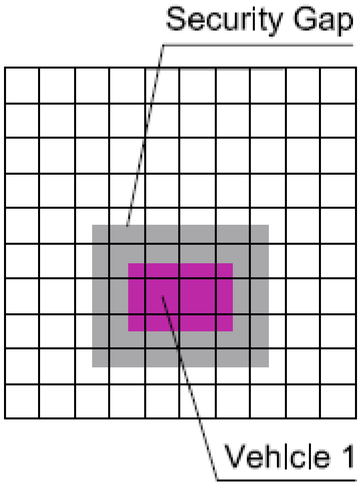


Figure 3 Case of Confirm Reservation

In the event that a conflict is detected, the intersection manager rejects the request. Clearly, since the system is based on an FCFS (First Come, First Served) protocol, the vehicle that first arrives has the right to cross the intersection.

Vehicles that make the request later will have to wait for the AIM response. For example, in *Figure 4 Case of Reject of Reservation* if vehicle 1 is the one that arrives first at the intersection has the right to pass. Vehicle's 2 path conflicts with vehicle's 1 path and therefore, not having priority, must give precedence. The AIM will evaluate the safe passage of vehicle 2 after vehicle 1.

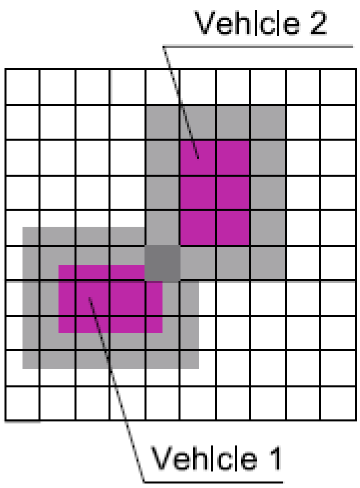


Figure 4 Case of Reject of Reservation

As shown in the figure the AIM will reject the request because the trajectories of the vehicles involved present conflicts. The system will consider both the actual size of the car and a safety gap around it. The vehicle that does not get the reservation of the intersection must necessarily stop at the beginning of the intersection and wait for the reservation. Any collisions are analyzed through information on the speeds and on the trajectories sent by the vehicles.

Tsz-Chiu Au and Peter Stone in [6] present a system for managing vehicle parameters (acceleration, speed) at the intersections in FCFS systems managed by AIM. They introduce a protocol to optimize vehicle routes to avoid possible collisions and then release the reservation imposing speed changes. The system is certainly very effective compared to a classic traffic light system, but in the opinion of the writer it is not very effective in situations where for example the priority vehicle, as the first to make the reservation request, has a low speed compared to a second vehicle which, despite being able to cross the intersection effectively first, will have to wait for the priority vehicle. This could lead to abrupt requests to slow down fast vehicles that could easily cross the intersection thanks to small reductions in the speed of other vehicles which, despite being closer to the intersection, have low speeds.

1.2.2 CVIC for intersection manager

The authors in [7] present a Cooperative Vehicle Intersection Control (CVIC) that manages the trajectories of the vehicles involved in an intersection so that the vehicles do not suffer collisions. The approach involves the exclusive presence of automatic vehicles. The intersection manager processes all the trajectories of the vehicles involved in crossing the intersection to avoid that there may be overlaps. In this way all vehicles not subject to trajectory overlaps with other vehicles will have the authorization to proceed.

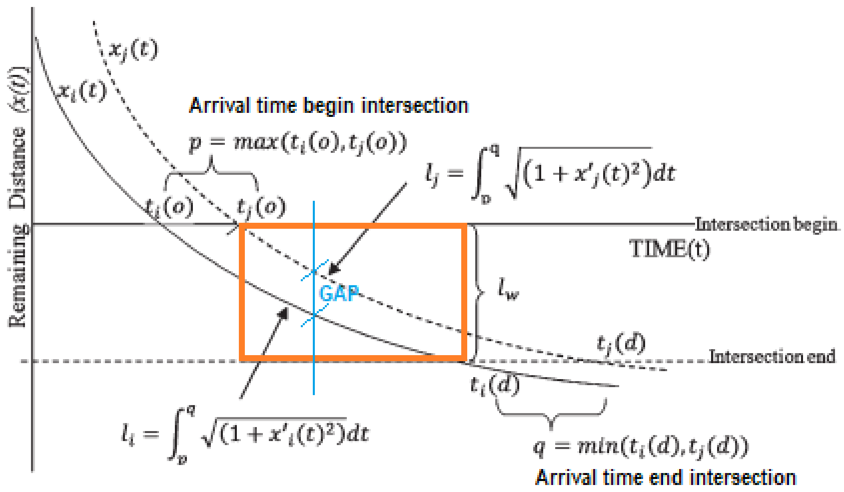


Figure 5 Cooperative Vehicle Intersection Control (CVIC) Trajectory Overlaps[2]

Vehicles with incompatibility trajectory will be placed in a non-priority condition and all the time trajectories of the vehicles involved will be optimized. The elaboration takes place by mathematical processing of the trajectories, constraining a minimum guaranteed gap between the vehicles.

[2] Source: J. Lee and B. Park, "Development and Evaluation of a Cooperative Vehicle Intersection Control Algorithm Under the Connected Vehicle Environment", 2012

1.2.3 BRIP protocol

In [8] a system based on Vehicle-to-Vehicle communications (V2V) is proposed. The system provides for the modulation of the speeds for the synchronized crossing of vehicles at intersections. The presented protocol Ballroom Intersection Protocol (BRIP) takes inspiration from the synchronization of the participants in the Ballroom dancing. The system is very efficient as the vehicles will arrive at the intersection at the same time and will cross it at the same time occupying a very precise cell. The intersection is divided into cells never occupied by several vehicles at the same time. This system, although very efficient in homogeneous traffic conditions, has several limitations: all vehicles must have the same speed and same size; it is a system without priority and with difficult management of emergency; in non-homogeneous traffic conditions this system does not dispose of the traffic by balancing the congestions for their quick disposal. Furthermore, this approach cannot be used in certain types of intersections such as roundabouts, 2 lane intersections and connecting ramps.

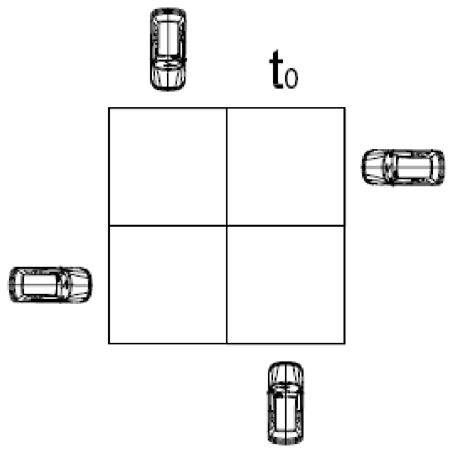


Figure 6 Vehicle Synchronized at Begin Intersection

The protocol is based on the Synchronized Intersection Arrival Pattern (SIAP), i.e. each vehicle When defined both the access point and the exit point will synchronize with the other vehicles in order to reach the start of the intersection with the same speed. At this point the vehicles will cross the intersection, mixing without collision.

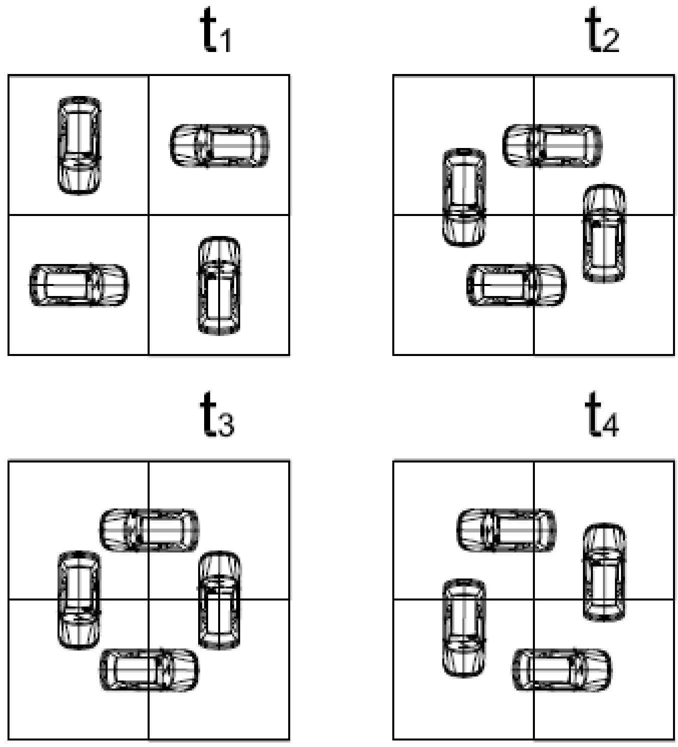


Figure 7 Ballroom Vehicle Crossing

Subsequent vehicles that must cross the intersection must respect a distance from the vehicle that precedes them equal to at least twice the sum of the length and width of the vehicles. The constraints on the speed and on the distances between successive vehicles together with the synchronization will guarantee the collision-free crossing. This system, although presenting a remarkable effectiveness in terms of crossing times, presents the big limits already previously exposed, therefore, of difficult practical application in most real contexts.

1.2.4 Autonomous Vehicle Protocol for Merge Points

In [9] a system based on the priority of lanes is proposed. The protocol in question is based on V2V communications and sensor-based perception systems. The vehicles will need to have a database with maps, sensors, vehicle control device, wireless communication systems and a location system. Merge points are intersections generated by the meeting of two lanes with different priorities. In this protocol 10 types of interaction messages between vehicles are identified. In particular the following messages will be used: ENTER, CROSS (indicates that the transmitting vehicle is inside the intersection), EXIT (Indicates the vehicle exit from the intersection), ABOUT TO EXIT (message used to support road restrictions), REQUEST (vehicle in phase of approach to the intersection), DECLINE (message of response to the Request), APPROVE (message of approval to an approach request to the intersection), INTERRUPT, YIELD, STATE TRANSITION (indicates the congestion of the intersection).

The CROSS message according to this protocol is not necessary as the sensor system is supposed to be sufficient to detect the presence of vehicles inside the intersection.

The System has two different protocols, one to be applied in the presence of vehicles of an automatic type only and one with the presence of non-automatic vehicles. In the case of automatic vehicles, the approach is based on three different states: *Full-Prioritized*, *Semi-Prioritized* and *Fair-State*. The system provides for the blocking of vehicles in non-priority lanes and does not manage any congestion conditions of the lanes. In the *Full Prioritized* state only vehicles that occupy the priority lane will pass the intersection, the other vehicles will cross only in the absence of conflicts. In the state *Semi-Prioritized* the vehicles on the non-priority lane can cross the intersection only if the other lane, the priority lane, is not congested. The *Fair State* is a change of state that can receive the vehicle stationary on a non-priority lane.

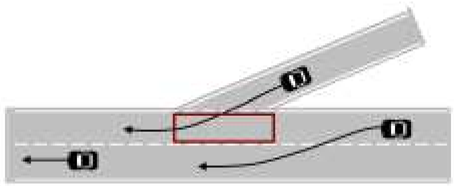


Figure 8 Detect Zipper[3]

In the case of the presence of manual vehicles, on the other hand, some tools are introduced such as the *DetectZipper*, through which an automatic vehicle can change lanes to make the insertion of a manual vehicle easier. This tool requires the perfect perception of what is happening through a sensor system, one of the major limitations of this intersection management system.

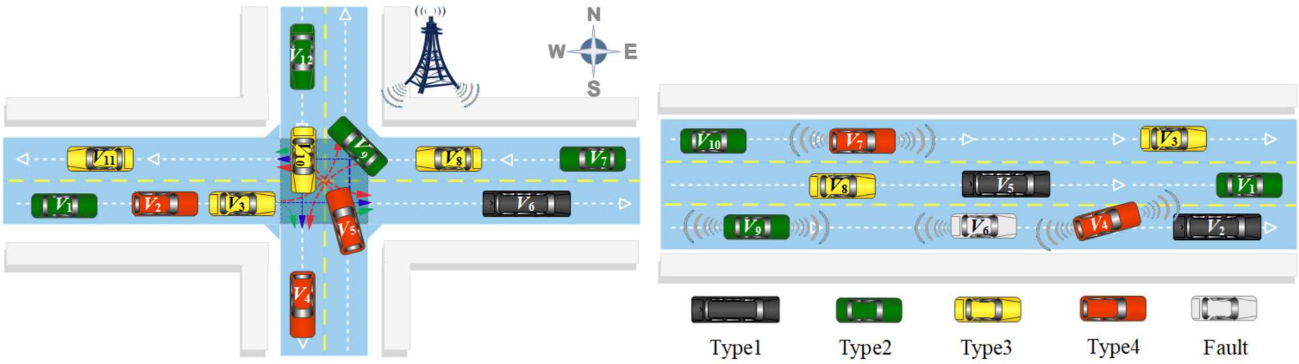
[3] Source: S. Aoki and R. Rajkumar, "A Merging Protocol for Self-Driving Vehicles", 2017

1.2.5 MILP

The authors in [10] propose an approach based on an intersection manager that processes the parameters of all the vehicles periodically determining the optimal solution for crossing the intersection. The system is applied assuming straight trajectories of the vehicles that therefore will not turn at the intersections. An algorithm based on vehicle arrival priority is applied.

1.2.6 Service-Oriented Cooperation Policies

The authors in [11] propose procedures for the management of vehicles according to priorities related to services. In essence, solutions are proposed for priority vehicles regardless of, for example, ambulances or law enforcement or emergency vehicles in general. The authors, for the first time in 2018, consider the algorithm FAFP (First Arrive, First Pass) as the basic algorithm; on this algorithm they build a series of sub algorithms (FAFP-SV, FAFP-SQ, FAFP-SQ-SV, HQEP, HWFP, ...) based on the priority, besides the arrival time, also on their priority status (L-Low, M-Medium, H-High) and on a possible priority of the lanes involved. The work compares the efficiency of the crossing times of the priority vehicles.



(a) Cooperation at One Intersection with Eight Lanes (b) Cooperation on Three Straight-going Lanes

Figure 9 FAFP Algorithm Traffic Scene[4]

This work compares the FAFP algorithm in the case of application to individual vehicles, in the case of platoon application (PAP), in the case of multi-lane authorization policy (MLAP) both in the case of management of the intersection area as a single cell and in the case of subdivision of the same into sub-cells. All the work is simulated through a proprietary simulator developed in C language of their realization called QoS-CITS. The results are interesting even if the analysis of the platoon management takes into account the timing of crossing the priority vehicle and does not take into account the average crossing time of all other vehicles and therefore any congestion determined.

[4] Source: K. Zhang, C. Xie, Y. Wang, M. Wang, A. De La Fortelle, W. Zhang and Z. Duan, "Service-Oriented Cooperation Policies for Intelligent Ground Vehicle Approaching Intersections," *MDPI Applied Sciences,* 2018

1.2.7 A consensus-based distributed trajectory control

The authors in [12] proposes a distributed cooperation for the resolution of possible conflicts of the trajectories of automatic vehicles at an intersection.

The work presented refers to the resolution of the conflict of vehicle trajectories with the aim of optimizing the crossing times of the intersection. The system is based on a distributed approach in which vehicles agree on the trajectories to be followed by means of a non-linear function. The aim is to minimise the crossing time and speed variation.

The agreement phase between the vehicles has been managed in processing steps where the process ends when the vehicles have agreed on their time trajectories.

This paper develops a distributed coordinated signal-free intersection control logic (DC-SICL) to prioritize conflicting movements in intersections when no traffic control devices exist.

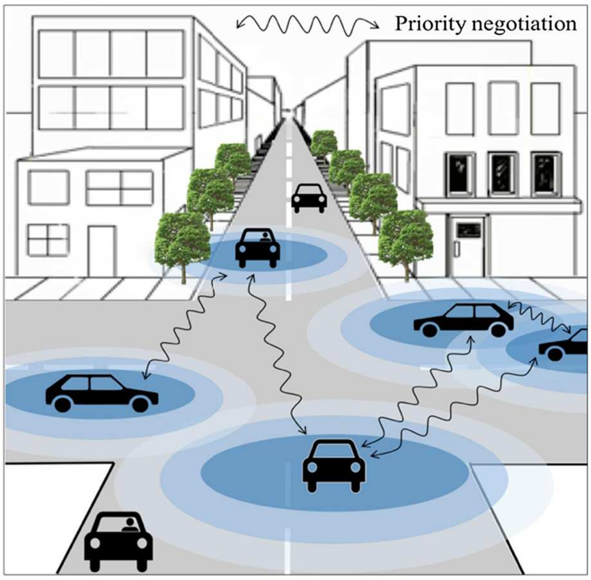


Figure 10 Consensus formation on movement priorities for conflicting trajectories[5]

The cooperative trajectory planning problem is formulated as vehicle-level mixed-integer non-linear programs (MINLPs) that aim to minimize travel time and avoid near-crash conditions for each vehicle. The System adjusts the speed of the vehicles in order to obtain the smallest possible distance between the vehicles in order to reduce the crossing times.

[5] Source: A. Mirheli, M. Tajalli, L. Hajibabai and A. Hajbabaie, "A consensus-based distributed trajectory control in a signal-free intersection," *Elsevier,* 2019

1.2.8 On the V2X speed synchronization at intersections: Rule based System for extended virtual platooning

The authors in [13] proposed the synchronization of the vehicles for the platoon crossing. The system is based on V2X communications and requires all vehicles to have the list of vehicles involved with their parameters.

In this paper they consider that each vehicle receives a presence list of all vehicles. This is an extend list built by the geonetworking standard. In the presence list, the vehicle finds all other precedent vehicles, their position, speed and desired destination, i.e. go straight, turn right and turn left.

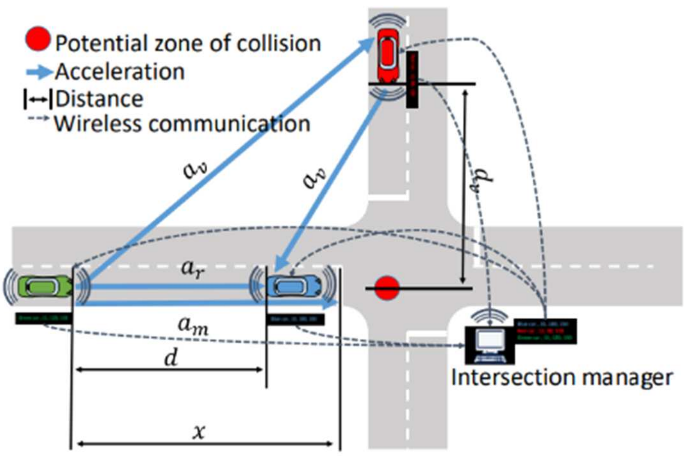


Figure 11 Speed synchronization at intersection[6]

As shown in the Figure, the blue car is first, red car is second and green car is the last one. Green car must consider both red and blue cars to adapt its speed. The intersection manager periodically broadcasts the updated list according to the periodically received messages from cars.

The presence list can be built either by each vehicle through V2V communication or by the intersection manager.

The approach uses the FCFS system and provides for a platoon crossing for all vehicles that do not have conflicts with the vehicles queuing them in the list.

1.2.9 Proposed Solution

Our starting point is based on a V2V approach to overcome the large initial implementation limit that would require an intersection manager approach instead. Our research uses a FRFP protocol

[6] Source: 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" *Elsevier,* 2018

that allows a more natural management of trajectories over time. The research was born from the objective of implementing an innovative algorithm that can guarantee better efficiency in the management of the road intersection in general. The proposed System has been conceived starting from what man tries to do at an intersection before a traffic light system. In fact, in this case man tries to understand if the vehicle he is driving has the capacity, according to its speed and the distance from the intersection in relation to the estimated speed of the other vehicles involved, to cross the intersection avoiding the collision with the other vehicles. In case it should be necessary, the driver will also estimate a possible acceleration to overcome the intersection, thus avoiding having to decelerate and wait for other vehicles, clearly this possible choice is usually made according to the pressure and possible expenditure of petrol needed. Let us imagine now that all this can be done directly by the vehicle having at its disposal much more data and agreeing with the other vehicles involved. Eventually deciding whether to apply a fuel-saving driving, therefore applying the lowest possible accelerations/decelerations, or time saving. In the future, it may also be possible to give priority in return for money.

The system will be compared with different systems at the state of the art and in different conditions and in different types of intersections, including roundabouts. We will see how the proposed system minimizes the changes in the parameters of the vehicles involved so it is more effective in terms of consumption, emissions, wear and time of crossing.

The book is structured as follows: Section I Introduces the work by defining its context and objectives, describes the state of the art on the topics covered and then focuses on the motivations of the research carried out. Section II SUMO: describes the used tools. Section III and IV System description: Our system is described in detail. Section V: FRFP Protocol using blockchain. Section VI Results: The results obtained are presented. Section VII: Conclusions and Future work: the conclusions are described, and possible future works are exposed. Section VIII: Bibliography.

