New Intersection Control for Conventional and Automated Vehicles without Traffic Lights

A combination of self-regulation and individual control

Anna Cristofoli

TU Delft

New Intersection Control for Conventional and Automated Vehicles without Traffic Lights

by

Anna Cristofoli

to obtain the degree of Master of Science at the Delft University of Technology, Faculty of Civil Engineering and Geosciences, Department "Trasport and Planning".

Defence date
<>

Student number
4505409

Graduation committee

Prof. dr. ir. S. P. Hoogendoorn	TU Delft, CEG	Chairman
dr. V. Knoop	TU Delft, CEG	Daily Supervisor
dr. M. Spaan	TU Delft, EWI	External Supervisor
ir. W. van Nifterick	ARS T&TT	Company Supervisor

An electronic version of this thesis is available at http://repository.tudelft.nl/.

This thesis has been conducted in cooperation with ARS Traffic & Transport Technology

Preface

This thesis is the end result of a long and challenging research endeavour. This work is intended for everyone who is interested in automated driving, vehicle-to-infrastructure communications and intersection control management.

I had a hard time trying to manage together the graduation research, a part-time job and my tendencies to not work structurally. A recipe for disaster. The first one to fall was the motivation: finishing the graduation project at times seemed an unreachable goal. Nevertheless, here I am with a final report I am (somewhat) happy with. Needless to say, without the help and support - both personally and professionally - of certain persons I definitely would not have made it.

With respect to my committee, I want to thank especially Victor Knoop for his supervision. His knowledge, his sincere feedback and his constant and genuine support have made him an ideal first supervisor for me.

Next, I thank my past Company supervisor Boudewijn Schokker for engaging in interesting discussions and sharing his practical knowledge of diverse topics. A special thanks goes to my current Company supervisor Wim van Nifterick who has accepted to join the committee at few weeks of his (unofficial) retirement.

Furthermore, I thank my external supervisor Matthijs Spaan and the chairman Serge Hoogendoorn for their feedback and guidance during the meetings.

At last, I want to thank my family, friends and old roomates who always supported me and who made my study time unforgettable. A special thank goes to my boyfriend Erwin who experienced first hand the my *ongezellige* mood during this long journey and who is looking forward to have an engineer girlfriend.

Anna Cristofoli
Den Haag, February 2019

Summary

Typically, an intersection consists of a number of approaching roads and a crossing area. The conventional approach to solve this problem is to assign sequentially the right-of-way to a stream of vehicles, grouped by compatible (non-conflicting) directions. The effectiveness of a conventional traffic controller is highly affected by how traffic demand is modelled and whether the signal plan can be responsive to the changes in traffic condition. Intuitively, real-time traffic measurements and the optimization of both the sequence and signal timing leads to the most flexible and responsive control strategy. The significant progress achieved in the development of vehicles automation and telecommunication technologies is promising to offer the right tools to achieve such strategy. The main focus of the available literature has been directed at developing solutions for a traffic scenario with full penetration of connected or automated vehicles. Solutions developed for the transitional phase mainly consider how connected or automated vehicles can be used to improve the efficiency of the intersection within the conventional control strategy.

This thesis aims to develop a novel control strategy that efficiently controls traffic with different penetration rates by relying purely on wireless communication to integrate the control of automated and human drivers. Automated vehicles are considered to be also connected while conventional vehicles are not. The intersection does not need traffic lights so while V2I communications can be used to individually control automated vehicle, the motion of conventional vehicles will be indirectly influenced by controlling the speed of automated vehicles. Radar detection technology is used at the intersection to detect and track all vehicles present. Within the scope of this research, the main research question has been formulated as follows:

How can the performance of an intersection be improved based on a certain objective by using a speed and sequence control strategy for traffic with a mixed level of automation?

In order to answer the main question, after formulating the control strategy in detail, a simulation environment is used to evaluate the performance of the proposed controller. This study, opposite to most literature, focuses on coordinating mixed traffic (automated and conventional) using a strategy commonly used to coordinate automated traffic only. In order to achieve this, traffic is modelled in induced platoons (*i*-platoon) and it is controlled by a combination of self-organization and individual coordination. The concept of *i*-platoons originate from the need to use automated vehicles to indirectly control conventional vehicles. Generally, an induce platoon is defined as a cluster of vehicles with time headway small enough to assume that the driving behavior of the vehicles is significantly influenced by their leader. If the leader is an automated vehicle, the *i*-platoon is defined controlled, uncontrolled otherwise. The controller uses real-time information on all *i*-platoons to optimize the crossing of controlled *i*-platoons via V2I communication. The scheduling of controlled *i*-platoons aims to decrease the total delay by guiding them to drive through the intersection without stopping. The uncontrolled *i*-platoons enter the intersection on their own accord, following standard traffic rules at an unsignalized intersection.