Chapter 2

SUMO (Simulation of Urban Mobility)

In the initial phase the most important choice was the most suitable simulator to test the implemented algorithm and to compare it with the algorithms already known at the state of the art. Several simulators were evaluated and the hypothesis of developing one's own simulator was initially also considered.

After a careful period of analysis and evaluation, it was decided to use one of the most important and popular simulation software for vehicular mobility: SUMO (Simulation of Urban Mobility). The reasons that led to use this platform are different. SUMO has a library of functions that allows us a complete and simple control of vehicles, you can create any type of intersection and, above all, it allows us to evaluate in a simple and fast way many parameters such as emissions, routing times, fuel consumption. We have set ourselves the goal of minimising intersection times, CO2 emissions and fuel consumption. It is of fundamental importance to avoid and dispose of road congestion caused by traffic and/or other situations such as accidents.

The first version of SUMO, totally open source, was released in 2002 by the German Aerospace Centre (DLR). The choice to realize an open source software has allowed a constant evolution of the software that now is not only a traffic simulator but also a set of applications able to prepare and manage the simulation [14].

Basic concepts

SUMO is designed to simulate a road network from a single 2-lane junction to a complex road network the size of a large city. This simulation allows to model not only cars but also other categories of vehicles such as, for example, public transport systems, rail networks, motorcycles and so on. In addition, vehicles with different performance and size characteristics can be simulated at the same time. The movement of a vehicle or an individual is described by an instant of departure and by a path that it will go to complete; this path is in turn composed of sub-paths that describe the individual modes of movement. The vehicle will move with certain characteristics in terms of speed, acceleration and following the desired path. All this will be managed through the simulator by the algorithms tested.

Traffic flows are microscopically simulated, i.e. each vehicle within the network is individually modelled and has its own position and instantaneous speed. At each step of the simulation, which lasts one second, these values are updated according to the previous vehicle with respect to the direction of travel and the type of road in which the vehicle is moving. The simulation of the vehicles is a discrete time and continuous space. During the simulation, the attributes that characterize a certain road, such as the direction of travel and the speed limit, are respected by the vehicles that run it.

The only limit for our simulation is the single step of the simulation that has duration equal to one second; such limit could turn out tight for the narrow times of elaboration demanded in correspondence of a road intersection. The result has been however very positive guaranteeing therefore optimal performances in the real case where the times of elaboration can be notably reduced regarding those of the simulator. At each step of the simulation the speed of a vehicle is adapted according to the algorithm tested and according to the speed of the vehicle that precedes it with a minimum guaranteed gap, this produces a system capable of avoiding collisions in the next step. A maximum speed is also set both for each vehicle and as a limit speed on a given route. To perform a simulation in SUMO it is first necessary to define the scenario in which the vehicles can circulate or the road network including lanes, traffic lights, intersections and other structures useful for the representation of reality, then you must create the mobility you need to simulate defining the types of vehicles, the possible routes and the list of vehicles that will take part in the simulation.

Creation of road networks

For the creation of road networks you can use NETEDIT, a visual network editor, or even a system of automatic generation of real road networks through the platform https://www.openstreetmap.org which will then be converted through tools in SUMO. NETEDIT can be used to create networks from scratch and to modify all aspects of existing networks.

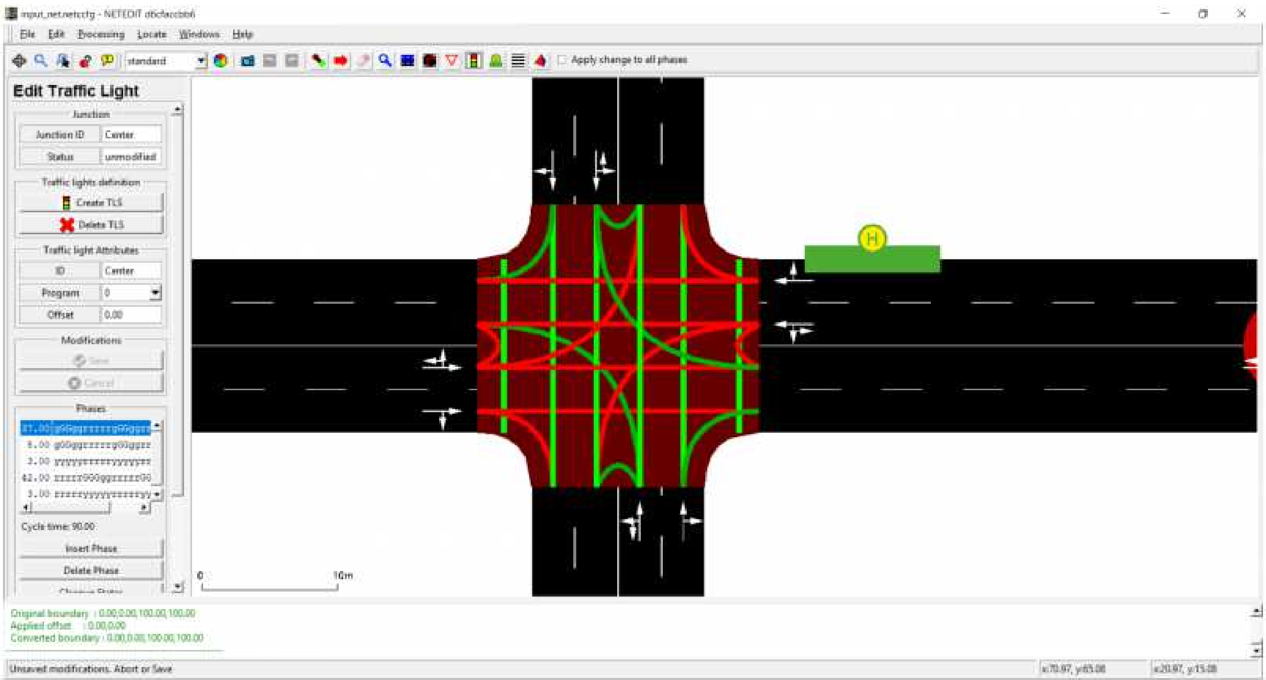


Figure 12 Schermata grafica NETEDIT Tool

A SUMO network file generated for example through NETEDIT has the format "name.net.xml" and describes the part of the map affected by the traffic.

These files then define the road network and all the information useful to address and manage traffic flows (junctions, lanes, traffic lights, etc.). The network files, being in XML format (eXtensible Markup Language), are readable directly by the user, however, given their complexity to the growth of the network represented, they are not designed to be modified manually; for this reason SUMO provides a set of applications that can create scenarios based on XML descriptions. These descriptions can be defined by the user through text editors or, preferably, imported from other sources with the help of applications, such as NETCONVERT, which allow the conversion of different formats into SUMO network files.

In addition to defining the basic structure of the road network, therefore the crossings and roads must also consider the presence of other information and structures that can complete the representation of the network such as: number of lanes for each road, each lane has its own position, its own size and its own speed limit, the presence of constraints related to the direction of travel, the connections between lanes at various intersections, the position and logic of the traffic lights present.

In addition, each road will have a very specific priority: at the intersection, vehicles occupying a lane with low priority should give precedence to vehicles with higher priority. Therefore, during the creation of the road network we will set all the necessary rules starting from the priorities, to the speed limits for each route up to the logic of the traffic light systems.

In our case, for the algorithm implemented, we have freed the vehicles from any priorities; in fact, from the point of view of vehicles with automatic guidance it makes no sense, in most cases, to give priority because this would limit the efficiency of the management of intersections. Instead, we have found that in the management of roundabouts the use of priorities is very effective. In fact, the only way to avoid congestion and effectively dispose of traffic is always to give priority to those who occupy the roundabout. Giving priority to vehicles that have to enter the roundabout would mean congestion.

When the structure of the road network has been defined, including all the characteristics described above, it is also possible to insert sensors that, during the simulation phase, will give us useful information for the management of the algorithm. In particular, in our system to simulate communication between vehicles we have inserted detection points to detect: the entry of the vehicle in the area near the intersection (communication area V2V at the intersection); and the exit of the vehicle from the critical area of the intersection. All this is managed directly under the NETEDIT application. As mentioned before, through the NETCONVERT module we can import road networks from different sources and generate networks directly usable by SUMO applications.

When a network has been generated, the SUMOGUI graphic interface can be used to visualize it; in this way, we will have the opportunity to explore the scenario created, containing all the communication routes, road signs and previously defined structures.

Generation of the road population

To complete the input necessary to start a simulation, it is necessary to generate and introduce into the road network the vehicles that will form the traffic flows. This is called "traffic demand" and is defined by files in ".rou.xml" format that, within them, specify: the types of vehicles, in particular their size, the minimum gap required with respect to the vehicle that precedes them, the characteristics in terms of speed, maximum accelerations and decelerations, the colour of the representation they will have in the simulation, the routes that vehicles can travel, the vehicles that will take part in the simulation. Within the ".rou.xml" files it is first necessary to define the categories of vehicles participating in the simulation. In this way it will be possible to classify the individual vehicles so that they reflect the macro-features of the type of belonging. The attributes describe the vehicles according to their physical characteristics such as, for example, acceleration, speed and length.

When all the characteristics of the vehicles involved have been defined, their route must be defined. You can define two types of route: trip and route. With trip we define the movement of a vehicle from one position to another in the scenario, this movement is defined by: a start identifier, a destination identifier and a start time. Instead, with route, the one we use, we define a very precise route, i.e. not only the departure and arrival roads are specified, but all the roads that the vehicle will have to cross. The file ".rou.xml" must contain within it the list of possible routes that can be followed by vehicles. Clearly, in order to generate a realistic simulation, the list of lanes that make up a route must have sequentially connected joints; if this is not the case, a vehicle along that route will be teleported from one joint of the map to another when the next joint is not connected to the previous one. The programming inside the simulator is done through high level programming languages. In particular we have generated the file ".rou.xml" through a script in python language where the paths, the number and all the characteristics of the vehicles are defined.

```
def generate_routefile():
    random.seed(100)  # make tests reproducible
    N = 1000  # number of time steps
    # demand per second from different directions
    pWE = 1. / 4
    pNS = 1. / 3

    with open("data/automatic.rou.xml", "w") as routes:
        print("""<routes>
        <vType id="typeWE" accel="4" decel="3" length="5" minGap="1" maxSpeed="13" guiShape="passenger"/>
       <vType id="typeNS" accel="4" decel="3" length="5" minGap="1" maxSpeed="13" guiShape="passenger"/>

        <route id="right" edges="gne_sx gnesx1 gnedx1 gne_dx" />
        <route id="down" edges="gne_nord gnenord1 gne_sud" />""", file=routes)
        lastVeh = 0
        vehNr = 0
        for i in range(N):
            if random.uniform(0, 1) < pWE:
                print('    <vehicle id="right_%i" type="typeWE" route="right" depart="%i" color="0,1,0"/>' % (vehNr, i), file=routes)
                vehNr += 1
                lastVeh = i

            if random.uniform(0, 1) < pNS:
                print('    <vehicle id="down_%i" type="typeNS" route="down" depart="%i" color="1,0,0"/>' % (vehNr, i), file=routes)
                vehNr += 1
                lastVeh = i

        print("</routes>", file=routes)
```

Figure 13 Script Python to generate .rou.xml file

Within the script shown in *Figure 13 Script Python to generate .rou.xml file* all the elements for generating the file that defines the traffic demand are defined.

```
<routes>
            <vType id="typeWE" accel="4" decel="3" length="5" minGap="1" maxSpeed="13" guiShape="passenger"/>
             <vType id="typeNS" accel="4" decel="3" length="5" minGap="1" maxSpeed="13" guiShape="passenger"/>

      <route id="right" edges="gne_sx gnesx1 gnedx1 gne_dx" />
      <route id="down" edges="gne_nord gnenord1 gne_sud" />
   <vehicle id="right_0" type="typeWE" route="right" depart="0" color="0,1,0"/>
   <vehicle id="right_1" type="typeWE" route="right" depart="4" color="0,1,0"/>
   <vehicle id="right_2" type="typeWE" route="right" depart="5" color="0,1,0"/>
   <vehicle id="right_3" type="typeWE" route="right" depart="7" color="0,1,0"/>
   <vehicle id="right_4" type="typeWE" route="right" depart="8" color="0,1,0"/>
   <vehicle id="down_5" type="typeNS" route="down" depart="8" color="1,0,0"/>
   <vehicle id="right_6" type="typeWE" route="right" depart="9" color="0,1,0"/>
   ................................
   ................................
</routes>
```

Figure 14 Example of file.rou.xml

Simulation

After that the phases of definition of the input have been completed, therefore having at disposal both a network on which to carry out the simulation and the mobility demand, it is possible to proceed with the real simulation. The simulation is composed therefore of the following phases:

1. Reading the road network .net.xml;
2. Reading the ".det.xml" file containing any detection points
3. Opening the ".rou.xml" file and reading the first n steps;
4. Reading of any additional files for the generation of statistics;
5. Execution of the simulation with loading of the script of the algorithm implemented for n step.

The simulation will start in step 0 and will end when the last active vehicle has finished its journey. The simulation is performed step-by-step, where each step represents, by default, a second of the simulated scenario. The simulation can be started from the terminal using the commands:

```
sumo -c config.sumocfg
sumo-gui -c config.sumocfg
```

It is possible to visualize graphically the course of the simulation using the second command. Inside the configuration file (config.sumocfg) the various input files, described in the previous paragraphs, are specified according to the following structure:

```
…..
  <input>
    <net-file value="automatic.net.xml"/>
    <route-files value="automatic.rou.xml"/>
    <additional-files value="automatic.emission.xml, automatic.statistic.xml, automatic.det.xml"/>
  </input>
  <time>
    <begin value="0"/>
  </time>
  <report>
    <verbose value="true"/>
    <no-step-log value="true"/>
  </report>
</configuration>
```

In our case the command is directly inserted in the python script of the implemented algorithm. The SUMO application provides a fairly complete library of functions for vehicle management and simulation in general. TraCI is the short term for " **Tra**ffic **C**ontrol **I**nterface ". Giving access to a running road traffic simulation, it allows to retrieve values of simulated objects and to manipulate their behaviour "on-line". To do this, it uses a TCP (Transmission Control Protocol) client/server architecture in which SUMO plays the role of server and other external applications play the role of client. The client can give commands to control the execution of the simulation, to influence the behaviour of individual vehicles or to ask the server for details regarding the simulated environment. SUMO will respond to each of these commands with response status and additional information depending on the type of command.

Figure 15 Example of how to use the TRACI library

```
vehicle_nord_out = traci.inductionloop.getLastStepVehicleIDs("P3")
vehicle_sx_out = traci.inductionloop.getLastStepVehicleIDs("S3")

#LIST FCFS
#delete outgoing vehicles
dim_fcfs = range(0, len(list_fcfs))
if len(list_fcfs) >= 1:
        if len(vehicle_sx_out) >= 1:
                dim_sx = range(0, len(vehicle_sx_out))
                for i in dim_sx:
                        dim_fcfs = range(0, len (list_fcfs))
                        if len(list_fcfs) >= 1:
                                index_rm = 0
                                for j in dim_fcfs:
                                        if vehicle_sx_out [i] in list_fcfs[j-index_rm]:
                                                del list_fcfs[j]
                                                index_rm = index_rm + 1
                dim_fcfs = range (0, len(list_fcfs))
                if len (vehicle_nord_out) >= 1:
                        dim_nord = range (0, len (vehicle_nord_out))
                        for i in dim_nord:
                                dim_fcfs = range (0, len(list_fcfs))
                                if len (list_fcfs) >= 1:
                                        index_rm = 0
                                        for j in dim_fcfs:
                                                if vehicle_nord_out [i] in list_fcfs[j-index_rm]:
                                                        del list_fcfs[j]
                                                        index_rm = index_rm + 1
        print(list_fcfs)

#check congestion pre-crossing
        if ((traci.edge.getLastStepOccupancy("gnesx1"))>(traci.edge.getLastStepOccupancy("gnenord1")+0.2) or …
                state_congestion = 1
        else:
```

Chapter 3

Developed algorithm: FRFP

The idea developed is based on the principle that whoever has the potential to arrive first at the end of the intersection must be the one who has the priority to cross the intersection. This principle is believed to be very effective as the speed variations of the vehicles involved are certainly lower than other systems such as the FCFS system [4]- [5]. Just think of the case in which a vehicle, at a greater distance from the intersection than another, has a much higher speed than the one closest to the intersection. If the time required for the arrival at the end of the intersection of the fastest vehicle (t_1) is less than the time required by the other vehicle (t_2), priority will be given to the vehicle at a greater distance.

In the most optimistic case the fastest vehicle could pass the crossing even without requiring any reduction in the speed of the other vehicle involved in crossing the intersection; it would be very unnatural and certainly expensive to give priority to a very slow vehicle by imposing a strong reduction in the speed of the other vehicle. Therefore, it is believed that the system adopted can have significant advantages in terms of reducing the average crossing time, reducing fuel consumption, reducing vehicle wear and reducing CO2 emissions.

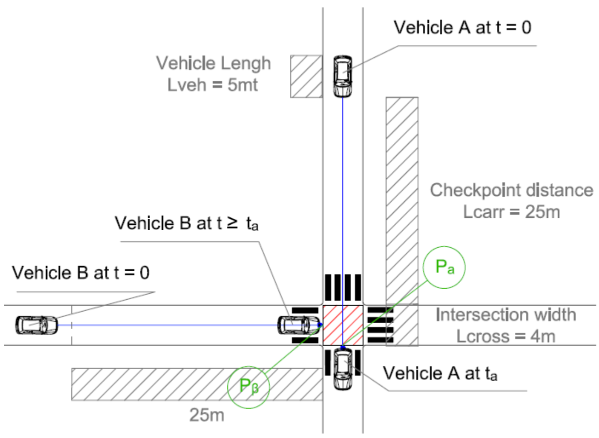


Figure 16 Intersection management with only two vehicles

We will then start to consider the simplest case of intersection, i.e. only 2 vehicles involved that can cross a single lane incident without being able to turn. The vehicle near the intersection communicates its position [S_A], its speed [v_A] and therefore the estimated arrival time at point P_A [t_A]. The vehicle that can arrive in less time has the priority. The vehicle with less priority will notice that it will pass through the intersection on second place and it will calculate the deceleration [a_B] that it will have to maintain.

The speed variation to be applied in this case is optimized with advantages for both fuel consumption and car wear. On the contrary, it would be more disadvantageous to apply a speed variation to that car with the highest priority. Considering, for example, the extreme case of a car travelling at 100km/h and another car travelling at 20km/h. If we consider that the fastest vehicle has priority, a slight change in slow vehicle speed is certainly more advantageous than a sharp change in the speed of the fastest vehicle.

The vehicle with less priority must arrive at the beginning of the crossing at the same time or after the exit of the priority vehicle from the intersection (point P_A). In this case, as in the rest of our research, to ensure maximum security we are setting the entire intersection area as an area that will have to be occupied by only one vehicle at a time. The same system can be used by dividing the intersection area into cells. In this case we would increase the efficiency of the system, but we would inevitably reduce the safety margins. In this case only the area occupied by the vehicles with a small safety margin will have to be occupied only by one vehicle at a time. Vehicle A communicates with its cadence [S_A, v_A, t_A] in the course of a path, so that vehicle B can constantly check that the calculation made at start does not need to be modified. Vehicle A, assuming it moves at constant speed, will arrive at point P_A at the time

$$t_A = \frac{S_A}{v_A} \; ; \tag{1}$$

where $\mathbf{S_A}$ is the distance between the vehicle and the P_A point (assuming the distance of the checkpoint from the intersection of 25m [L_{carr}], the width of the intersection equal to 4m [L_{cross}] and the length of the vehicle equal to 5m L_{veh})

where $\boldsymbol{v_A}$ is the speed of the vehicle A

$$t_A = \frac{S_A}{v_A} = \frac{L_{carr} + L_{cross} + L_{veh}}{v_A} \tag{2}$$

From equation (2), in the present case $\mathbf{t_A = 8.5s}$ [we are assuming a constant speed of 4m/s]. If Vehicle B can arrive to the end of the intersection in less time it will be the priority vehicle and will cross the intersection first. If the Vehicle B doesn't have the priority it crosses the intersection on the second place. Vehicle B will have to reach P_B point after a time greater than or equal to t_A. Applying formula 4 the vehicle will be able to calculate the deceleration necessary to cross the intersection safely.

$$S = \frac{1}{2} a\, t^2 + \; v_0\, t + S_0 \tag{3}$$

In equation (3), time t will be equal to t_A, space S will be equal to the distance between vehicle B and P_B point, speed V_0 will be equal to vehicle B speed (assumed constant), S_0 is the assumed starting point. Thus, obtaining the following reverse formula for a it yields:

$$a = 2 * \frac{S_B - v_B\, t_A}{{t_A}^2} = 2 * \frac{S_B - v_B\, \frac{S_A}{v_A}}{\left(\frac{S_A}{v_A}\right)^2} \tag{4}$$

The speed limit and the distance of the communication must guarantee a feasible acceleration or deceleration. Assuming the constant speed V_B equal to 3.8m/s and replacing the values in the example:

$$a = 2 * \frac{26.5 - 3.8 * 8.5}{(8.5)^2} = 0.1605 m/s^2 \tag{5}$$

In this case, the vehicle B can accelerate to get to P_B point at the same time as the arrival of the vehicle A to the point P_A. If the value of a is positive the vehicle B ,as obtained in formula (5), can decide to accelerate to increase its speed and reduce the mean travel time. The decision may depend on the driving mode chosen and, in any case, it should be known by the other vehicles. It will be communicated together with the parameters. If the value of a is negative, the vehicle B must apply the required deceleration. The accelerations or decelerations to be applied will be the lesser abrupt as the greater the communication distance will be. Now suppose we have a more complex situation, i.e. an intersection with 2 lanes but with many more vehicles involved. In this case, each vehicle, through V2V communication, will communicate its parameters to the other vehicles involved in the intersection. Then each of them will create a priority list based on the arrival time of each vehicle $t_x = \frac{S_x}{v_x}$. Thus, assuming the zero-error communication system, each vehicle will know exactly the priority list. The list of priorities will be determined considering that if the system provides only one lane per direction, the vehicles of the same lane will necessarily have to respect the sequence of their positions.

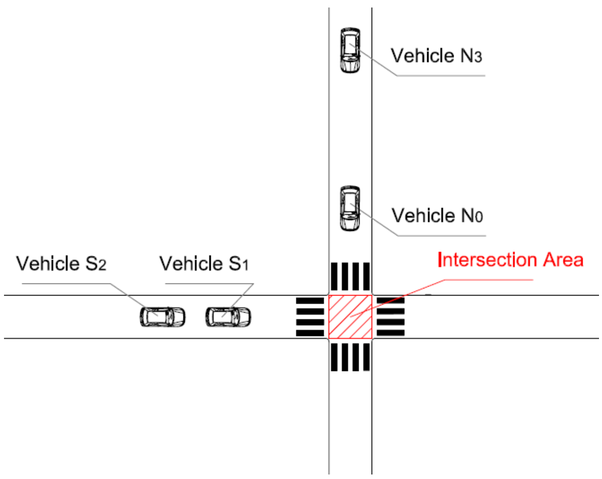


Figure 17 Intersection management 2 lanes intersection

In *Figure 17 Intersection management 2 lanes intersection*, for example, the priority will be the following [N_0, S_1, S_2, N_3, …] so the vehicle S_1 will adapt its speed according to the vehicle N_0, the vehicle S_2 will adapt its speed only to guarantee the minimum distance of safety from S_1, the vehicle N_3 will adjust its speed so as to arrive at the intersection at the moment when S_2 will arrive at the end of the intersection and so on. Therefore, in the list of priorities the vehicles will distinguish the vehicle with conflict (coming from other lane) from those with no conflict (on the same lane). The calculations described so far do not take into account two fundamental factors: maximum speed on the lane and maximum acceleration of each vehicle. In the implemented system the priority calculation is performed not considering the vehicle speed but its potential speed ($v = v_0 + a\,\Delta t$). Clearly, we must consider the speed limit and the performance in terms of maximum acceleration of the vehicle. In this particular case, we might think that the vehicle can be set in different driving modes. A vehicle set in Eco drive will not want to apply high acceleration with the goal, for example, to reduce fuel consumption. In this case, the maximum acceleration desired by the vehicle will be taken into account in the calculations.

The time to calculating the priority and the acceleration / deceleration to be applied cannot still be simply the one considered until now $t_x = \frac{S_x}{v_x}$. We must take into account various factors such as the maximum acceleration that the vehicle intends to apply and the maximum speed required for the lane in question.

The procedure for calculating the priority for "regulation status" follows:
- Calculate the distance of vehicle list[i] to the end of the intersection [Dist]
- Survey the speed [Vel]

Determine priority as a time needed to carry out the vehicle from intersection.
- Apply maximum acceleration until maximum speed is reached
- If the required distance has been travelled without reaching maximum speed, calculate the time needed, which also corresponds to the priority, between the inverse formula of (3)

$$S = \frac{1}{2} a\, t^2 + v_0\, t + S_0$$

Where *S* is the distance to be made, *a* the maximum applicable acceleration, v_0 is the initial speed

$$t = \frac{-v_0 + \sqrt{{v_0}^2 - 2a_{max}S}}{a} \quad (6)$$

- If the distance to be made is higher than the maximum distance in acceleration, calculate the necessary time, which corresponds also to priority, between the sum of t_{max} and t

Where

$$t_{max} = \frac{v_{max} - v_0}{a_{max}} \tag{7}$$

from the inverse formula of

$$v_{max} = v_0 + at \tag{8}$$

We get

$$t = \frac{Dist - Dist\_max}{v_{max}} \tag{9}$$

Dist_max is the distance leaving after the time t_{max}.

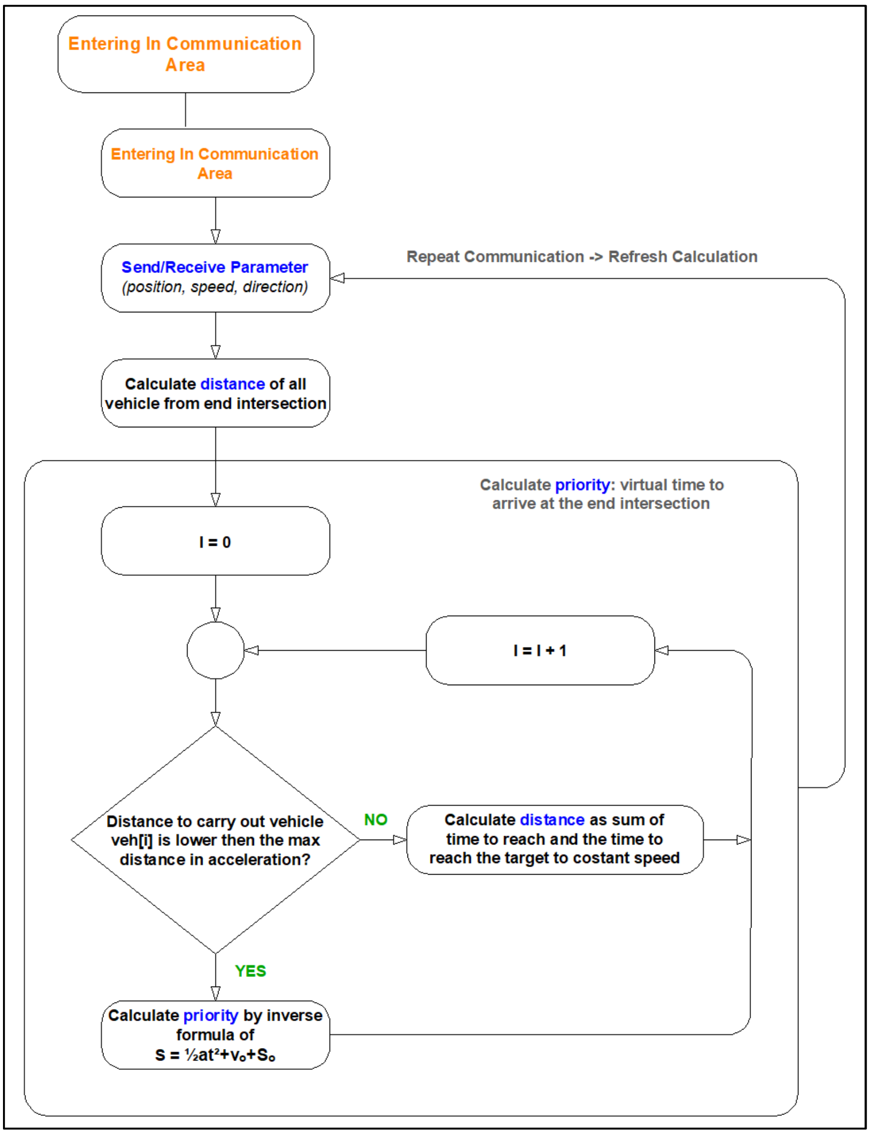


Figure 18 Block Diagram for priority calculation.

When calculating the priority, it must also be taken into account that regardless of speed, vehicles occupying the same lane must have sequential priority. The vehicle closest to the intersection will still have priority over the vehicles that occupy the same lane (except in the case where there are more lanes dedicated to overtaking). In this case, in calculating the priorities of the vehicle x we will not use the velocity v_x but we will use the speed of the vehicle that precedes it v_{x-1}. At this point we order the list of vehicles from two or more lanes according to the list of calculated priorities. We provide an example of priority list: [north_1, right_1, right_2, north_2, right_3,north_3,….]. Calculated vehicle priorities will apply the speed adjustment function by calculating the acceleration / deceleration to be applied for each of them. Then we apply *function_decel* between first vehicle and second one, after the second and third, and so on. If the vehicles are in the same lanes, we do not modulate any of them. Other modules are as specified following: the $vehicle_{[0]}$ will accelerate to the maximum value, the $vehicle_{[1]}$ modules the speed that will be reached at the start of the crossing after the $vehicle_{[0]}$ will be outside and so on for the other vehicles $_{[i]}$ and $_{[i+1]}$. In this function, the time needed by the priority vehicle to exit from intersection is calculated:

$$t = Dist/speed \tag{10}$$

For safety we consider that the vehicle is moving at the speed communicated without applying any acceleration. Then we calculate the acceleration with the following form:

$$a = 2 * \frac{S - v_0 t}{t^2} \tag{11}$$

Where *S,* in formula (11) , is the distance from the start of the intersection an v_0 is the speed of the non-priority vehicle, when the time *t* is calculated by the previous form (time so that priority vehicle can reach the end of crossing). The system described up to this point, although it is very efficient, presents a critical point. Let's assume we have a lot of high speed Sx vehicles and only one very low speed vehicle Nx at the intersection. In this limit condition we may find ourselves never to witness the crossing of the Nx vehicle. To deal with these situations in conjunction with, in general, situations of particular intersection congestion, we have introduced three different work states. The one seen so far represents the "Regulation state" and manages the intersection as long as there are no blocking or intersection congestion situations. In case of congestion and / or self-intersection blocking we will talk about "Balance state". In case of post-intersection congestion, for example due to a vehicle failure, we will talk about "Freeze state".

The "Balance State" status is introduced both to eliminate the previously presented case and to more effectively manage high congestion characterized by many low-speed vehicles. The last state "Freeze state" provides that since there is a post-intersection block all vehicles in that direction will remain in the pre-intersection lane giving priority to vehicles with free lanes. The state "Balance state" was initially implemented with a platoon algorithm, thus foreseeing the passage of vehicles no longer according to the list of priorities according to the "regulation state" but balancing the crossing of a number of vehicles per lane depending on the percentage of vehicle presence on the lane interested. For example, if we have 10 vehicles involved in a lane and 5 vehicles in the other, double the number of vehicles in the lane that is more congested, then the smaller group will pass. For example, two vehicles N and four S vehicles. From the simulations carried out, this type of approach was less efficient than the FCFS system, which was then adopted for the Balance status. Conditions that imply the state of balance clearly in an evolution of the system may differ depending on the type of intersection. In our implementations we have imposed a percentage threshold of the presence of vehicles on one lane compared to the others involved.

Balance State: when we have a congestion or when the gap between the vehicles blocks the passage from the other lane.

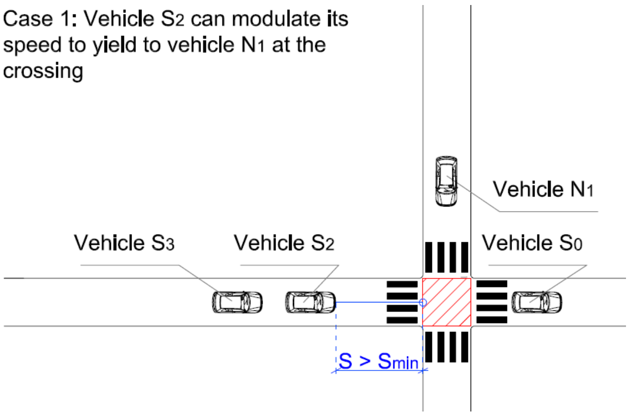


Figure 19 Regulation State case 1

If the vehicle N_1 is stopped and it must restart to reach the end of the crossing, we then assume that the initial speed [v_i] is 0 and a maximum acceleration a. The time necessary to cross (t_c) is:

$$S = \frac{1}{2}at^2 + v_i t; \qquad t_c = \sqrt{\frac{2S}{a}} \tag{12}$$

We can consider two different scenarios:

1- *Figure 19 Regulation State* case 1 The vehicle S_2 has a long gap with the vehicle S_0 and has the time t_i to permit the crossing of the vehicle N_1; in this case *ti*, the time required by vehicle S_2 to arrive at the beginning of the intersection must be greater than or equal to t_c, the time required by vehicle N_1 to reach the end of the intersection.

2- *Figure 20 Balance State case* 2 The vehicle S_2 has a short gap with the vehicle S_0 and the priority may never be released to the vehicle N_1. The necessary minimum time to crossing the intersection of vehicle N_1 is given by following inverse formula:

$$S_1 = \frac{1}{2}a{t_{cmin}}^2 + v_i t_{cmin} \tag{13}$$

Where *a* is the maximum acceleration applicable, S_1 is the distance of vehicle N_1 from the end of the intersection, t_{cmin} is the minimum time required from vehicle[1] to reach the end intersection. The real solution of the follow formula is the minimum time required to guarantee the crossing the intersection of vehicle N_1

$$t_{cmin} = \frac{-v_i \pm \sqrt{{v_i}^2 + 2aS_1}}{a} \tag{14}$$

The maximum time required by vehicle S_2 to arrive at the beginning of the intersection is given by the following inverse formula:

$$S_2 = \frac{1}{2}a{t_{max}}^2 + v_i t_{max} \tag{15}$$

Where *a* is the maximum deceleration applicable, S_2 is the distance of vehicle N_1 from the beginning of the intersection, t_{max} is the maximum time required by vehicle S_2 to reach the beginning of the intersection. The real solution of the follow formula is the maximum time required to reach the beginning of the intersection of vehicle S_2

$$t_{max} = \frac{-v_i \pm \sqrt{{v_i}^2 + 2aS_2}}{a} \tag{16}$$

If t_{max} will be greater than t_{cmin} the vehicle N_1 will be able to have priority and therefore will be able to cross the intersection; otherwise the vehicle N_1 may never have priority. To guarantee the release of the priority, the minimum distance of vehicles from the beginning of the intersection must be over:

$$S_{min} = \frac{1}{2}a{t_c}^2 + v_i t_c \tag{17}$$

Where t_c is the time requested to vehicle N_1 to reach the end of intersection.

To manage case 2 we'll change the status from regulation in balance as shown in the following Figure 20 *Balance State case 2*.

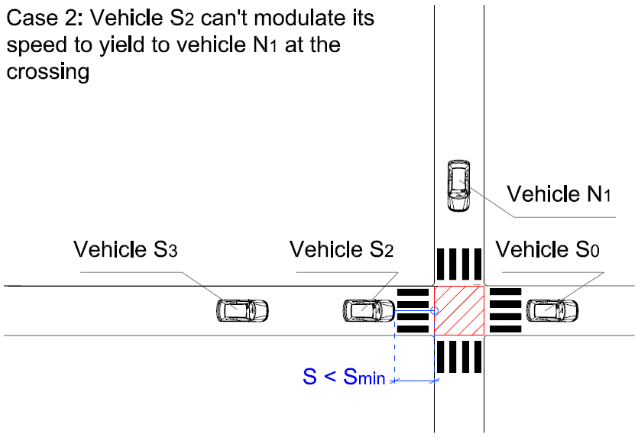


Figure 20 Balance State case 2

The basic concepts remain almost identical but with the appropriate considerations in intersections of different types. The system has also been applied to on-ramp systems, 8 lanes intersections and roundabouts.

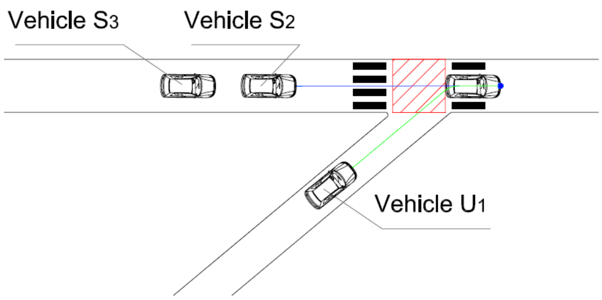


Figure 21 On-ramp intersection

Figure 21 On-ramp intersection shows the case of the on-ramp intersection. In this case, the vehicle trajectories must be considered until they arrive at the end of the intersection. In general, vehicles coming from the low lane must travel a greater distance to reach the end of the crossing. Therefore, in the formulas previously presented, the relative distances between the vehicles and the end of the intersection must be entered considering the trajectories that the vehicles will carry out. In the systems applied to 8 lanes intersections, the algorithm has been applied considering also the possibility of vehicles to turn at the intersection. In this case, obviously, the different distance of travel must be considered to exit the intersection zone with respect to a vehicle that continues straight.

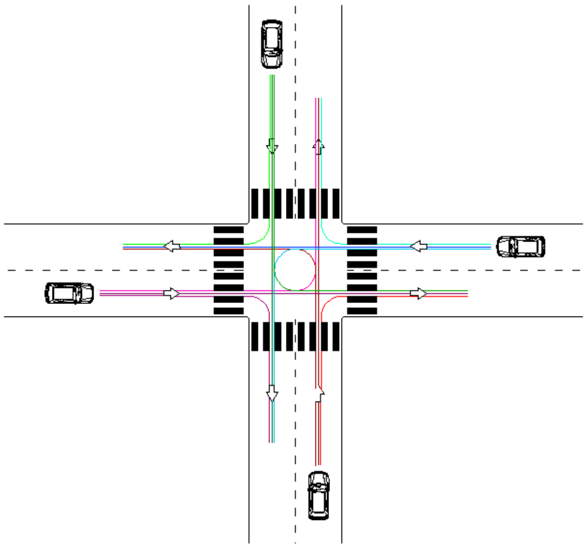


Figure 22 All possible vehicle trajectories

In this type of crossing each vehicle can travel 3 different trajectories. Thus, in total there will be 12 different possible trajectories. For each possible trajectory it is necessary to identify the trajectories without any collisions between them and therefore the critical ones with possible collisions that must be managed by the vehicle speed control. The system determines the list of priorities by calculating the estimate of the time needed to reach the end of the intersection point. At this point the vehicle that holds the priority passes first and in cascade all adapt to the vehicle that precedes them. In the modulation of speed, the vehicles, to adapt to the competitors that precede them, will take into account the fact that the trajectories of the other vehicles may or may not be compatible with their route.

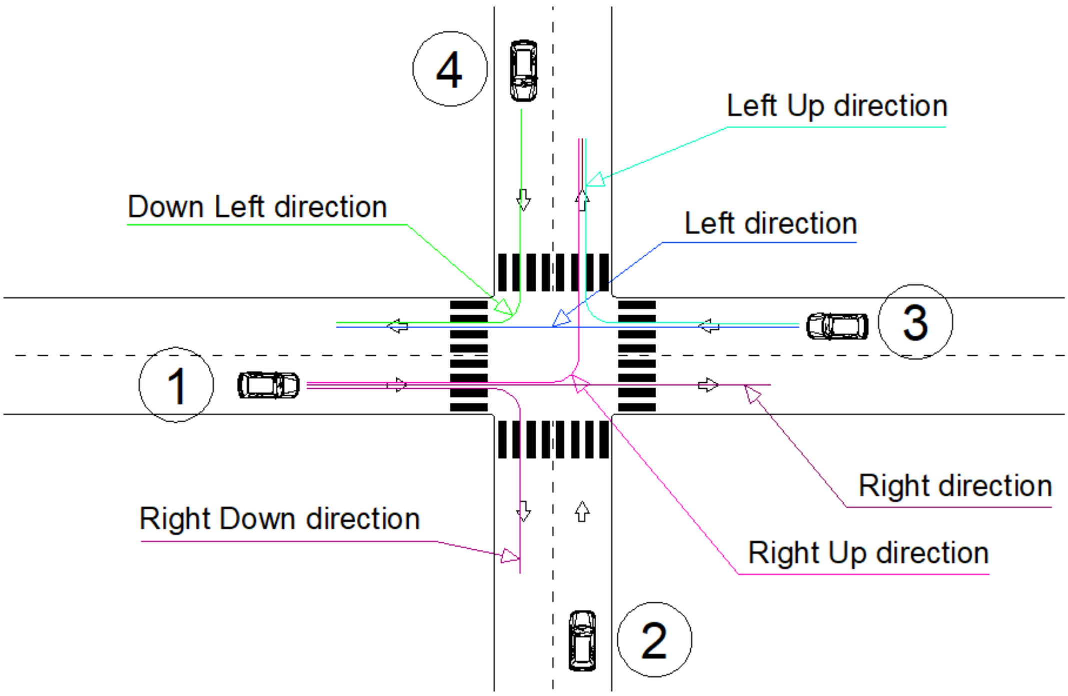


Figure 23 Trajectories without collisions with the cars in the right direction

.