From a design perspective, the solution framework is translated into a bi-level optimization problem with the ultimate goal of computing the acceleration profile (control signal) of controlled i-platoons that yields the minimal total delay for all i-platoons in the network. The upper level optimization is a combinatorial optimization that aims to find the sequence of i-platoons with the least delay. It is solved by using a branch and bound technique with deep-first search algorithm. In the upper level, each sequence is associated with a control signal which generate the least delay for that sequence. This optimization is solved in the lower level, where, given that sequence, the best acceleration profile is computed. Considering that the lower optimization is computed iteratively, the trajectory optimization is simplified to an piece-wise acceleration profile. Controlled i-platoons can either cruise until leaving the intersection with the current speed or decelerate with the maximum comfortable deceleration until they reach a speed with which they can use to cruise for the rest of the journey. The movement of uncontrolled i-platoon needs to be predicted to avoid collisions at the crossing area from conflicting streams with controlled i-platoons. This is achieved using car-following and basic crossing models.

The effectiveness of the proposed traffic controller is evaluated in terms of average total delay and average number of stops. The performance of the controller is compared to the performance of fixed traffic light controller. The controllers are tested in 48 different scenarios considering penetration rates and traffic volumes. This testing is executed using the simulation platform Vissim-COM in Matlab. Vissim is a microscopic road traffic simulator based on individual behavior of vehicles. The intersection environment is simulated in VISSIM whereas the controller is programmed in Matlab. The COM Interface is a technology designed to enable inter-process communication between the Vissim software and Matlab, thus enabling the simulation of V2I communications.

The results show that the traffic control strategy is able to improve the efficiency of the intersection under unsaturated conditions (scenario 1,2,3). In this case, the coordination of vehicles lowers the need of stopping at crossing and make the journey of vehicle hindered-free. This results is achieved already with low penetration rate of 20%. Along with a reduction of delay, the strategy achieves a higher fairness in the delay distribution, as vehicles (in form of i-platoons) are sorted based on their exit time. During saturated condition (scenario 4), the control strategy shows a performance drop. At the lowest penetration rate 20%, the performance is even worst than traditional strategies and overturns only with higher penetration rate, in particular from the rate 60%. When uncontrolled i-platoons start queuing on the minor road and there is not enough platoons on the major road to slow down in favor of the minor road, the traffic condition cannot be improved. This results consist in a drawback of the strategy because it relates to the combination with self-organizing rules which do not allow for control over uncontrolled i-platoons.

The results have shown that the new strategy of self-regulating and individual control is able to efficiently coordinate vehicles at an intersection. In addition, the concept of induced platooning has proven to be an interesting way to integrate the control of conventional and automated vehicles. Due to the novelty of the control strategy, the research considered a very simple intersection layout to validate first its feasibility. In order to investigate further its applicability, a more complex environment should be considered in future work. Better balancing between computation effort and model accuracy is also an important element for improvement, even more so considering implementations in real-life cases.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1. Motivation

Signalized intersections can be considered natural bottlenecks as the coordination between vehicles with incompatible paths leads to a decrease of road capacity. In an urban environment characterized by a grid of numerous intersections, the traffic condition is heavily affected by the performance of control systems at these nodes. It is estimated that delays at traffic signals contribute a 5 to 10 percent of all traffic delays, or 295 million vehicle-hours of delays, on major roadways alone in the USA (National Transportation Operations Coalition, 2012). Therefore, the traffic signal control systems play an essential role in optimizing the flow of traffic through the city.

Typically, an intersection consists of a number of approaching roads and a crossing area. The crossing area is a space used by vehicles with conflicting directions. The traffic control goal consists in transforming inflows from all approaching roads into outflows while preventing collision and achieving a performance objective (Yan et al., 2011). The conventional approach to solve this problem is to assign sequentially the right-of-way to a stream of vehicles, grouped by compatible (non-conflicting) directions. The timing of the right-of-ways is provided in a signal-time plan which is displayed to the traffic via a traffic light showing green,yellow and red lights. The calculation of this plan is based on the traffic demand with the objective of optimizing (not always explicitly) performance metrics. However, the traffic demand is an unpredictable variable and fluctuates in time (Hoogendoorn, 2016). The effectiveness of a conventional traffic controller is therefore highly affected by how traffic demand is modelled and whether the signal plan can be responsive to the changes in traffic condition. Intuitively, real-time traffic measurements and the optimization of both the sequence and signal timing leads to the most flexible and responsive control strategy.