Figure 23 Trajectories without collisions with the cars in the right direction shows the collision-free trajectories with vehicles in the right direction (vehicle 1), direction highlighted in violet. In this case the collision-free paths are those of the vehicles in the down_left (vehicle 4), left (vehicle 3), left_up (vehicle 3), right_down (vehicle 1) and right up (vehicle 1) direction.

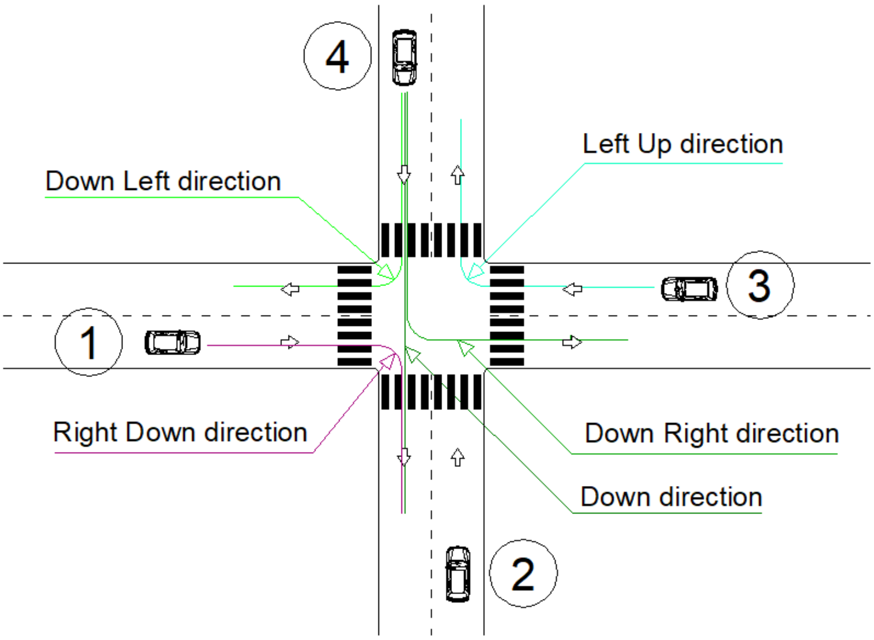


Figure 24 Trajectories without collisions with the cars in the down_right direction

.

Figure 24 Trajectories without collisions with the cars in the down_right direction shows the trajectories without collisions with the vehicles in the down-right direction (vehicle 4), the direction highlighted in green. In this case the collision-free paths are those of the vehicles in the down_left (vehicle 4), down (vehicle 4), left_up (vehicle 3) and right_down (vehicle 1) direction. Therefore, all possible collisions in the possible routes must be detected, in this case 12, and then manage the regulation state according to the fact that the two consecutive priority vehicles can have collisions or not. In the event of a possible collision we will adjust the speed of consecutive vehicles. Otherwise the vehicles will safely cross the intersection without any adjustment. *Figure 25 Block Diagram* follows the block diagram of the implemented system.

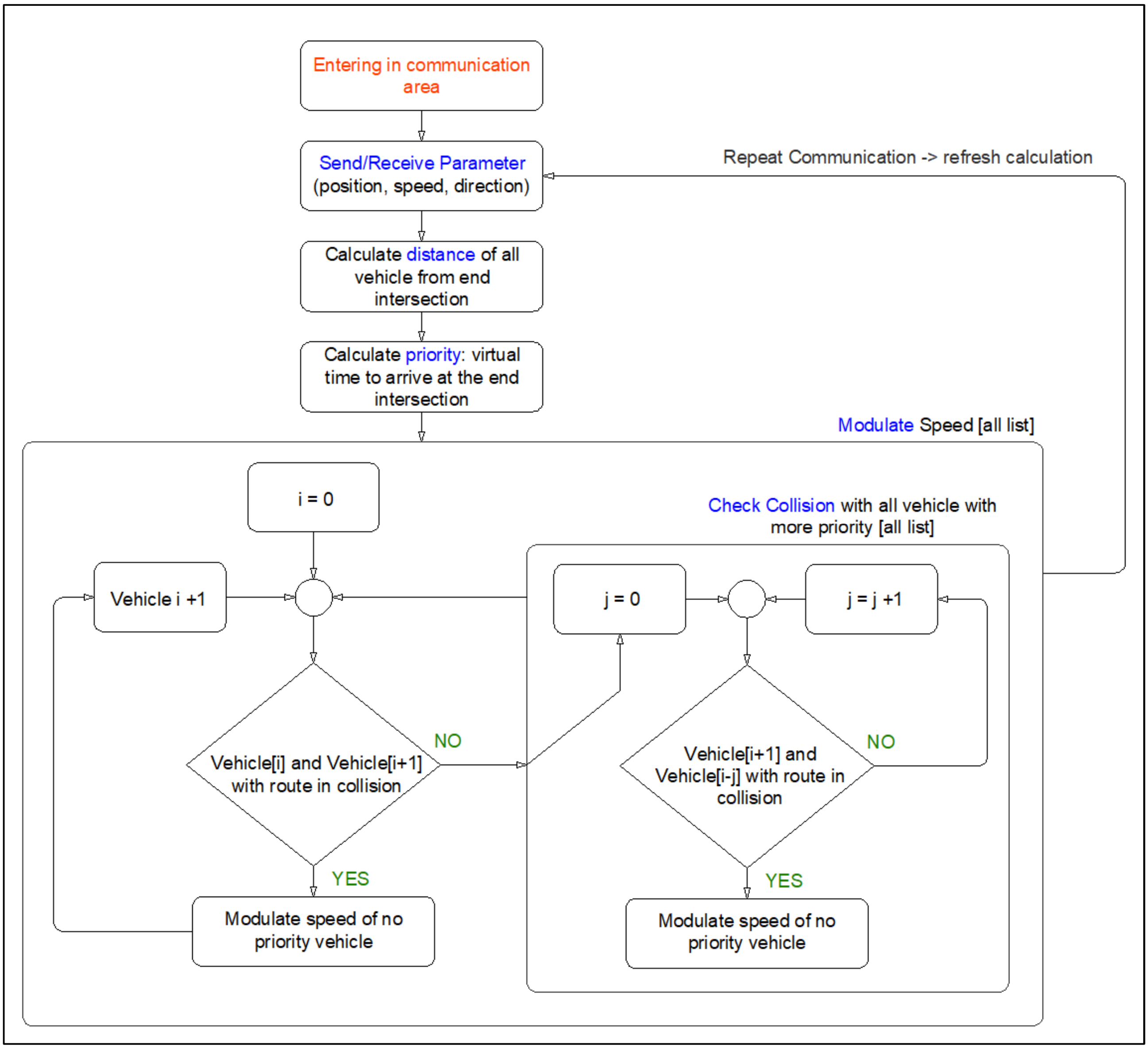


Figure 25 Block Diagram

The same system can be applied at roundabouts. In this case the system is complicated because basically we have a system composed of crossings; in the example in *Figure 26 Roundabout System* the roundabout has 4 intersections.

The principle remains the same, but in this case, we have to manage 4 different intersections. Referring to the figure, the vehicles of list 1 and list 8 will have to work together to form list 2. It will take into account both the presence of vehicles that will exit the roundabout before meeting the vehicles on list 1 and the lanes to be occupied according to the path that must be followed. The vehicles ready to exit the roundabout will occupy the most external lane, the others the internal one.

In the same way the vehicles of the list 2 will collaborate with those of the list 3 and so on. The optimization of the roundabout certainly has as main objective to dispose as quickly as possible the cars already inside itself. In fact, faster entry into disposal will lead to an inevitable saturation with a consequent slowing down of the vehicles involved. As we will see later at the simulations performed, the use of roundabouts is not suitable for optimizing travel time in the presence of driverless vehicles.

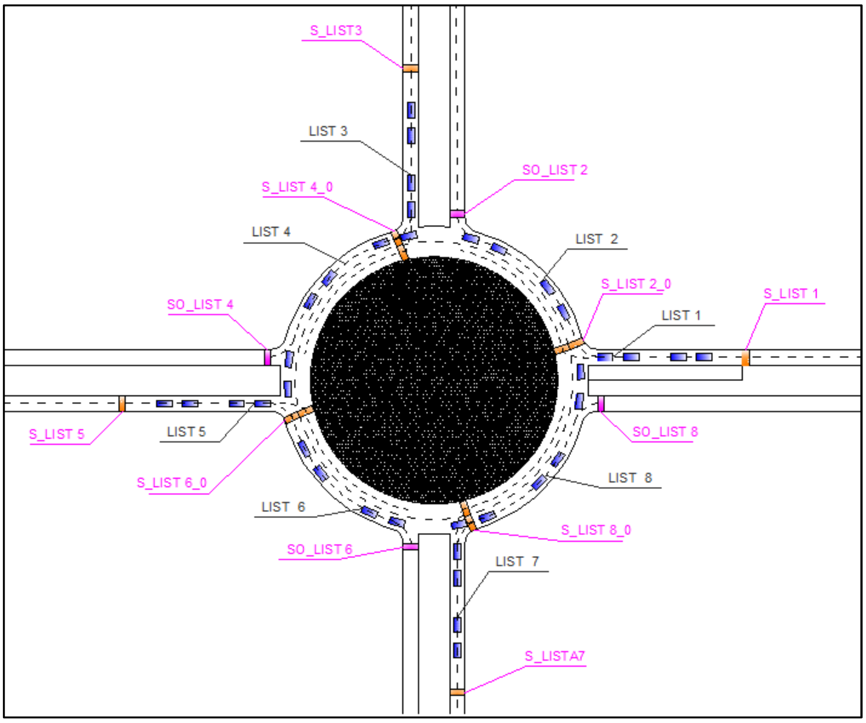


Figure 26 Roundabout System

Chapter 4

Implementation of the FRFP system

In this chapter we will see in detail what were the necessary steps to implement the proposed system and then the tools used for the development and final analysis of the system.

Creation of road networks

The first step was to implement the simplest road intersection through NETEDIT, a visual network editor, and then to use within SUMO the intersection created. The first intersection realized was the 2-lane one where at the intersection the vehicles can go straight or turn.

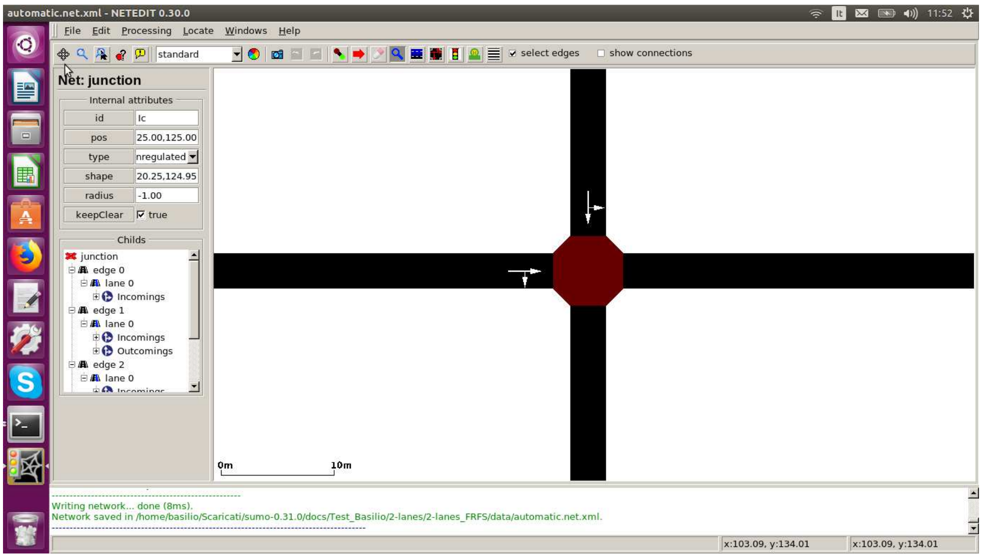


Figure 27 Intersection to two lanes (Netedit)

Following the development of the two-lane intersection, other increasingly complex types of intersections were implemented. In particular, the following types of intersections have been implemented: on ramp intersection, eight-lane intersection, roundabout.

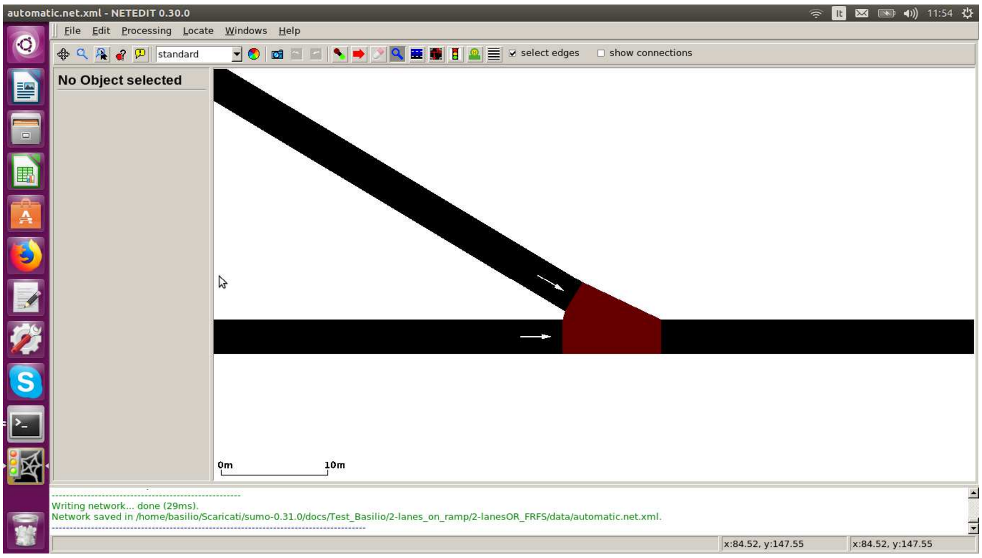


Figure 28 Intersection on ramp (Netedit)

The on-ramp type intersection has a management very similar to the 2 lanes intersection and is the simplest type of intersection.

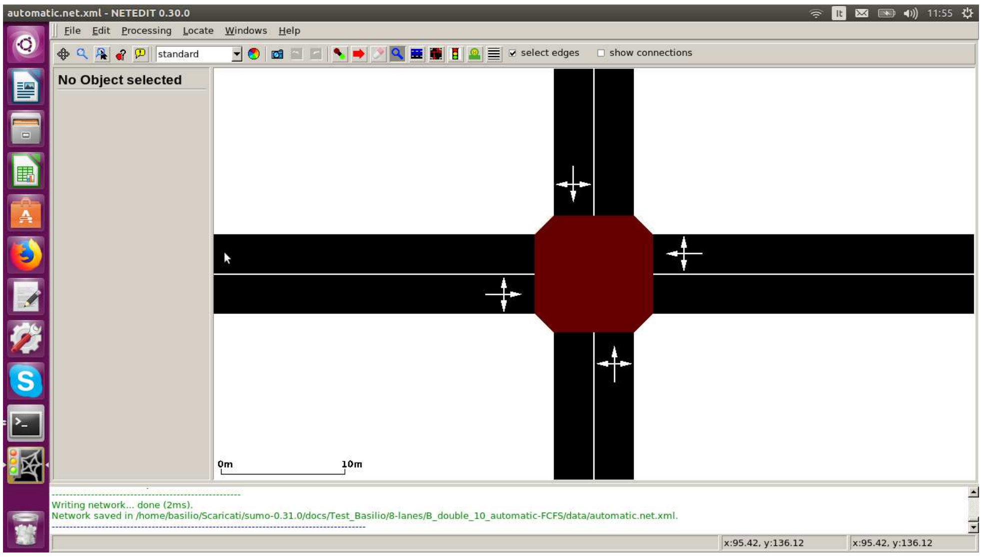


Figure 29 Intersection to 8 lanes (Netedit)

When the simplest intersections had been studied, the more complex intersection of 8 lanes was moved on. In this case the vehicles, at the intersection, can take any direction; in particular, each vehicle can go straight or rotate to the right or left. The population that will be created for this type of intersection will have the possibility to follow any lane present in a completely random way.

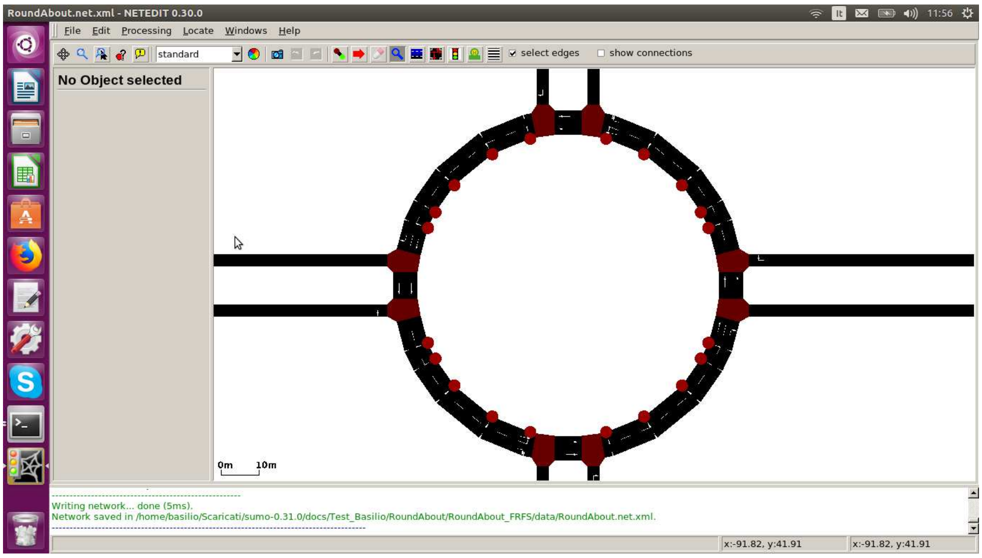


Figure 30 Roundabout system (Netedit)

The last type of road intersection analysed was the roundabout, a system that to date has been little analysed from the point of view of road vehicle management at this type of intersection. It will be of fundamental importance to understand which the best algorithms to use for this type of intersection are, and the comparison between these types of structures, now widely used for manual vehicles, and a classic eight-lane intersection.

Generation of the road population

After the implementation of the intersection, the population of vehicles and the algorithm were implemented using Python. In particular, we have generated the file ".rou.xml" through a script in python language where the paths, the number and all the characteristics of the vehicles are defined. Using Python scripts, we define for each type of vehicle the physical characteristics of the vehicles (dimensions), the parameters (maximum acceleration, maximum deceleration, maximum speed) and the route. The maximum distances between vehicles during the simulation are also defined. The maximum speed was set at 13m/s [46.8 km/h], the maximum acceleration was set at 4m/s^2 [14.4 km/h/s], and the maximum deceleration at 3m/s^2 [10.8 km/h/s], the minimum gap between vehicles was set at one meter. The routes generated were all those possible for each type of intersection, the route is set randomly. To make debugging easier, different colours are defined for the vehicles entered according to the different routes. In simulation each vehicle will be identified by a unique id. For each simulation related to the same type of intersection, different scripts were created to increase the population of vehicles at the intersections. In particular, situations with light traffic were simulated up to traffic situations at the limit of the capacity of the intersection itself. Limit capacity is the maximum level of intersection saturation.

```
def generate_routefile():
    random.seed(100)  # make tests reproducible
    N = 1000  # number of time steps
    # demand per second from different directions
    pWE = 1. / 4
    pNS = 1. / 3

    with open("data/automatic.rou.xml", "w") as routes:
        print("""<routes>
        <vType id="typeWE" accel="4" decel="3" length="5" minGap="1" maxSpeed="13" guiShape="passenger"/>
        <vType id="typeNS" accel="4" decel="3" length="5" minGap="1" maxSpeed="13" guiShape="passenger"/>
        <route id="right" edges="gne_sx gnesx1 gnedx1 gne_dx" />
        <route id="down" edges="gne_nord gnenord1 gne_sud" />""", file=routes)
        lastVeh = 0
        vehNr = 0

        for i in range(N):
            if random.uniform(0, 1) < pWE:
                print('    <vehicle id="right_%i" type="typeWE" route="right" depart="%i" color="0,1,0"/>' % (
                    vehNr, i), file=routes)
                vehNr += 1
                lastVeh = i

            if random.uniform(0, 1) < pNS:
                print('    <vehicle id="down_%i" type="typeNS" route="down" depart="%i" color="1,0,0"/>' % (
                    vehNr, i), file=routes)
                vehNr += 1
                lastVeh = i

        print("</routes>", file=routes)
```

Figure 31 Script Phyton to generate file.rou.xml

Developed algorithm

The development of the algorithm starts with the creation of the lists of vehicles present in the different lanes; this is done by inserting checkpoints in the lanes, generated with NETEDIT, for the detection of vehicles both at the entrance and at the exit from the intersection. In the actual situation the vehicles will communicate with each other via V2V communication or via an intersection operator (V2I) so each vehicle will know the presence of the other vehicles. SUMO does not allow us to simulate any type of communication between vehicles, so by adding the entry and exit checkpoints we emulate the communication between vehicles and the transfer of parameters. In particular, the vehicle will communicate its presence and all its parameters only after crossing the entry check point and will end its communication, giving the information that the vehicle no longer involved in the intersection, When the exit check point is crossed. The checkpoints are positioned at a distance of about 35 meters from the intersection in line with the real field of action of communications between vehicles equal to 50-100 meters from the intersection. When the list of vehicles involved in the intersection has been defined, the processing of the information to define the behaviour of the vehicles begins. Four different operating states have been defined: *Calculation*, *Regulation*, *Balance* and *Freeze*.

The first operating state "***Calculation***" is the state in which there are no vehicles, or the vehicles involved do not require any modulation of speed. For example, the vehicles involved all arrive from the same lane, so theroutes free of possible collisions or the vehicles have routes that do not contains any collision.

The operating state "***Regulation***" is the state in which the speed modulation of the vehicles according to the FRFP protocol takes place. Each vehicle has the list of vehicles involved in the intersection and therefore it is necessary to establish who holds the priority and in particular what will be the sequence of vehicles that will have to cross the intersection. To determine the sequence, each vehicle in the list will be assigned a value using the priority_list function.

```
def priority_list(_list_):
        priority_list_ = []
        dimensione_sx=range(0,len(_list_))
        if len (_list_) >=1:
                for i in dimensione_sx:
                Dist_temp = traci.vehicle.getPosition(_list_[i])
                        auto = _list_[i]
                        auto = auto[0:7]
                        if auto[0:3] == 'rig':
                                if (auto == 'right_u'):
                                        Dist = (127 - Dist_temp[0])+(133 - Dist_temp[1]) # x + y
                                elif (auto == 'right_d'):
                                        Dist = (123 - Dist_temp[0])+(Dist_temp[1] - 117) # x + y
                                else:
                                        Dist = (133 - Dist_temp[0])
                        if auto[0:3] == 'lef':
                                if (auto == 'left_up'):
                                        Dist = (Dist_temp[0] - 126)+(133 - Dist_temp[1]) # x + y
                                elif (auto == 'left_do'):
                                        Dist = (Dist_temp[0] - 123)+(Dist_temp[1] - 117) # x + y
                                else:
                                        Dist = (Dist_temp[0] - 117)
                        if auto[0:3] == 'up_':
                                if (auto == 'up_righ'):
                                        Dist = (123 - Dist_temp[1])+(133 - Dist_temp[0]) # y + x
                                elif (auto == 'up_left'):
                                        Dist = (127 - Dist_temp[1])+(Dist_temp[0] - 117) # y + x
                                else:
```

Figure 32 List_priority function

The function calculates the arrival time of each vehicle at the end of intersection point. Clearly the calculation is done by determining for each vehicle the distance from its current position to the adequate end point, according to its route. When the distance to the end point of intersection has been determined, the time required to reach this point is determined.

```
        Vel_temp = traci.vehicle.getSpeed(_list_[i])
        distance_max_acc=((_ACCEL*((MAX_SPEED-Vel_temp)/_ACCEL)*((MAX_SPEED-
                        Vel_temp)/_ACCEL))/2)+Vel_temp*(MAX_SPEED-Vel_temp)/_ACCEL
        if distance_max_acc > Dist:
                Value = (-Vel_temp+(math.sqrt(Vel_temp*Vel_temp+2*_ACCEL*Dist)))/_ACCEL
        else:
                Value = ((MAX_SPEED-Vel_temp)/_ACCEL) + ((Dist-distance_max_acc)/MAX_SPEED)

        # to prevent faster vehicles from taking priority

        if i > 0 and Value <= priority_list_[i-1]:
                Vel_temp = traci.vehicle.getSpeed(_list_[i-1])
                if distance_max_acc > Dist:
                        Value = (-Vel_temp+(math.sqrt(Vel_temp*Vel_temp+2*_ACCEL*Dist)))/_ACCEL
                else:
                        Value =((MAX_SPEED-Vel_temp)/_ACCEL)+((Dist-distance_max_acc)/MAX_SPEED)
        priority_list_.append(Value)
return priority_list_
```

Figure 33 Time Calculation to reach end intersection

The calculation of the time required to reach the end point of the intersection is made considering the vehicle speed, the maximum applicable acceleration, the maximum vehicle speed and the maximum lane speed. It is taken into account that the vehicle, according to its own parameters, can travel the stretch of road in uniformly accelerated motion, until reaching the maximum speed, and at constant speed for the following time. It should also be borne in mind that vehicles belonging to the same track cannot exceed the vehicles in front of them even if they have a higher priority. In the simulations made there is no overtaking lane, therefore, in the case just mentioned the vehicle will have a priority determined by the speed of the vehicle in front of it. All vehicles will calculate the list of priorities and must coincide with the list calculated by all other vehicles involved, otherwise the vehicles will stop at the beginning of the intersection and everything will be recalculated. In fact, this aspect is taken into account in this work, but it is considered that in the actual situation if this happens, for example due to communication errors, the system resets and a new security condition is set. Basically, the system of lists generates a blockchain with a consensus between vehicles set at 100% of the subjects involved. When the priority list has been determined, the only thing left to do is to adjust the speed of the vehicles according to the priority set. The vehicle with the highest priority, the one with the lowest arrival time at the end of the intersection, will have no modulation. The next vehicle, in terms of priority, will adjust its speed according to the vehicle with the highest priority. Speed modulation will allow the second vehicle on the list to reach the start of the intersection at the same time as the priority vehicle leaves the intersection. Similarly, the third vehicle in the list will adapt to the second vehicle to reach the point of intersection at the same time as the second vehicle has left the intersection and so on.

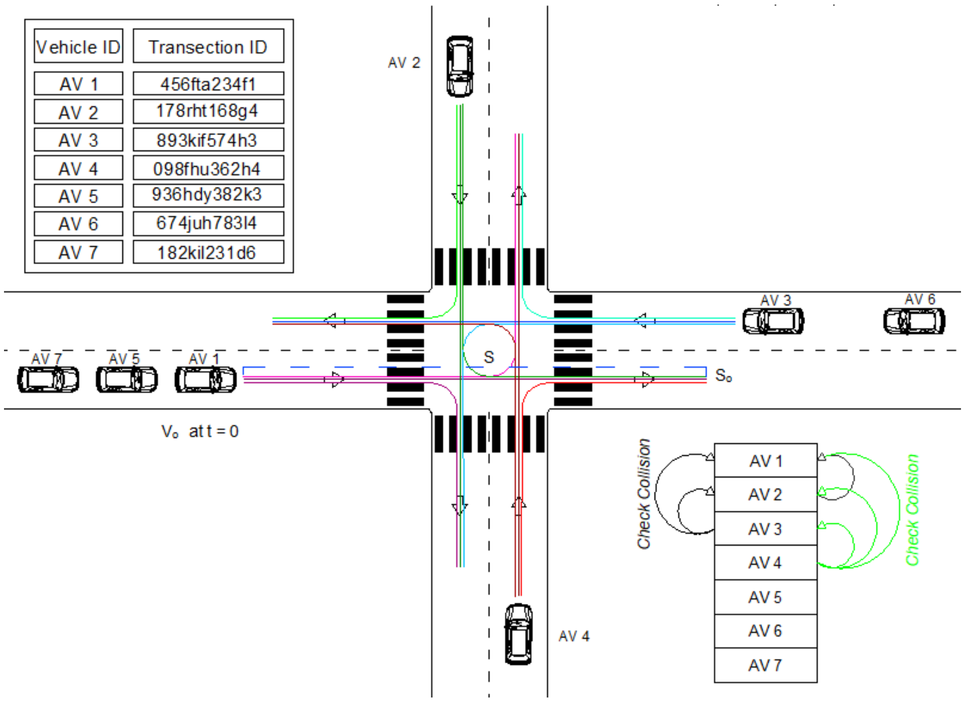


Figure 34 Priority List in Intersection

The modulation function will regulate the speed of the least priority vehicle according to the next priority vehicle. Modulation will only occur if the vehicles concerned [Vehicle_Priority(i) and Vehicle_Priority(i-1)] have trajectories that end in a collision, otherwise no speed modulation makes sense. In fact, for example, if the vehicles AV-3 and AV-4 in *Figure 34 Priority List in Intersection* proceed according to a route that does not foresee collisions (AV-3 straight and AV-4 right turn) no modulation takes place because it would be of no use and would only result in a reduction of the crossing time with all that follows. In the simulations carried out, it was decided to work by setting the entire intersection area as a safety area, i.e. to modulate the speed of the vehicles so that they entered at the intersection only after the exit of the vehicle with the highest priority. This choice, made only for maximum safety, can be theoretically optimized by dividing the intersection into safety sub-cells. Since the safety of these systems is of vital importance, it was decided to make a choice in absolute favour of safety.

```
def function_decel(speed_p,veh_priority,veh):
    Sb = traci.vehicle.getPosition(veh)
    Sa = traci.vehicle.getPosition(veh_priority)
    vb = traci.vehicle.getSpeed(veh)
    va = traci.vehicle.getSpeed(veh_priority)

    if veh[0:3] == 'rig':                    #distance of the non-priority vehicle to the start of the junction
            dist_b = (119 - Sb[0])
    elif veh[0:3] == 'lef':
            dist_b = (Sb[0] - 131)
    elif veh[0:3] == 'up_':
            dist_b = (119 - Sb[1])
    elif veh[0:3] == 'dow':
            dist_b = (Sb[1] - 131)

    if veh_priority[0:7] == 'right_r' and (133 - Sa[0]) >= 0:   #time needed for the priority vehicle to reach the end of the junction
                        #for large times where dist_b>vb*temp the function becomes a hyperbole with non-real solutions
            temp_a = (133 - Sa[0])/speed_p
            if dist_b < vb * temp_a and temp_a > 50:
                    temp_a = dist_b/vb
                    decel = -_DECEL
            else:
                    decel = 2 * (dist_b - (vb * temp_a)) / (temp_a * temp_a)
                    #deceleration to be applied to the least priority vehicle
    elif veh_priority[0:7] == 'right_u' and ((127 - Sa[0])+(133 - Sa[1])) >= 0:
            temp_a = ((127 - Sa[0])+(133 - Sa[1]))/speed_p
            if dist_b < vb * temp_a and temp_a > 50:
                    temp_a = dist_b/vb
                    decel = -_DECEL
            else:
                    decel = 2 * (dist_b - (vb * temp_a)) / (temp_a * temp_a)
```

Figure 35Function of Modulation speed

The speed modulation function will determine the time it takes for the vehicle with the highest priority to reach the end point of intersection (tend) and then will impose an acceleration or deceleration on the non-priority vehicle to ensure that it arrives at the start of intersection exactly at the same time (t_end). For safety purposes we could have avoided to impose an acceleration but in doing so we are able to optimize the crossing time of all vehicles. Clearly in a real situation this could be a constrain choice depending on traffic conditions, i.e. imposed in high traffic conditions. Even it could be a choice of each vehicle. If the owner wants to reduce travel time at the expense of consumption, he will decide to apply the necessary acceleration, while if he wants to reduce the consumption, he will decide not to apply the acceleration.

Another operating state is "Balance", this mode has been provided for all those cases of high congestion. A different algorithm to improve the efficiency of the system in conditions of congestion must be applied. Two different congestion situations can occur: one determined by the high presence of vehicles, and a second congestion situation determined by a lane with low speed vehicles and a second lane with vehicles close to each other at a high speed.

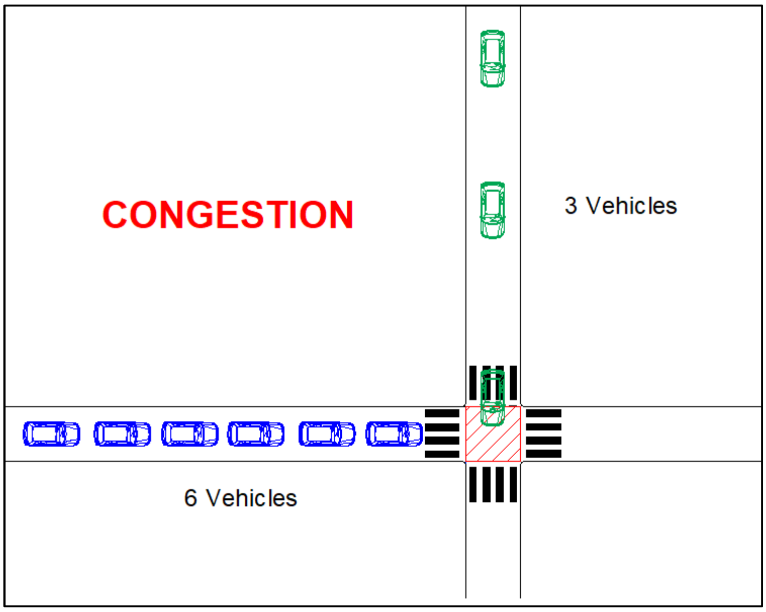


Figure 36 Congestion

As shown in *Figure 36 Congestion* if green vehicles have high speed and are close enough to each other, blue vehicles from the other lane may never be entitled to priority under the FRFP algorithm. Therefore, the **State Balance** has been introduced to overcome this and to optimize intersection management in high congestion conditions.

In an initial phase this state was managed through the platoon crossing, in particular the number of vehicles belonging to the crossing group was directly proportional to the number of vehicles on the lane compared to the number of vehicles on the competing lane. In the example in *Figure 36 Congestion* the number of blue vehicles is twice as high as the number of green vehicles of the competing lane, therefore, by fixing a minimum number of elements of a platoon of 2, there will be an alternating passage of 4 blue cars and 2 green cars and so on. So, in an initial phase we thought of a balanced crossing according to the vehicles involved regardless of the parameters of the vehicles. In this case the vehicles priority are not provide for by the Regulation state and their priorities will accord to a balance between the lanes. Subsequently, the system developed was compared with other methods (Platoon, FCFS, Priority Right, Ballroom, Traffic Light) and from the comparison it was decided to use the FCFS algorithm for all situations of congestion. In fact, in similar speed situations between vehicles and with reduced values (typical congestion situation) our algorithm is very similar to the FCFS algorithm.

In case of congestion the behaviour between FRFP and FCFS is very similar. But the FRFP algorithm in this case has the disadvantage of being able to generate a block of vehicles in a lane. As shown in *Figure 36* Congestion if green vehicles have high speed and are close to each other, blue vehicles from the other lane may never be entitled to priority under the FRFP algorithm. The FCFS system presents in case of congestion a good efficiency and does not generate in any case the blockage of vehicles as shown in *Figure 36* Congestion.

When we are in a situation of high congestion, we can assume that all vehicles can be found at low speeds and therefore at similar speeds. In the latter case, the advantages between FRFP and FCFS flatten out and it makes sense to give priority to those closest to the intersection.

Therefore, the system developed determines a congestion situation when more vehicles on one lane are stationary or when one lane has a high number of vehicles compared to the other lane. In this situation the system recognizes a state of congestion entering the state of Balance. The FCFS algorithm has been implemented exactly as foreseen for FRFP. In particular, priority lists are generated but in this case, they will be a function of the proximity to the intersection instead of the time needed to reach the end point of intersection.

The process of speed modulation will always be the same. The priority vehicle will arrive at the end of the intersection at the same time as the next vehicle will arrive at the beginning of the intersection. This will ensure a safe and collision-free crossing.

Another limit situation is the one covered by the operating state ***Freeze***, a typical situation of a blockage of a vehicle (as shown in the Figure 37 *Freeze Pre intersection*), for example due to an anomaly in the vehicle. In this case, the vehicle sends an alert to the other vehicles. If the congestion occurs before the junction, all vehicles in front of it will not be entitled to priority until the vehicle resumes its motion.

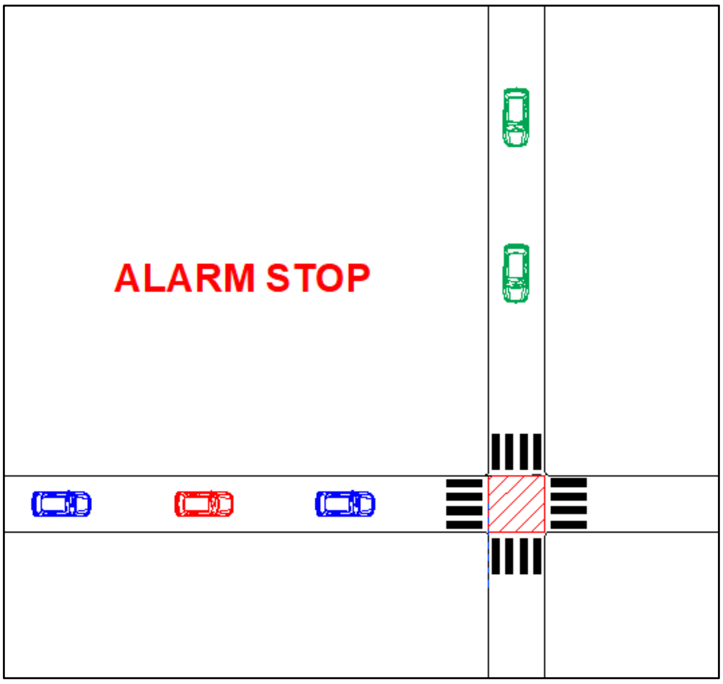


Figure 37 Freeze Pre intersection

In the case of post-crossing blocking, however, all vehicles that have blocked lanes on their route will not cross the crossroads, always keeping the intersection free. This choice allows you to limit the blocking of intersections in case of vehicle anomalies or in case of accidents.

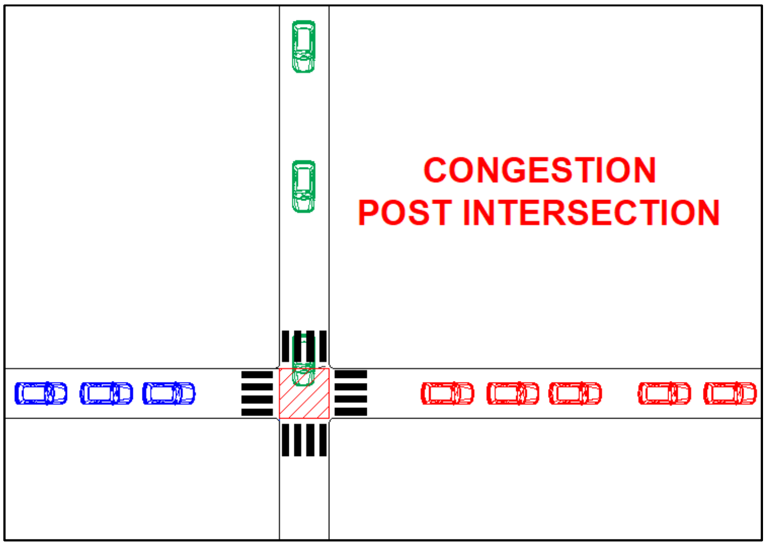


Figure 38 Freeze Post intersection

This type of intersection management obviously allows for priority vehicles regardless of the characteristics of the vehicles. For example, an ambulance or an alarmed police vehicle is a super user who always has the highest priority over the intersection, regardless of the operating state. With reference to the tests carried out on the roundabout, we managed the intersections with different protocols. In particular, we compared the FCFS protocol, the FRFP protocol and a system based on priorities, setting as a priority the distance of vehicles already within the roundabout. We also compared the management of an intersection by roundabout with the classic 8 lanes intersection.

The results obtained, as shown in the following chapter, show that the management of the roundabout with the best efficiency is that by prioritizing vehicles already within the roundabout. This is in line with common sense, if we do not prioritize the release of vehicles within the roundabout, we risk creating congestion that collapses the system. It is also very interesting to note that in a system characterized by the exclusive presence of automatic vehicles is absolutely inconvenient to use roundabouts. In an automatic system, classic intersections are much more effective than roundabouts.