This strategy cannot be implemented at the current state of traffic management for a number of reasons. The most common sensor technology for vehicle detection is induction loop. These sensors are installed under the pavement of the road and they are usually located at the proximity of the stop-line and at a certain upstream distance. Loops can only provide instantaneous vehicle information when a vehicle is passing over the detector and cannot measure the vehicle state as it approaches and eventually reaches the intersection (Jing et al., 2017). Moreover, the utilization of multiple loops in order to increase the detection capabilities is discouraged because of their intrusive nature and consequent high cost of installation and maintenance. The lack of coverage coupled with cost inefficiency have encouraged the interest in new non-intrusive detection technologies. Among them, the traffic radar technology appears to have the greatest potential by being able to detect and track vehicles trajectories for long stretches of roads. Even though the technology has yet to be implemented in large scale applications, the integration of these sensors in the current framework of a traffic controller has been tested in real-life cases study (Krikke, 2017),(Kooijman, 2016). Under the assumption that accurate information can be obtained and delivered with a sufficient update rate, the conventional control strategy is still limited by the fact that real-time optimization of both the sequence and signal timing would lead to an abrupt variation of right-of-ways (green light). Such signal can be perceived too chaotic and confusing to road users and can create a safety problem when drivers are not capable of reacting in time to the unpredictable light change.

The significant progress achieved in the development of vehicles automation and telecommunication technologies during the recent decades is promising to offer solutions to overcome the aforementioned limitations. Within the framework of connected vehicles, vehicles can communicate wirelessly with each other V2V (Vehicle-2-Vehicle) and with the infrastructure V2I (Vehicle-2-Infrastructure). Through this channel, it possible to track the movement of vehicles by providing continuous detailed information such as individual position, speed, acceleration and route choice. Hence, communication between vehicle and infrastructure supplies the controller with an accurate image of the incoming traffic condition ahead of time. In addition, such V2I technology offers authorized parties the possibility of controlling the trajectory of vehicles. Intersection controllers can exploit these opportunities to coordinate the known arrival of vehicles individually. Undoubtedly, the automated vehicle environment has the potential to revolutionize the conventional approach of the traffic management problems. For these reasons, wireless communication and automated driving have drawn the attention of numerous amount of research.

The main focus of the available literature is directed at developing solutions for a traffic scenario with full penetration of connected or automated vehicles. This condition allows to take full advantage of the new vehicular technologies and it improves significantly the performance of an intersection controller that applies traditional strategies. The realization of the full penetration scenario is expected to go through a long period of transition where vehicles with different levels of automation will have to co-exist on the road. Only few papers have addressed the impact of the transition phase on the performance of the controller (Yang et al., 2016),(Ilgin Guler et al., 2014). These papers mainly considered how connected or automated vehicles can be used to improve the efficiency of the intersection within the conventional control strategy. This thesis aims to tackle the current limitations of the literature by testing a novel control strategy that controls traffic with conventional and automated vehicles. The proposed strategy intents to rely purely on wireless communication to integrate the control of automated and human drivers so that complete flexibility of traffic control can be achieved. The following section describes what are the challenges of applying this control strategy, which will be used as basis to define the research goal of this thesis.

1.2. Problem Description

This research focuses on an urban intersection that relies purely on wireless communication to coordinate right-of-ways of the traffic. No traffic lights are used for ordinary operations. The incoming vehicles should be controlled in such a way that the V2I communication is able to influence the crossing behavior of human and automated vehicles at the urban intersection in a safe and efficient manner. By equipping the intersection infrastructure with radar detection technology, the radar detection of conventional vehicles can act as proxy for the communication from the vehicle to the intersection infrastructure. The communication from the infrastructure to the conventional vehicles cannot be facilitated. Thus, it needs to be investigated how the controller can use the intersection environment in order to indirectly control the conventional vehicles. Assuming that the behavior of a driver can be influenced by the behavior of the preceding vehicle, it needs to be investigated how the speed of the automated vehicles can influence the conventional drivers to the extend that their behavior can be controlled. The opportunity to coordinate the vehicles individually leads to the question of which strategy can be used to schedule the intersection crossing and dynamically assign the right of ways to the vehicles. The proposed control strategy should be considered as an opportunity to improve the current performance of conventional controls when automated vehicles start to be driven in the urban environment. Therefore, the ultimate goal of the research consists in designing a traffic controller able to safely and efficiently coordinate a mixed flow of conventional and automated vehicles, while aiming to improve the performance of the intersection controlled by standard traffic controllers.