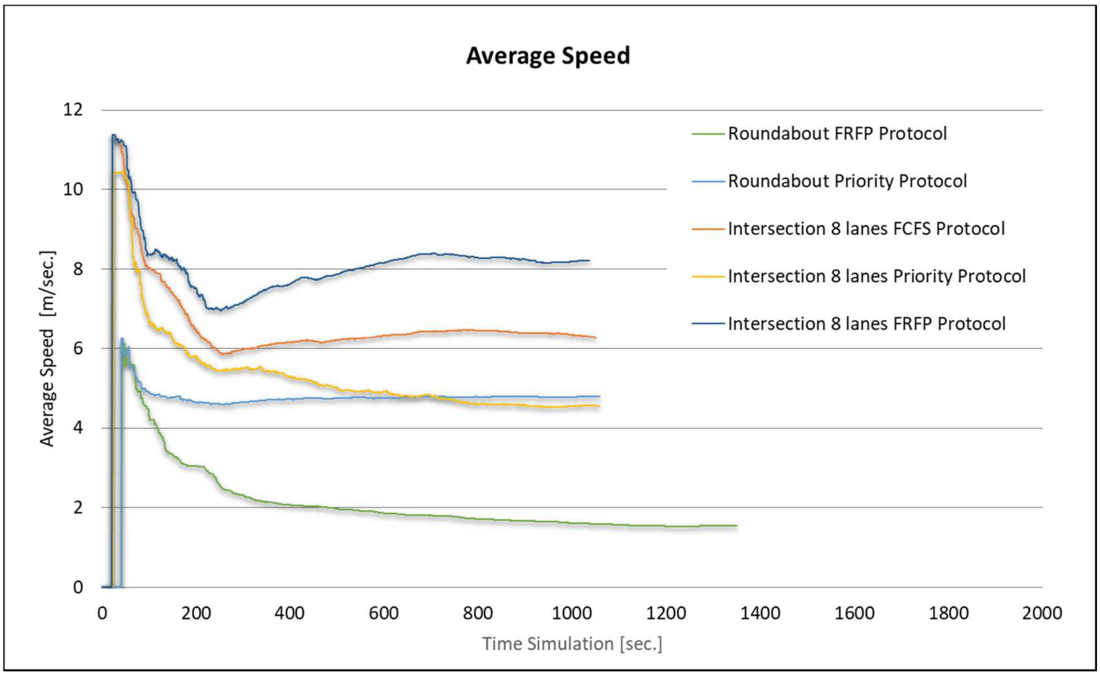


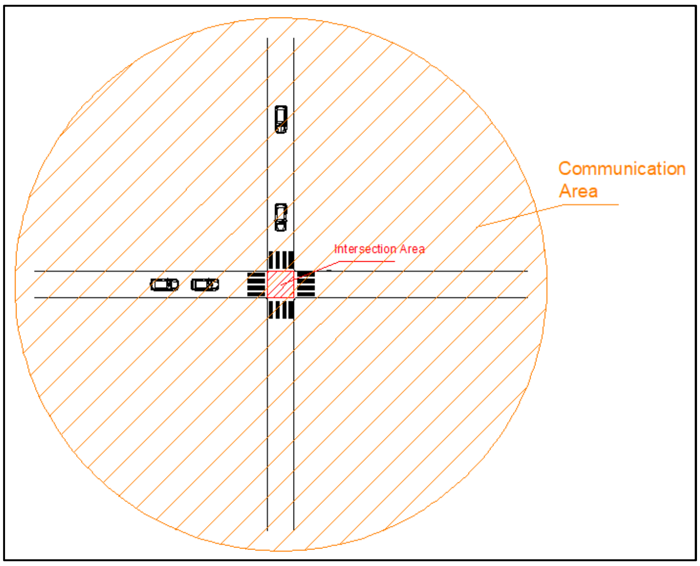
Figure 39 Roundabout Vs Intersection

Chapter 5

Result

The developed algorithm has been simulated and compared to different state-of-the-art systems. In particular, it has been compared with the classic traffic light system, with the FCFS system, with the Ballroom system, and with the priority system on the lane. Four different intersection systems have been implemented: 2 lanes intersection, on-ramp, 8 lanes intersection, and roundabout. The implemented systems have been tested under various congestion conditions. In our simulations, we have used SUMO (Simulation of Urban MObility) and NETEDIT to implement the intersection system. All the simulations were performed by setting a maximum speed of 13 m/s, a minimum gap between vehicles equal to 1m, a maximum acceleration of 4 m/s^2 and a maximum deceleration of 3 m/s^2. The communication system between vehicles is considered to be devoid of any error and a distance of 40 meters from the intersection under consideration. The refresh time of the communication is equal to 1 second.

Intersection type 1 (2 lanes):



For the 2 lanes intersection the two FCFS and FRFP sets have very similar performances. The simulation was performed generating a flow of vehicles equal to about 1950 per hour in a first case and equal to about 3450 per hour in a second case. The following algorithms have been compared: FCFS, FRFP, Traffic Light, Priority Right. For high congestions, it is normal not to notice any difference between the FCFS and FRFP algorithms. In fact, the intersections will be so crowded that the algorithms will behave in an almost equivalent way. The advantage over conventional systems is noticeable. Thus, in FRFP case we see an increase respect traffic light system in average speed of around 143% and a reduction in emissions of about 60%.

<u>*Case1 (1950 Vehicles/Hour)*: The following table shows the considerable advantages of the</u> developed system compared to traditional intersection management systems. As we can see from the results, the developed algorithm (FRFP) presents considerably higher performance comparing it to the classic traffic light system, but it is not much better than the FCFS system. The FRFP system shows an increase of 143.8% on average speed compared to the traffic light system and 0.1% on the FCFS system. As shown in the table below, it exhibits better performance also in terms of average transit time, emissions and fuel consumption.

	AVERAGE SPEED [m/sec]	Increase FRFP Vs Algorithm	AVERAGE CROSSING TIME [sec]	Reduction FRFP Vs Algorithm
Traffic Light	4.49	**143.8%**	57.26	**60.1%**
Priority R.	6.07	**80.3%**	42.84	**46.7%**
FCFS	10.93	**0.1%**	22.87	**0.1%**
FRFP	10.94	0.0%	22.86	0.0%
	EMISSION CO2 [mg]	**Reduction FRFP Vs Algorithm**	**FUEL CONSUMPTION [ml]**	**Reduction FRFP Vs Algorithm**
Traffic Light	68,344,189	**37.2%**	29,379	**37.2%**
Priority R.	60,673,945	**29.3%**	26,082	**29.3%**
FCFS	42,990,996	**0.2%**	18,480	**0.2%**
FRFP	42,900,123	0.0%	18,441	0.0%

Table 1 Intersection 2 lanes 1950 Vehicles/Hour

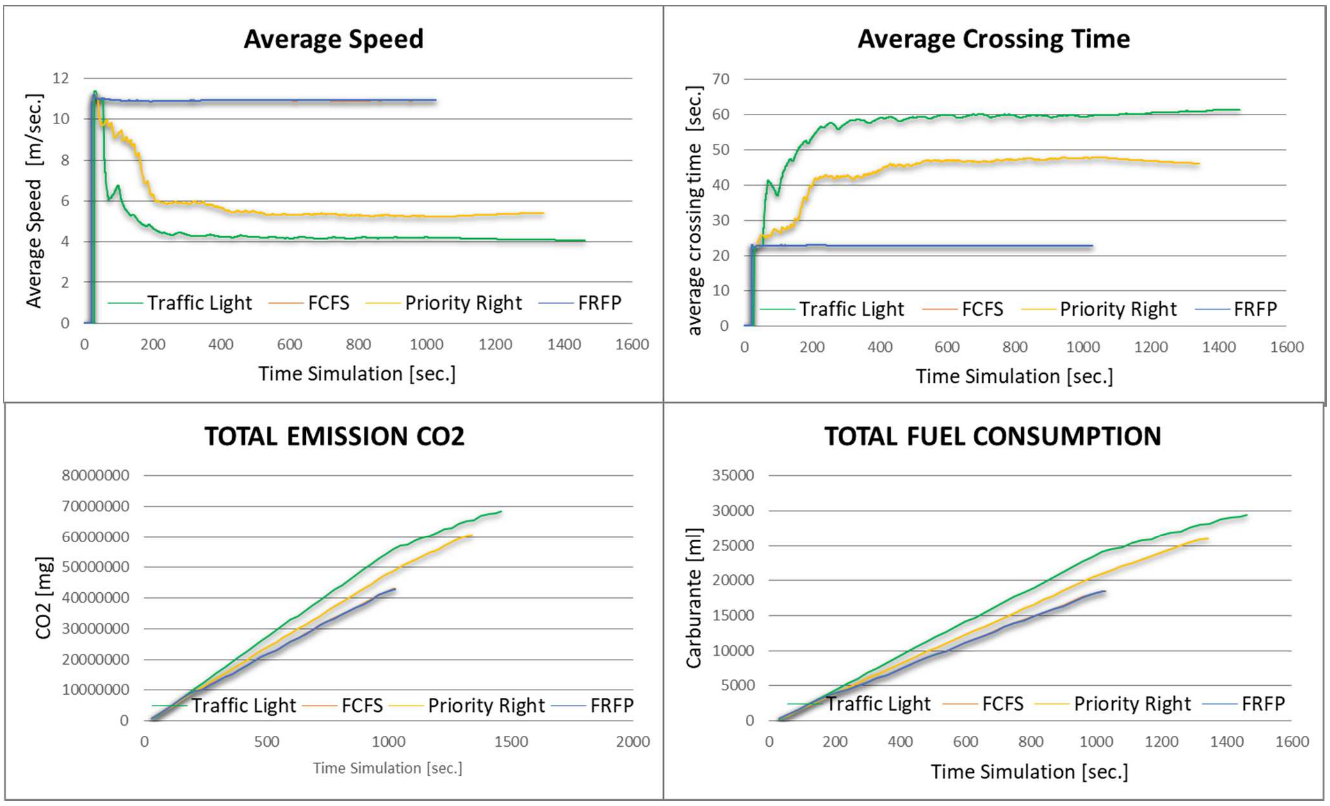


Figure 40 Crossing 2 lanes: inserted 558 vehicles equivalent to about 1956 vehicles/h for Algorithm Implemented [FRFP]

Case2 (3450 Vehicles/Hour): As we can see from the results, in this case, with a congestion considerably higher than the previous case, the algorithm developed (FRFP) presents a considerably higher performance comparing it to the classic traffic light system. And it performs a little better than the FCFS system. The FRFP system shows a 173.3% increase in average speed compared to the traffic light system and 0.8% compared to the FCFS system. As shown in the table below, it exhibits better performance also in terms of average transit time, emissions and fuel consumption.

	AVERAGE SPEED [m/sec]	Increase FRFP Vs Algorithm	AVERAGE CROSSING TIME [sec]	Reduction FRFP Vs Algorithm
Traffic Light	3.83	**175.3%**	66.41	**64.3%**
Priority R.	6.35	**66.2%**	44.01	**46.1%**
FCFS	10.46	**0.8%**	23.90	**0.8%**
FRFP	10.54	0.0%	23.71	0.0%
	EMISSION CO2 [mg]	Reduction FRFP Vs Algorithm	FUEL CONSUMPTION [ml]	Reduction FRFP Vs Algorithm
Priority R.	258,094,493	**36.7%**	110,946	**36.7%**
Traffic Light	255,912,571	**36.2%**	110,008	**36.2%**
FCFS	164,778,985	**0.9%**	70,831	**0.9%**
FRFP	163,263,697	0.0%	70,180	0.0%

Table 2 Intersection 2 lanes 3450 Vehicles/Hour

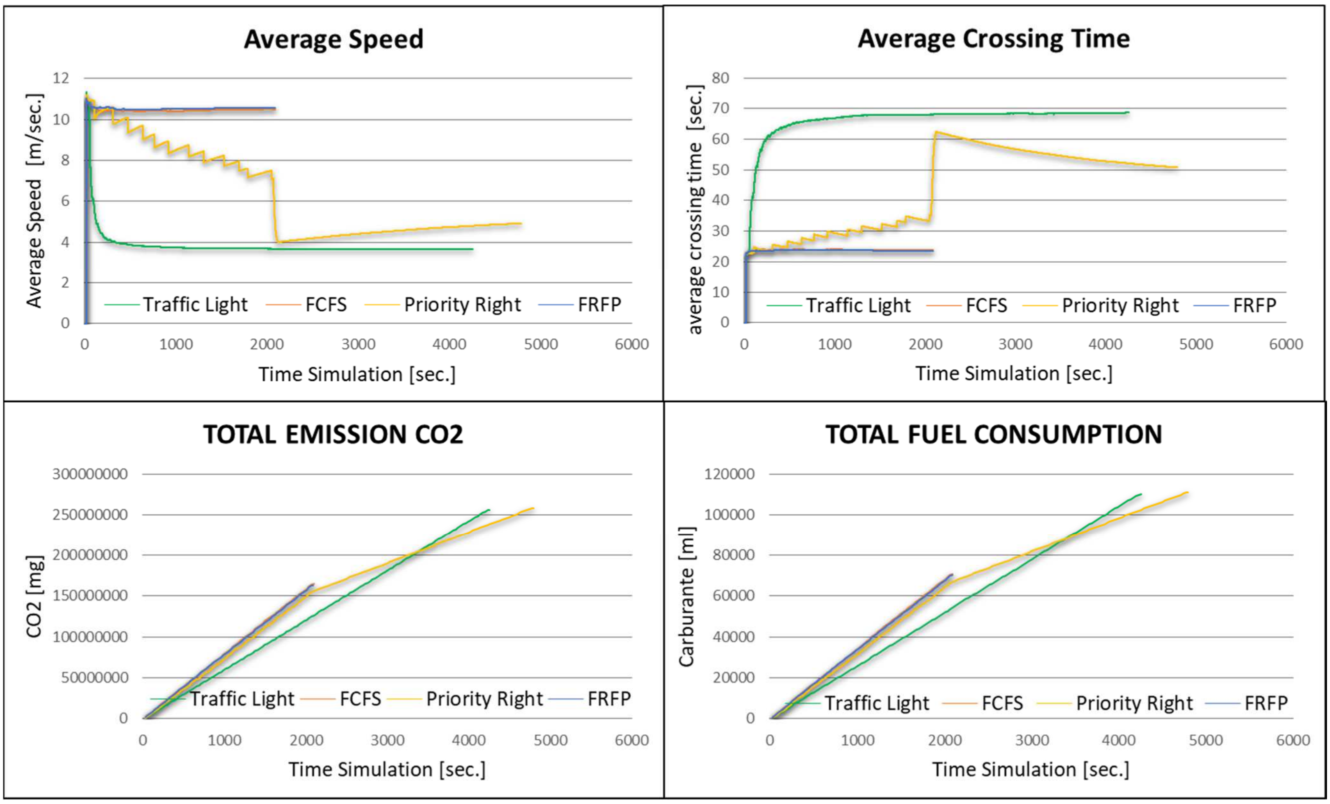
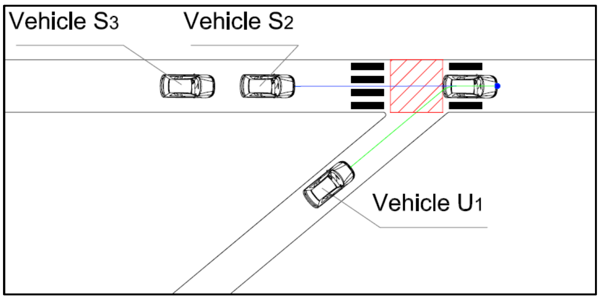


Figure 41 Crossing 2 lanes: inserted 2000 vehicles equivalent to about 3442 vehicles/h for Algorithm Implemented [FRFP]

Intersection type 2 (on-ramp):



The on-ramp system has been simulated by inserting around 500 vehicles for an equivalence of 1773 vehicles/h. Also in this case, the algorithm performs highly compared to traditional systems but very similar to the FCFS system. As for the 2 lanes intersections, for high congestions it is normal not to notice any difference between the FCFS and FRFP algorithms, in fact the intersections will be so crowded that the algorithms will behave in an almost equivalent way. In this case we see an increase in average speed of around 39% and a reduction in emissions of about 12.6% compared to the traffic light system.

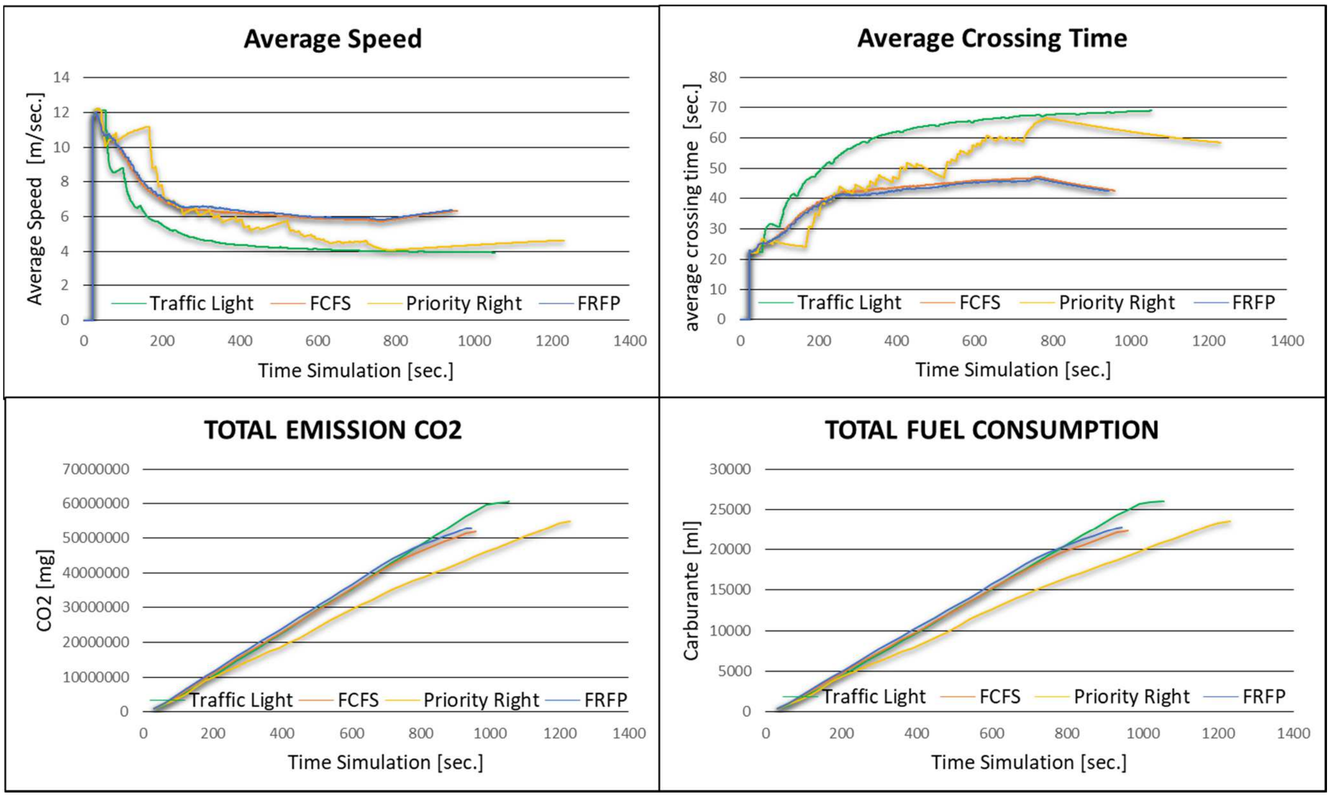


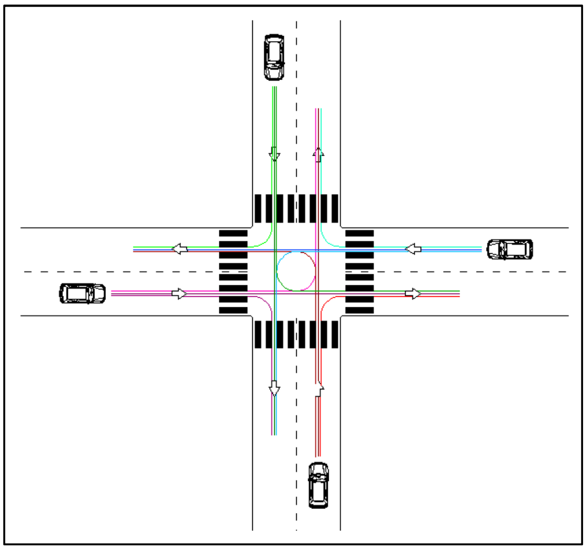
Figure 42 On-ramp crossing: inserted 465 vehicles equivalent to about 1773 vehicles/h for Algorithm Implemented [FRFP]

Clearly, the variation in performance is linked to the type of intersection and the vehicle flow. This simulation was set by forcing a large number of vehicles on the intersection ramp.