1.3. Research Objectives and Scope

In light of the problem and goal description presented previously, the main research question of the thesis can be formulated as follow:

How can the performance of an intersection be improved based on a certain objective by using a speed and sequence control strategy for traffic of automated and conventional vehicles?

There are different ways to develop this study. The scope of the research is to investigate the problem from a traffic engineering perspective. The focus will be on the methodology choices and techniques needed to design a control algorithm that will be able to perform according to the stated objectives. In order to support this investigation, the following sub-questions will be addressed:

1. What control approach is the most suited to deal with the traffic environment with a mixed flow of conventional and automated vehicles?
2. How can the coordination of the right-of-ways be assigned?
3. How can the speed profile of the automated vehicles be parameterized in order to optimize them?

The ultimate goal of the research is to improve to the performance of an intersection under the stated conditions. To quantify the effectiveness of the proposed solution and to investigate the condition for its hypothetical deployment, the following sub-questions will also be considered:

4. What is the range of penetration rate of automated vehicles necessary to affect correctly the human driving?

5. What is the improvement in terms of efficiency and qualitative driving comfort for different traffic flows compared to traditional signal control?

6. What are the implications of applying the control scheme to a more extended network of multiple intersections?

1.4. Research Approach

The thesis development consists of two main blocks: the controller design and the simulation platform, depicted in Figure 1.1. The first block describes the approach taken to design the controller and thus answer the methodology research sub-questions 1-3. The second block serves the purpose of evaluating the design made and thus answer the performance research questions 4-6. The development of these blocks is not strictly chronological and can run in parallel. In the remaining part of the section, each block is described in more detailed.

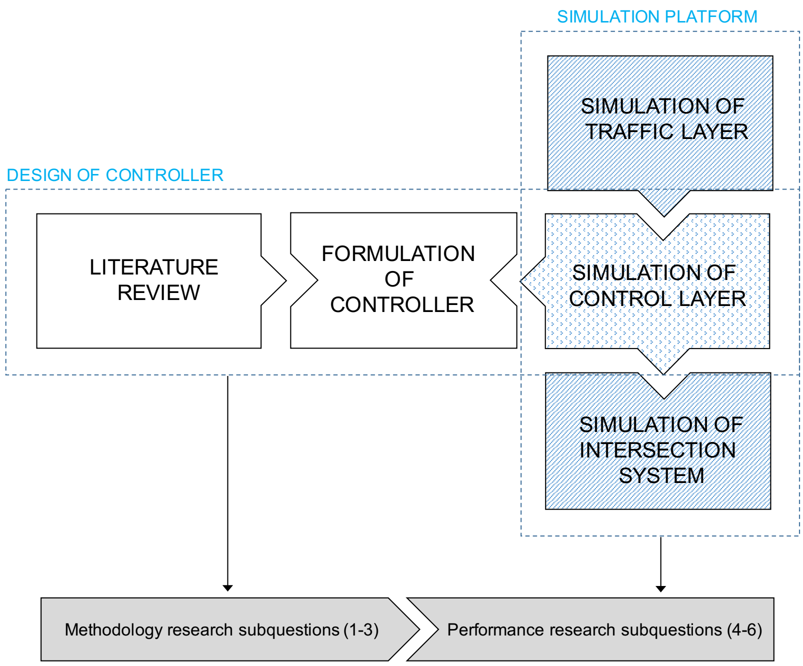


Figure 1.1: Flowchart of the Methodological Approach

1.4.1. Design of the Controller

The design of a traffic control system is a complex process that involves the formulation of several assumptions and choices that may influence the performance of the resulting system in different ways (Hegyi, 2012). Literature review is the input for the decision making process proving an insight on how these choices have been made in previous research which share similarity in their research objectives or research framework. The findings are then used as guidelines to formulate the controller mathematically. The formulation of the controller is divided into steps with simple goals, progressively increasing the complexity of its functionality. Before proceeding to the next step, the controller is evaluated to check whether the design achieves its goals. This is where the controller design interact with the simulation platform block. Once the controller reaches its final state, a complete simulation is run and thus ultimately evaluating its performance.