	AVERAGE SPEED [m/sec]	Increase FRFP Vs Algorithm	AVERAGE CROSSING TIME [sec]	Reduction FRFP Vs Algorithm
Traffic Light	4.91	**38.8%**	59.39	**31.0%**
Priority R.	5.96	**14.2%**	50.37	**18.6%**
FCFS	6.7	**1.7%**	41.73	**1.7%**
FRFP	6.81	0.0%	41.00	0.0%
	EMISSION CO2 [mg]	**Reduction FRFP Vs Algorithm**	**FUEL CONSUMPTION [ml]**	**Reduction FRFP Vs Algorithm**
Traffic Light	60,569,184	**12.6%**	26,037	**12.6%**
Priority R.	54,780,609	**3.4%**	23,548	**3.4%**
FCFS	51,981,274	**-1.8%**	22,345	**-1.8%**
FRFP	52,912,280	0.0%	22,745	0.0%

Table 3 Intersection on-ramp 1773 Vehicles/Hour

Intersection type 3 (8 lanes):



This type of intersection is the most complex. The "ballroom" algorithm has been implemented with considerable application limits. In this case we also compared the algorithm of the "ballroom". Although the latter algorithm has several limitations. In fact, in this case the vehicles cannot turn at the intersection, the vehicles can cross the intersection maintaining the same trajectory. Vehicles must have the same size. In the real application, this system, even if under some conditions it exhibits reasonable performance, is not easily applicable both for application limits and for the limited safety margins. The algorithm will no longer be considered in the other tests. This type of intersection has been tested with 4 different levels of congestion: 1640, 1950, 2380, and around 2530. The results are remarkable, showing at best an increase of 22.4% on average speed and a reduction of 14.4% on emissions compared to the FCFS system. The system is more effective the more difficult the congestion conditions are. Obviously, the results do not improve any more reaching the levels of saturation of the lanes.

Case1 (1640 Vehicles/Hour): In this case, which is the first case with greater complexity, we obtain an increase in average speed of 5.2% and a reduction in emissions of 1.3% compared to the FCFS system. The comparison with the traditional traffic light system leads to an increase in the average speed of 208.9%.

	AVERAGE SPEED [m/sec]	Increase FRFP Vs Algorithm	AVERAGE CROSSING TIME [sec]	Reduction FRFP Vs Algorithm
Traffic Light	3.41	**208.9%**	83.41	**71.6%**
BALLROOM	7.30	**44.4%**	34.26	**30.7%**
Priority R.	8.8	**19.7%**	28.42	**16.5%**
FCFS	10.02	**5.2%**	25.01	**5.1%**
FRFP	10.54	0.0%	23.73	0.0%
	EMISSION CO2 [mg]	**Reduction FRFP Vs Algorithm**	**FUEL CONSUMPTION [ml]**	**Reduction FRFP Vs Algorithm**
Traffic Light	81,138,991	**55.2%**	34,879	**55.2%**
Priority R.	41,503,757	**12.4%**	17,841	**12.4%**
FCFS	36,855,093	**1.3%**	15,842	**1.3%**
FRFS	36,371,317	0.0%	15,634	0.0%
BALLROOM	36,168,224	**-0.6%**	15,548	**-0.6%**

Table 4 Intersection 8 lanes 1640 Vehicles/Hour

From the table, it seems that the ballroom system exhibits better performance in terms of emissions and fuel consumption, but the comparison cannot be done on the same conditions given the considerable limits imposed to simulate the algorithm.

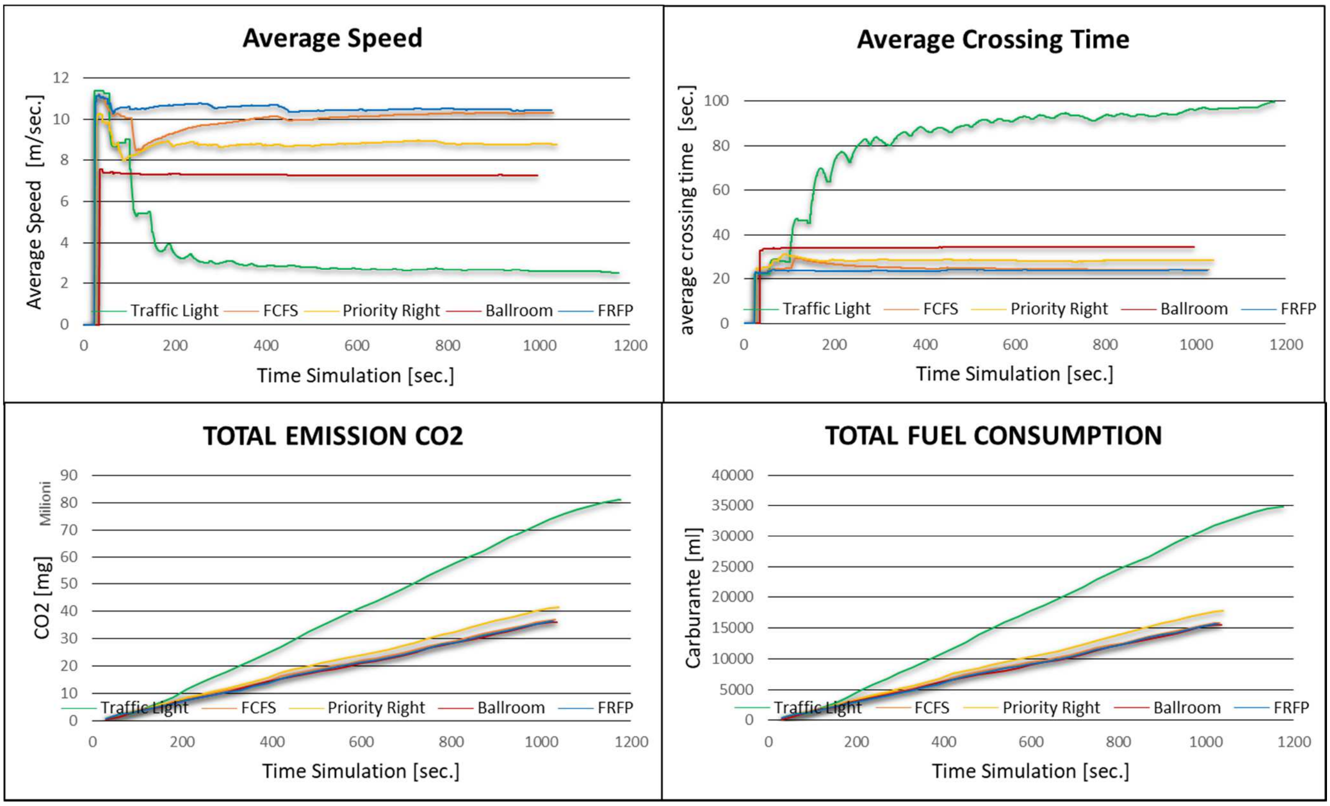


Figure 43 Crossing 8 lanes: inserted 468 vehicles equivalent to about 1640 vehicles/h for Algorithm Implemented [FRFP]

Case 2 (1950 Vehicles/Hour): *As the congestion increases, the performance of the developed system (FRFP) gets more efficient than the other systems.*

	AVERAGE SPEED [m/sec]	Increase FRFP Vs Algorithm	AVERAGE CROSSING TIME [sec]	Reduction FRFP Vs Algorithm
Traffic Light	2.73	**238.8%**	107.86	**74.9%**
Priority R.	7.56	**22.4%**	33.19	**18.3%**
FCFS	8.63	**7.3%**	29.04	**6.6%**
FRFP	9.25	0.0%	27.12	0.0%
	EMISSION CO2 [mg]	**Reduction FRFP Vs Algorithm**	**FUEL CONSUMPTION [ml]**	**Reduction FRFP Vs Algorithm**
Traffic Light	128,514,065	**61.9%**	55,245	**61.9%**
Priority R.	56,154,276	**12.8%**	24,138	**12.8%**
FCFS	49,186,934	**0.5%**	21,143	**0.5%**
FRFP	48,961,661	0.0%	21,047	0.0%

Table 5 Intersection 8 lanes 1950 Vehicles/Hour

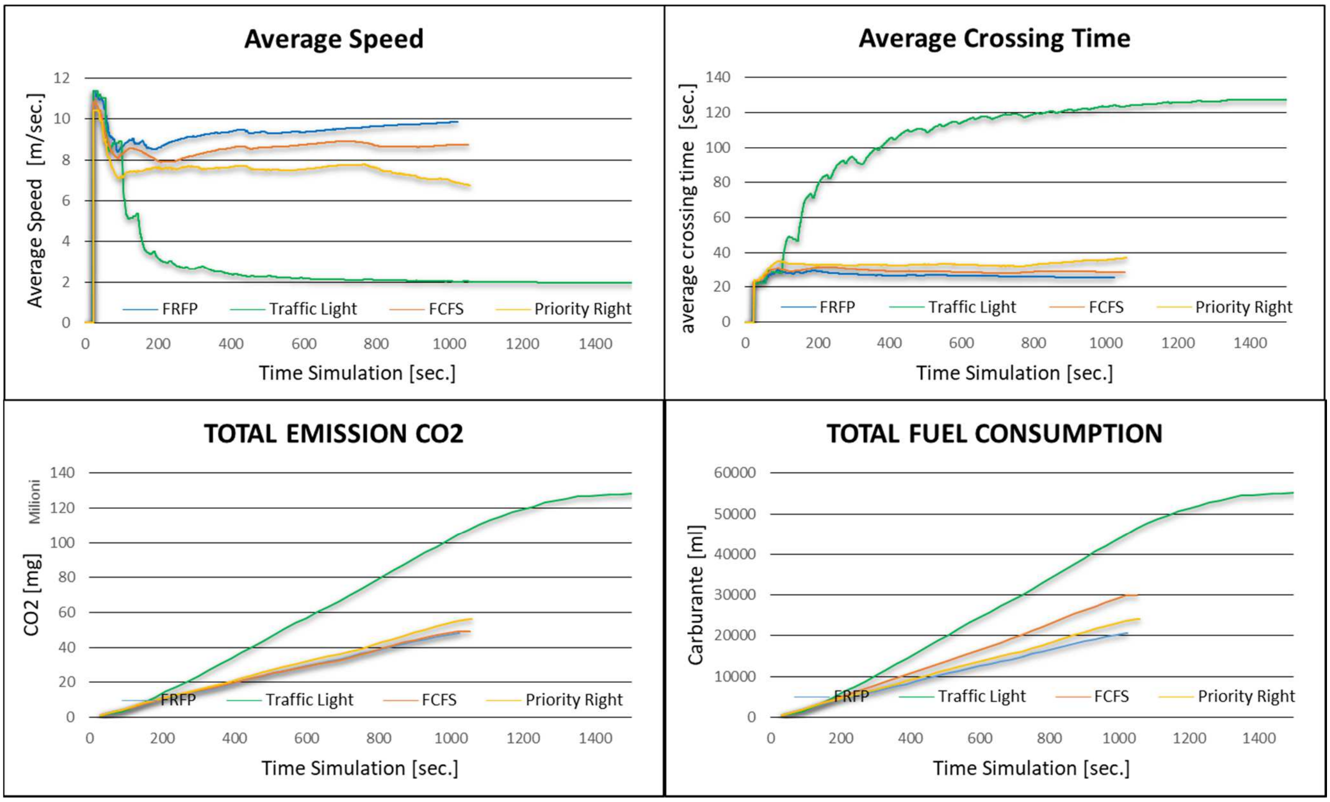


Figure 44 Crossing 8 lanes: inserted 587 vehicles equivalent to about 1950 vehicles/h for Algorithm Implemented [FRFP]

Case 3 (2380 Vehicles/Hour): Again, the performance of the developed system (FRFP) gets more efficient than the other systems.

	AVERAGE SPEED [m/sec]	Increase FRFP Vs Algorithm	AVERAGE CROSSING TIME [sec]	Reduction FRFP Vs Algorithm
Traffic Light	2.56	**217.3%**	113.45	**72.7%**
Priority R.	5.33	**52.5%**	48.46	**36.1%**
FCFS	6.64	**22.4%**	38.26	**19.1%**
FRFP	8.13	0.0%	30.97	0.0%
	EMISSION CO2 [mg]	**Reduction FRFP Vs Algorithm**	**FUEL CONSUMPTION [ml]**	**Reduction FRFP Vs Algorithm**
Traffic Light	155,409,183	**59.7%**	66,807	**59.7%**
Priority R.	77,506,256	**19.3%**	33,317	**19.3%**
FCFS	73,131,010	**14.4%**	31,436	**14.4%**
FRFP	62,571,993	0.0%	26,897	0.0%

Table 6 Intersection 8 lanes 2380 Vehicles/Hour

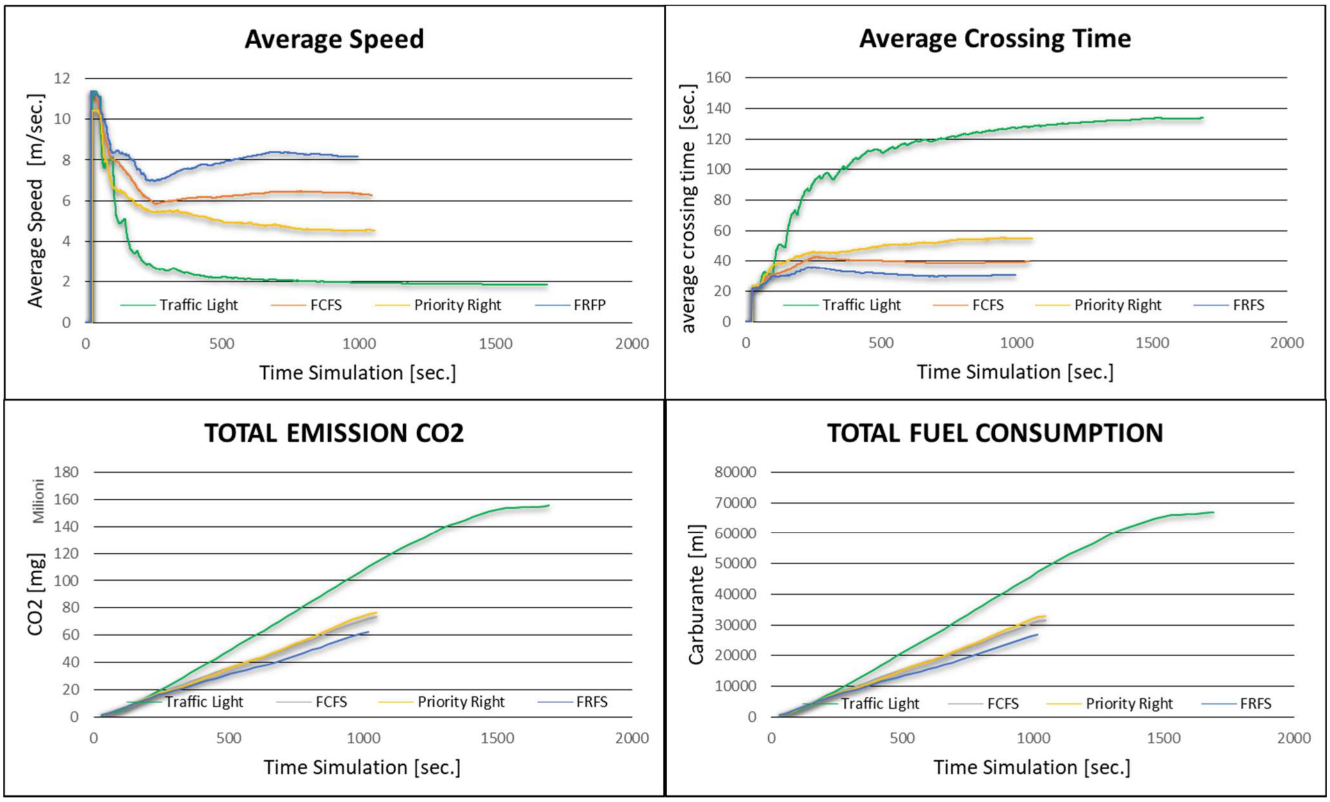


Figure 45 Crossing 8 lanes: inserted 685 vehicles equivalent to about 2380 vehicles/h for Algorithm Implemented [FRFP]

Case 4 (2535 Vehicles/Hour): In this case, we are close to the capacity limit of the lanes. In fact, we see a collapse of the performance of the tested algorithms. Also in this case our algorithm outperforms the other systems.

	AVERAGE SPEED [m/sec]	Increase FRFP Vs Algorithm	AVERAGE CROSSING TIME [sec]	Reduction FRFP Vs Algorithm
Traffic Light	2.52	**104.9%**	114.81	**55.2%**
Priority R.	4.42	**16.5%**	59.79	**13.9%**
FCFS	4.77	**8.0%**	56.17	**8.4%**
FRFP	5.15	0.0%	51.46	0.0%
	EMISSION CO2 [mg]	**Reduction FRFP Vs Algorithm**	**FUEL CONSUMPTION [ml]**	**Reduction FRFP Vs Algorithm**
Traffic Light	184,500,946	**45.9%**	79,312	**45.9%**
FCFS	107,698,980	**7.4%**	46,296	**7.4%**
Priority R.	101,270,765	**1.5%**	43,533	**1.5%**
FRFP	99,760,467	0.0%	26,897	0.0%

Table 7 Intersection 8 lanes 2535 Vehicles/Hour

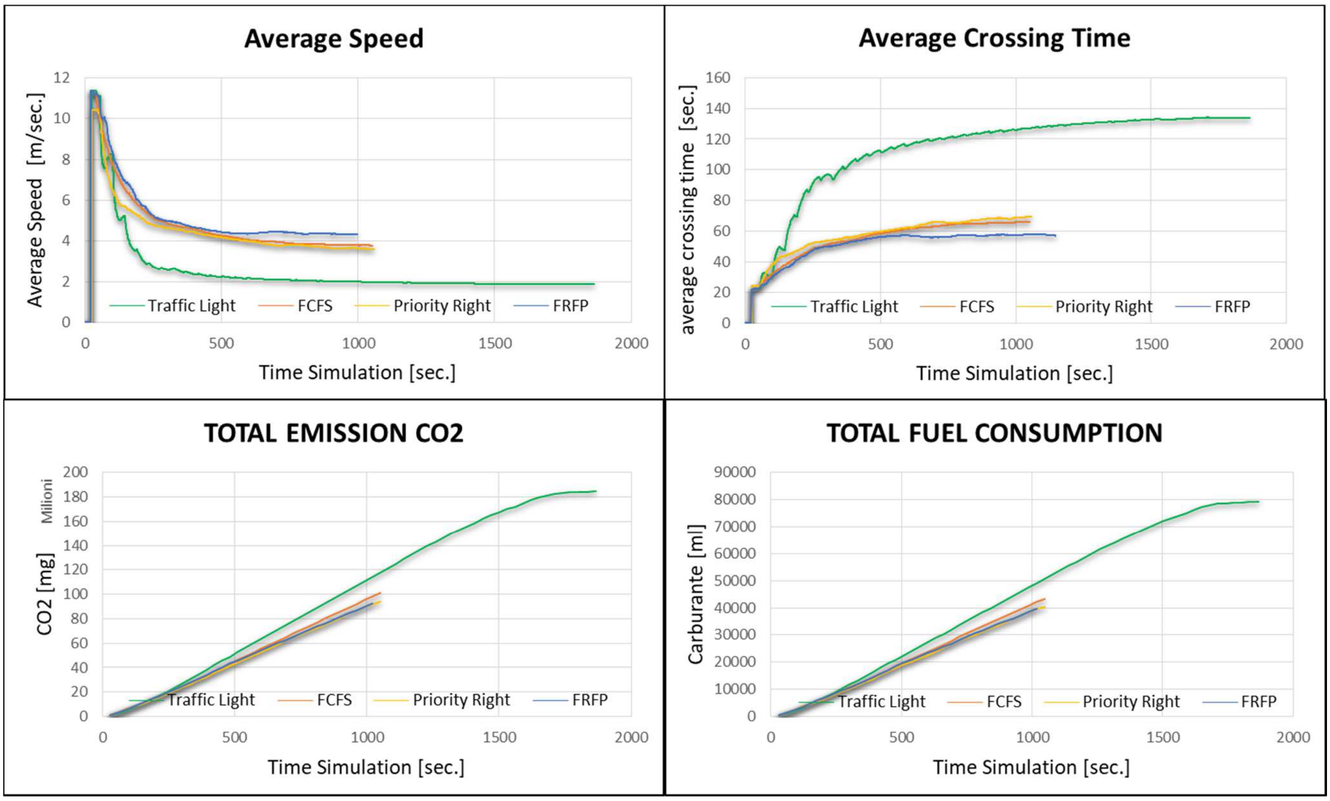
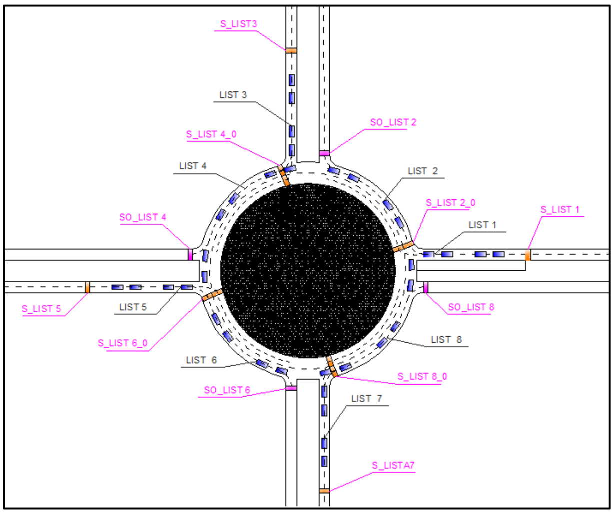


Figure 46 Crossing 8 lanes: inserted 808 vehicles equivalent to about 2535 vehicles/h for Algorithm Implemented [FRFP]

In the best case, we have an increase in average speed, with a consequent reduction in average travel time, equal to about 240% compared to the classic traffic light system and about 22% compared to the FCFS system. In the worst case, the percentages still better by 105% and 5% respectively.