1.4.2. Simulation of the Controller

In order to evaluate the performance of the proposed algorithm, a custom microscopic simulator needs to be developed. The simulator needs to model two interacting layers: the traffic layer simulating the traffic dynamics and the control layer executing the control strategy using the real traffic information. This task can be achieved by programming both layers in a computing environment or starting with existing simulation software that models the traffic layer and programming only the control layer. Building a traffic simulator involves an intensive effort and can be time-consuming, especially considering that it is an evaluation tool and not the focus of the research itself. Depending on the resources available, there are different open-source and commercially-available options. In this research, the simulation software VISSIM (PTV, 2013) is used. A key feature of the software is availability of a Application Programming Interface (API) module that serves as interface between the traffic simulator and the user defined applications, such as a traffic controller. The intersection environment is simulated in VISSIM whereas the controller is coded in Matlab (The MathWorks Inc., 2018). Using the COM Interface, the implementation of the controller algorithm in the modelled intersection is finally tested and the results analyzed. In this last phase, the customized simulation provides the opportunity to explore in which traffic condition the algorithm works best and which parameters affect its efficiency. In order to quantify the benefit of implementing the controller, a comparison of the intersection performance is made using as reference a fixed-time traffic signaling control algorithm.

1.5. Thesis Outline

This report is split into five parts, namely introduction, literature review, controller design, controller evaluation and final critical assessment. The controller evaluation is split in two chapters, dividing the evaluation methodology and the evaluation results. The critical assessment is also split in two chapters, dividing discussion and final conclusions of the research. Figure 1.2 illustrates how the thesis outline reflects the research approach described in the previous section. The literature review is presented in its homonym chapter and the formulation of controller is described in the Design of Controller chapter. The simulation platform block is presented in the Evaluation chapters. The planning and implementation of the simulation are described in the Evaluation Methodology, whereas the results of the simulation are presented in the Evaluation Results chapter. A reflection of both design choices and simulation results are presented in the Discussion chapter. The discussion also consider practical implementation. The Conclusions chapter provides an answer to the sub-questions and finally to the main question, along with recommendations for future work.

A short description of each chapter is given in the following list.

1. **Introduction**
   Problem description and motivation to investigate the subject. The resulting research questions and research approach are presented.

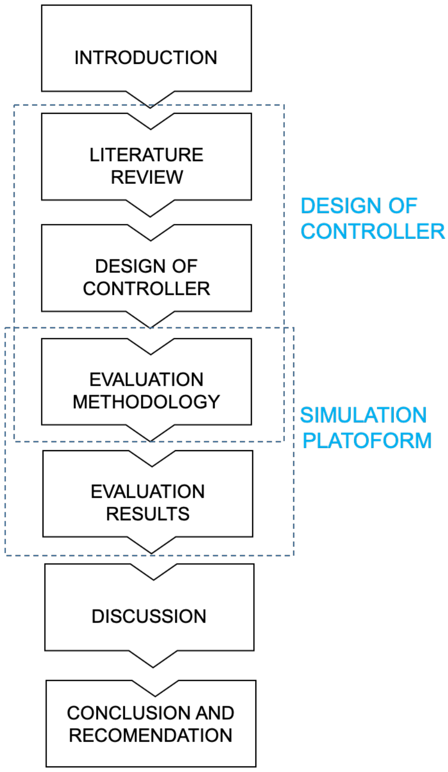


Figure 1.2: Flowchart of the Thesis Outline in relation to the Methodology Approach

2. **Literature Review**
   Theoretical background on the design of a traffic controller including the current practise and the proposed methodologies for automated vehicle scenarios.

3. **Controller Design**
   Assumptions and design choices made during the design process are explained along with the specification of the control algorithms.

4. **Evaluation Methodology**
   Proposed methodology to evaluate the performance of the traffic control. Both the evaluation plan and the simulation platform are described.

5. **Evaluation Results**
   Results of the simulations of the intersection according to the simulation platform and evaluation plan described in the previous chapter.

6. **Discussion**
   Critical assessment on the methodology chosen, assumptions made during the design choices and the results. Implication for real implementation are also considered.

7. **Conclusion and Recommendation**
   Final conclusions and recommendations for future studies.