Intersection type 4 (roundabout):



The comparison between the intersection and the roundabout highlights how in an automatic driving approach the roundabout is much less efficient compared to a classic crossing. In addition, the management of the roundabout is certainly more effective based on priority. Speeding up the entry of vehicles through other systems the roundabout is congestedand slows down the system. In the figures below and in the following tables we compare our algorithm (FRFP) and the same system applying the right priority. The right-priority system is more effective than any algorithm tested. The same graphs show another interesting comparison. In the same traffic conditions, the roundabout system is compared to the 8 lanes intersection. It is compared to the right-priority algorithms, FCFS and FRFP. The conclusion is that, for self-driving vehicles, the roundabouts are worse way than classic road intersections.

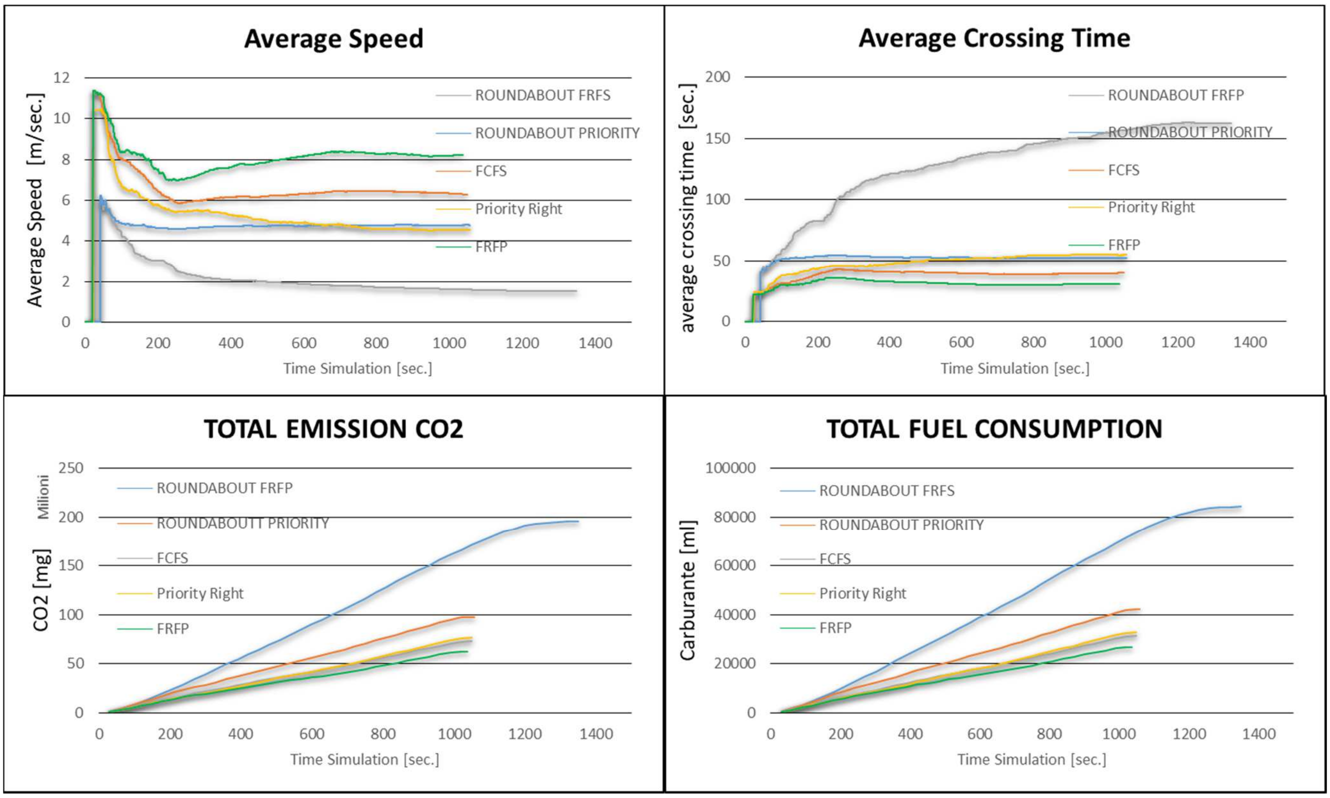


Figure 47 Crossing 8 lanes Vs Roundabout: inserted 685 vehicles equivalent to about 2378 vehicles/h for Algorithm Implemented [FRFP]

	AVERAGE SPEED [m/sec]	Increase FRFP Vs Algorithm	AVERAGE CROSSING TIME [sec]	Reduction FRFP Vs Algorithm
ROUND FRFP	2.11	285.5%	130.4	76.2%
ROUND PRIORITY	4.78	70.1%	52.34	40.8%
Priority R. 8 lanes	5.33	52.5%	48.46	36.1%
FCFS 8 lanes	6.64	22.4%	38.26	19.1%
FRFP 8 lanes	8.13	0.0%	30.97	0.0%
	EMISSION CO2 [mg]	Reduction FRFP Vs Algorithm	FUEL CONSUMPTION [ml]	Reduction FRFP Vs Algorithm
ROUND FRFP	195,903,582	68.1%	84,213	68.1%
ROUND PRIORITY	97,958,974	36.1%	42,108	36.1%
Priority R. 8 lanes	77,506,256	19.3%	33,317	19.3%
FCFS 8 lanes	73,131,010	14.4%	31,436	14.4%
FRFP 8 lanes	62,571,993	0.0%	26,897	0.0%

Table 8 Intersection 8 lanes Vs Roundabout 2378 Vehicles/Hour

Chapter 6

Main Contributions

In this thesis, a new algorithm for the management of road intersections is developed in the exclusive presence of automatically guided vehicles. The importance of the work developed lies in having found a reliable management of road intersections in terms of safety, guaranteeing a "zero collision" system in the absence of communication and evaluation errors of the position sensors on the vehicles. In addition, the system is well functioning both when the type of intersection to which it is applied varies but also according to different road congestion conditions.

The proposed method, which can be used in any type of intersection, is simulated and compared with other state-of-the-art algorithms in 2 lanes, 8 lanes, on-ramp and roundabout intersections. The vehicles involved are free to go straight or turn at the intersection. The entire intersection area is considered as a safety area that can be occupied by only one vehicle at a time. The aforementioned area can also be divided into cells. The proposed system does not necessarily require an intersection operator but is based on vehicle-to-vehicle communications (V2V).

The latter is another strength of the implemented system as it would allow a rapid application of the system in zero time without the need to install a large number of intersection operators. This, in fact, would certainly make the start-up phase very complex and very long time.

The results obtained are very promising. The best case is the 8-lane crossing, which coincides with one of the most critical cases and with the largest number of vehicles (2380 Vehicle/Hour). In this more critical case, with a high number of vehicles, we highlight the best performance of our system. The average increase in speed is 22.4% compared to the FCFS system and 217.3% compared to the classic traffic light system.

The reduction of crossing time has considerable benefits not only on the average journey time of vehicles, but also on fuel consumption, emissions and certainly also on people's quality of life. The simulations were carried out considering the entire intersection as an area reserved for a single vehicle in order to increase road safety. In reality, by maintaining a high level of safety, this area can be reduced and its benefits increased.

It also shows a 19.1% reduction in average crossing time compared to the FCFS system and a 72.7% reduction compared to the classic traffic light system; a 14.4% reduction in CO_2 emissions and fuel consumption compared to the FCFS system and a 59.7% reduction compared to the classic traffic light system. The system developed, as shown by the results obtained from the simulations, presents excellent performances in all the types of intersections examined. Another important result was obtained in the analysis of roundabouts in systems with only the presence of self-driving vehicles without driver. In fact, in this case, it is true that roundabouts are less effective than the classic road crossings and that through the proposed system (FRFP) both travel times, CO_2 emissions and consumption are considerably reduced. The system can be further improved by applying V2C communication systems to perform a pre-setting already at long distances.

In fact, it has also been seen how the algorithm can already be applied at considerable distances from the intersection by means of V2C communications, in this case obtaining high level optimizations also in routing and modulation of vehicle speeds in order to optimize fuel consumption, emissions and travel time.

Future Work

The work opens up different scenarios on what are automatic guidance systems.

1. Certainly through the application of the algorithm on V2C communications. Interesting would be to determine on different possible routing those with greater efficiency in terms of air quality in population centers as well as traffic congestion, consumption and travel times. It could be applied, for example, depending on weather conditions and air quality in a certain area and therefore to prefer alternative routes in order to reduce emissions in a certain area that could be critical at that precise moment. This could also be done for example by considering possible risk factors in the choice of routing.

2. Important evolution of this research work may be the integration of pedestrians into the system and the analysis of a pre-regulatory approach through the V2C Communication. This type of communication could also facilitate the routing of vehicles both according to different environmental parameters to reduce emissions and to facilitate pedestrian crossing in cases of high pedestrian presence.

3. Another aspect to be analysed, especially for the initial phase of application of these systems, is to foresee the presence of a certain percentage of non-automatic vehicles in the more complex context in the presence of manually driven vehicles and pedestrians.

4. Possible future work could certainly be the analysis of top priority vehicles such as security posts (ambulances, law enforcement) and possible high priority vehicles that could be given to public transport for example.

5. Analysis of the contribution of pedestrian crossing to road congestion, thus defining possible priority management algorithms between automatically driven vehicles, manually driven vehicles and pedestrians.

Chapter 7

Bibliography

[1] B. Filocamo, A. Galletta, M. Fazio, J. Alonso and M. A. Sotelo, "An Innovative Osmotic Computing Framework for Self Adapting City Traffic in Autonomous Vehicle Environment", MOCS, 2018.

[2] B. Filocamo, J. A. Ruiz and M. A. Sotelo, "Efficient management of road intersections for automated vehicles – The FRFP system applied to the various types of intersections and roundabouts," *IEEE ITS Magazine (submitted)* , 2019.

[3] A. Buzachis, B. Filocamo, M. Fazio, J. A. Ruiz, M. A. Sotelo and M. Villari, "Distributed Priority Based Management of Road Intersections Using Blockchain," *IEEE ISCC WKSHPS-MoCS,* 2019.

[4] K. Dresner and P. Stone, "Multiagent traffic management: A reservation-based intersection control mechanism", 2004.

[5] K. Dresner and P. Stone, "A Multiagent Approach to Autonomous Intersection Management" JOURNAL OF ARTIFICIAL INTELLIGENCE RESEARCH, 2008.

[6] T. Au and P. Stone, "Motion Planning Algorithms for Autonomous Intersection Management", 2010.

[7] J. Lee and B. Park, "Development and Evaluation of a Cooperative Vehicle Intersection Control Algorithm Under the Connected Vehicle Environment", 2012.

[8] R. Azimi, G. Bhatia, R. Rajkumar and P. Mudalige, "Ballroom Intersection Protocol: Synchronous Autonomous Driving at Intersections", 2015.

[9] S. Aoki and R. Rajkumar, "A Merging Protocol for Self-Driving Vehicles", 2017.

[10] S. A. Fayazi and A. Vahidi, "Mixed-Integer Linear Programming for Optimal Scheduling of Autonomous Vehicle Intersection Crossing", 2018.

[11] K. Zhang, C. Xie, Y. Wang, M. Wang, A. De La Fortelle, W. Zhang and Z. Duan, "Service-Oriented Cooperation Policies for Intelligent Ground Vehicle Approaching Intersections," *MDPI Applied Sciences,* 2018.

[12] A. Mirheli, M. Tajalli, L. Hajibabai and A. Hajbabaie, "A consensus-based distributed trajectory control in a signal-free intersection," *Elsevier,* 2019.

[13] 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," *Elsevier,* 2018.

[14] Behrisch M., Bieker L., Erdmann J., Krajzewic D., "SUMO - Simulator of Urban MObility," 2011, pp. pp. 55-60.

[15] X. Qian, F. Altchè, Jean Frègoire, A. De La Fortel, “Autonomous Intersection Management systems: criteria, implementation and evaluation,” 2015.

[16] N. Mensi, M. Guizani, A. Makhlouf, “Study of vehicular cloud during traffic congestion,” 2016.

[17] github.cm, “Github Repository: Hyperledger Caliper,” 2019. [Online]. Available: http://github.com/hyperledger/caliper/.

[18] J.A. Leon Calvo and R. Mathar, “Secure blockchain-based communication scheme for connected vehicles,” 2018.

[19] N. Lasla, M Younis, W. Znaidi and D. Ben Arbia, “Efficient distributed admission and revocation using blockchain for cooperative its,” 2018.

[20] Google car pulled over for being too slow, 20 04 2018. [Online]. Available: http://www.bbc.com/news/technology-34808105.

[21] A.I.M. MMedina, N. Van de Wouw ad H. Nijmeijer, “Cooperative Intersection Control Based on Virtual Platooning,” 2018.

[22] A. Buzachis, A. Celesti, A. Galletta, M. Fazio and, “A secure and dependable multi-agent autonomous intersection management (ma-aim) system leveraging blockchain facilities,” 2018.

[23] Z. Lu, W. Liu, Q. Wang, G. Qu and Z. Liu, “A privacy-preserving trust model based on blockchain for vanets,” 2018.

[24] Zhang, F. Wang and J. J., “Trasportation 5.0 in cpss: Towards acpbased society-centered intelligent transportation,” 2017.

[25] I. Parra, R Izquierdo, J. Alonso, A. Garcìa-Morcil, “The experience of DRIVERTIVE – DRIVERLESS cooperative Vehicle,” GCDC, 2017.

[26] S.A. Fayazi, A. Vahidi and A. Luckow, “Optimal Scheduling of Autonomous Vehicle Arrivals at Intelligent Intersections via MILP,” 2017.

[27] A.I.A. Ahmed, A.G., S.H.A.H., S.K., N.G, K.K., “Intersection-based distance and traffic-aware routing (idtar) protocol for smart vehicular communication,” 2017.

[28] L. Chai, B. ShangGuan and J. Wang, “Connected and Autonomous Vehicles Coordinating Method at Intersection Utilizing Preassigned Slots,” 2017.

[29] A. Celesti, A.G., L.C., M.F., M.V., “An iot cloud system for traffic monitoring and vehicular accidents prevention based on mobile sensor data processing,” 2017.

[30] M. Bashiri, C.H.F., “A Platoon-Based Intersection Management System for Autonomous Vehicles,” 2017.

[31] Broggi, A., P. Cerri, S. Debattisti, M. C. Laghi, P. Medici, D. Molinari, M. Panciroli and A. Prioletti, “Proud - public road urban,” *IEEE Transactions on Intelligent Transportation System,* pp. vol. 16, pp. 3508–3519, 2015.

[32] J. Ziegler, P. Bender, M. Schreiber, H. Lateg, ..., Making bertha drive - an autonomous journey on, IEEE Intelligent Transportation Systems Magazine, vol. 6, pp. 8–20, 2014.

[33] J. Wang, Y. Huang, Z. Feng, C. Jiang, H. Zhang and ..., “Reliable traffic density estimation in vehicular network,” 2018.

[34] cryptorating.eu, “CUBE Autonomous Car Network Security Platform based on Blockchain,” 2018. [Online]. Available: https://cryptorating.eu/whitepapers/CUBE/CUBEWhite_Paper-V1.3.pdf.

[35] M.-A. Lebre, F. Le Mouel and E. Menard, “Resilient, Decentralized V2V Online Stop-free Strategy in a Complex Roundabout,” 2018.

[36] E. Debada, L. Makarem and D. Gillet, “Autonomous Coordination of Heterogeneous Vehicles at Roundabouts,” *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC),* 2016.

[37] “Traffic Coordinator at Road Intersections: Autonomous Decision-Making Algorithms Using Model-Based Heuristics,” *IEEE Intelligent transportation systems magazine,* 2017.

[38] M. Choi and H. H. Choi, “Reservation-Based Cooperative Traffic Management at an Intersection of Multi-lane Roads,” *IEEE International Conference on Information Networking (ICOIN),* 2018.

[39] B. V. Philip, T. Alpcan, J. Jin and M. Palaniswami, “Distributed Real-Time IoT for Autonomous Vehicles,” *IEEE Transactions on Industrial Informatics,* 2018.

[40] S. Aoki and R. Rajkumar, “A Configurable Synchronous Intersection Protocol for Self-Driving Vehicles,” *IEEE 23rd International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA),* 2017.

[41] J. L. F. Pereira and R. J. F. Rossetti, “An integrated architecture for autonomous vehicles simulation,” 2015.

[42] D. Smith, S. Djahel and J. Murphy, “A SUMO Based Evaluation of Road Incidents Impact on Traffic Congestion Level in Smart Cities,” *Local Computer Networks Workshops (LCN Workshops), 2014 IEEE 39th Conference on,* 2014.

[43] S. Wang , S. Djahel, J. McManis, C. McKenna and L. Murphy, “Comprehensive performance analysis and comparison of vehicles routing algorithms in smart cities,” *Global Information Infrastructure Symposium - GIIS,* 2013.

[44] T.-C. Au, S. Zhang and P. Stone, “Autonomous Intersection Management for Semi-Autonomous Vehicles,” *Handbook of Transportation,* 2015.

[45] T. Ogitsu , M. Omae and H. Shimizu, “Automatic Driving Control for Passing through Intersection by use of Feature of Electric Vehicle,” *World Electric Vehicle Journal Vol. 4 - ISSN 2032-6653,* 2010.

[46] H. Ahn and D. Del Vecchio, “Semi-autonomous Intersection Collision Avoidance through Job-shop Scheduling,” *Association for Computing Machinery (ACM),* 2017.

[47] R. Azimi , G. Bhatia and R. Raj, “Intersection Management using Vehicular Networks,” *SAE Technical Paper,* 2012.

[48] G. Sharon and P. Stone, “A Protocol for Mixed Autonomous and Human-Operated Vehicles at Intersections,” *AMBUS,* 2017.

[49] W. Sheng, Q. Yang and Y. Guo, "Cooperative Driving based on Inter-vehicle Communications: Experimental Platform and Algorithm," *International Conference on Intelligent Robots and Systems,* 2006.

[50] M. Hausknecht , T.-C. Au and P. Stone, "Autonomous Intersection Management: Multi-Intersection Optimization," *IROS,* 2011.

[51] T.-C. Au, C.-L. Fok, S. Vishwanath, C. Julien and P. Stone, "Evasion Planning for Autonomous Vehicles at Intersections," *IROS,* 2012.

[52] D. Carlino, S. D. Boyles and P. Stone, "Auction-based autonomous intersection management," *IEEE Intelligent Transportation Systems Conference,* 2013.

[53] D. Carlino, M. Depinet , P. Khandelwal and P. Stone, "Approximately Orchestrated Routing and Transportation Analyzer Large-scale Traffic Simulation for Autonomous Vehicles," *IEEE Intelligent Transportation Systems Conference,* 2012.

[54] G. Sharon, J. Hanna, T. Rambha , M. Albert, P. Stone and S. D. Boyles, "Delta-Tolling: Adaptive Tolling for Optimizing Traffic Throughput," *International Workshop on Agents in Traffic and Transportation,* 2016.

[55] A. Maitra , S. Bhadkaria , C. Ghosh and S. Paul, "An Integrated Transport Advisory System for Commuters, Operators and City Control Centres," in *First International Workshop on Vehicular Traffic Management for Smart Cities (VTM)*, 2012.

[56] H. Rashid, M. J. Fazel, M. Azizi and M. Reza, "Intelligent Traffic Light Control Based on Clustering using Vehicular Ad-hoc Networks," in *International Conference on Information and Knowledge Technology*, 2015.