CHAPTER 2

Literature Review

As described in Chapter 1, the goal of this research is to tackle the gap knowledge of mixed automated traffic control strategy by designing a traffic controller that uses a new control concept. In order to assist the decision making process necessary to develop the controller, research sub-questions on the methodology were formulated. In this chapter, a literature review is carried out to find how these questions have been answered in previous research. The results of this review will be used to select the control design most appropriate. In Chapter 3 these choices will be applied within the specifics characteristics of the research scope.

2.1. Connected Vehicles and Automated Vehicle Environments

Vehicle automation and vehicle communication have often been treated as distinct paradigm of the broad Intelligent Transport System (ITS) filed (Zhang et al., 2018). Within the framework of traffic control systems, both phenomena are considered game changers on how traffic management is implemented. However, little attention has been paid on defining their coexistence. This section aims to understand the interdependence of these two technologies and clarify their role within this research.

2.1.1. Connected vehicles

Connected vehicles combine several emerging technological advances, such as advanced wireless communications, on-board computer processing, advanced vehicle sensors, GPS navigation and smart infrastructure to provide a networked environment (Jing et al., 2017). The concept of distributing and sharing information was firstly promoted for safety purposes. The intention is to provide a safety message to warn drivers about hazards they may encounter through dedicated short-range communications (DSRC) (Al-Sultan et al., 2014). The amount of allocated bandwidth of DSRC has been shown to exceed the needs for its use by traffic safety applications. As such, several applications have emerged to utilize the additional bandwidth (Florin & Olariu, 2015). Connected vehicles, as an emerging technology, can mainly communicate with each other (V2V) and with the infrastructure (V2I). Within the scope of traffic management, the main system components of the wireless technology are the on-board unit (OBU) and road-side unit (RSU). Each vehicle is equipped with an on-board unit device and a set of sensors to collect, process the information and send it on as a message to other vehicles or RSU. The vehicles connect to the road-side unit or to other vehicles through a wireless link based on the IEEE 802.11p radio frequency channel, also called WAVE (Wireless Access in Vehicular Environments). When vehicles communicate with each other (V2V), they are responsible for the direct communication, without relying on the roadside infrastructure. In the case of infrastructure to vehicle communication, the RSU acts as information source and internet connectivity provider to the surrounding vehicles. In addition, RSU can also expand the communication range by forwarding information to other RSU that will forward it to the surrounding OBUs (Weil, 2008). Experimental tests have shown that the DSRC performs well within a range of 300 meters in terms of latency and message loss due to high speed of vehicles (Z. Xu et al., 2017). The RTT (Round Time Trip) of the DRC communication has been tested to be under 100 ms and the PLR (Packet Loss Rate) is between 0.5% and 1% under vehicle speed changes from 60 km/h to 120 km/h. However, a key drawback of DSRC is its low scalability which hinders the performance in dense traffic. A potential solution to overcome this drawback and also expand the communication coverage is launching cellular-based LTE-Vehicle (LTE-V) networks. As a primary use-case of the next generation wireless networks (5G), LTE-V is now under live trail test stage in China and European countries. A third option has also been proposed, a one that envision the integration of both DSRC and LTE-V techniques (Abboud et al., 2014).

As it is, the vehicular communication technology requires vehicle to be equipped with a set of sensors, computing and storing devices to process real-time data provided by the sensors and wireless transceivers to communicate sensor data to the RSUs. In addition, the communication protocols used to communicate with the RSUs should follow a standard, independent of the automotive brands.

2.1.2. Vehicle automation

Vehicle automation involves a wide variety of technologies ranging from mechanics, artificial intelligence, and multi-agent systems. The International Society of Automotive Engineers' (SAE) has provided a systematic classification of vehicle automation((SAE, 2016)), which helps to map the level of automation with the requirements of the traffic solution. The classification defines six levels according to how the *dynamic driving task* is divided between the human driver and the machine. This task is performed entirely by a human driver at Level 0 (no automation) and entirely by an automated driving system at Level 5 (full automation). In the intermediate levels, the task is shared simultaneously or sequentially, requiring a necessary but critical human-machine interaction.

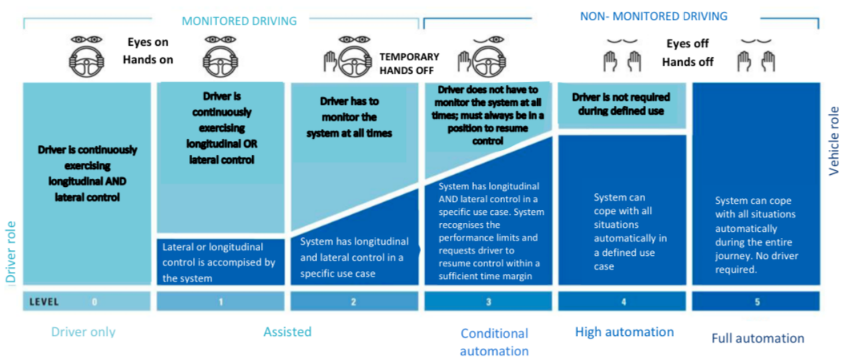


Figure 2.1: SAE classification of vehicle automation. Source (European Commission, 2018)

Vehicles capable of driver assistance fall in Level 1, such as adaptive cruise control or lane-keeping. Either the longitudinal or the latitudinal control is performed by the vehicle. When the assistance include both steering and acceleration/deceleration, vehicles are capable of partial automation (Level 2). Both of these levels assume that the human driver continues to actively monitor the driving environment. The introduction of conventional cars capable of operating without this active monitoring represents the threshold between partial automation (Level 2) and conditional automation (Level 3), defining also the line between non-automated and automated vehicles. Levels of automation beyond conditional automation can operate sorely on inputs coming from the vehicle's sensors and from other vehicles or infrastructure units. The difference between Level 4 (high automation) and Level 5 (full automation) is that the first is capable of operating in only some contexts or *driving modes*. Vehicles assisting the driver are already available on the EU market (levels 1 and 2) and automated vehicles that can drive themselves in a limited number of driving situations (levels 3 and 4) are being tested and some of them should be available by 2020 (European Commission, 2018). From a traffic management point of you, the interesting use cases of automated vehicles involve the application of level 3 or higher. When the system performs the driving tasks, the vehicle trajectories can be influenced.

The Society of Motor Manufacturers and Traders states that vehicles with some level of automation do not necessarily need to be connected, and vice versa, although the two technologies can be complementary (SMMT, 2017). It is likely that vehicles with autonomous capabilities will increasingly rely on connectivity (i.e. the ability to receive and transmit data) to achieve autonomy, and that technology convergence will result in vehicles that are both connected and autonomous (CAVs) (Lyons & Babbar, 2017). Part of the technologies that increase the level of automation are relying on the vehicle connectivity principles, such as cooperative adaptive cruise control (C-ACC) and vehicular platooning.

Under these consideration, this research will focus on the coexistence of two vehicle types: manually driven vehicles (Level 0-2), and connected automated vehicles (Levels 3-5). For convenience, these groups will be referred as *conventional vehicles*, and *automated vehicles* respectively. Conventional vehicles have no means to communicate with the environment (V2V or V2I) and even though they might have some driver assistant feature, their movement is determined by its human driver at all times. Automated vehicles support V2I and V2V communications and at least while driving in urban intersections, the driving task is assigned to the vehicle's control unit.

2.2. Traffic Control Systems

This section investigates the theoretical framework for traffic control systems from the perspective of traffic engineering and control engineering. This research provides the tools to facilitate a structured design process for the new traffic control, as well as support the analysis of the control approaches taken in literature.

2.2.1. System Components

A signalized intersection can be considered as a controlled system. As any system, the intersection can be defined as a process with input and output. A graphical representation is given in Figure 2.2. The input of the process consists of controllable $u(t)$ and uncontrollable $d(t)$ variables that influence its behavior. Uncontrollable variables are also called disturbances and often originate from the system's environment. One of the main influential uncontrolled variable is the traffic demand. The output $y(t)$ of the process consist of the measurable components of the system and are typically used for monitoring the system and evaluate its performance, e.g. speed, flow. The process itself consists in the traffic behavior of vehicles at the intersection.

The dynamics of the process are expressed in the state of the system $x(t)$. The state of a system is a mathematical quantity $x(t)$ that describe unambiguously the current and dynamic knowledge about the system at a specific instant in time, i.e. queue length. A future state $x(t+1)$ can be computed by applying a system equation that relates the state with inputs and signal control $x(t+1) = f(t, x(t), u(t), d(t))$. A second equation, called *measurement equation* relates the output $y(t)$ to the state $x(t)$, $y(t) = g(t, x(t)$. The formulation of the state $x(t)$ is dependent on the problem at hand and the models that are used to describe the system's behavior (Hegyi, 2012).

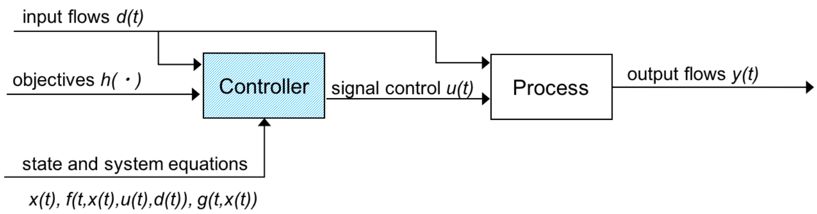


Figure 2.2: Signalized intersection as a controlled system

Based on the knowledge of the traffic demand and the traffic dynamics, the controller design the signal control $u(t)$ in such a way that the output of the process correspond to a desired behavior, reflecting desired objectives $h(\cdot)$.

2.2.2. Controller

The traffic system can be controlled by a number of different control methodologies. The type of controller implemented at an intersection depends on a number of aspects, namely the control objective, any existing constraints and the disturbances that act on the process.

Generally, there are two categories under which we can group the main goals of traffic controllers: improving the traffic condition and/or improving the environmental impact of vehicles (pollution). The control objectives are typically formulated in terms of measurable outputs $y(t)$ and sometimes in terms of estimated or predicted states $x(t+1)$. In regards to the traffic condition, common output-based objectives are minimizing travel time or delay and maximize the throughput or the speed. Common state-based objectives are queue length or waiting time. In dynamic traffic control measures, the choice of the control objectives often depends on the input information and measurements available. Constrains might exists due to physical limitation of the system such as space available for queuing at turning lanes or may describe the desired system behavior, such as obeying speed limits. Obviously, they have to be formulated in terms of the chosen objectives.

Most of the control methodologies available can be grouped in three categories, namely feed-forward control, feedback control and predictive control methodologies.

Feed-forward control methodologies incorporate information regarding only the measurable disturbances $d(t)$ to determine the control signal $u(t)$. The main advantages are that the complete system is stable if the controller and the process are stable, and that its design is in general simple. The disadvantage of this method is that if the desired system behavior is not achieved by the control signal, the controller is not informed and no action are taken.

Feedback control methodologies consider measurements of the process $y(t)$ or the state of the system $x(t)$ to determine the control actions $u(t)$. The feedback connection creates a closed loop in the block diagram. In case the state is used to determine the control signal is called *state feedback*. The advantage of feedback controllers is that the results of the control actions and the consequences of non-measurable disturbances will be used in the subsequent control actions. The disadvantage is that the feedback loop may lead to instabilities even if both the controller and the process are stable themselves.

Predictive control methodologies use forecasting models to predict the future state evolution $x(s), s \geq t+1$. Using state estimate and additional historical data, the model predicts the traffic operations for a specific future time horizon H. The predictive controller determines the actual control signal from these predictions.

Optimal control is a type of predictive control-based model. The main characteristic of this methodology is that an objective function J is explicitly optimized. This objective function describes the predicted performance of the system, starting from the current state $x(t)$. Multiple objectives can be integrated by means of weighting, which serves as useful tool to combines the needs of different stakeholders. When the optimal control methodology operates using a rolling horizon, it is usually called model predictive control (MPC).

2.3. Application of Control Strategy

This section provides an overview of how the theoretical framework of traffic control systems is being applied in the current and future traffic environments. In particular, it will focus on the literature that involves connected and automated vehicles.

2.3.1. History of Implemented Strategies

The most commonly implemented signal control strategies have gone through three different stages: fixed-time, actuated, and adaptive. Fixed-time control systems do not use any real-time information of vehicle arrivals. Instead, the signal timing plans are determined offline using historical traffic data, $u(t) = const, t \in T$. The drawback of the fixed signal control is obvious: it can not respond to any change in the traffic condition. Actuated traffic signal control systems use real-time measurements $d(t)$ provided by inductive loop detectors that are usually installed in the upstream proximity of the stop-lines. Each detected vehicle will generate a call to request additional green time $u(t) = f(d(t))$. Adaptive traffic signal control systems also use loop detectors to retrieve information $d(t)$, such as vehicle passages, occupation time and speed. These control systems employ prediction models to forecast the vehicle arrivals $\tilde{d}(t+1)$ and estimate queue lengths at each intersection $x(t)$. The control signal is therefore determined based on the estimated future state $u(t) = f(x(t), d(t), \tilde{d}(t+1),)$. The performance of the timing plan thus depends on the prediction accuracy.

Current practise in traffic management has shifted to actuated and adaptive schemes for isolated intersection or network-based optimization. While actuated signaling is a feed-forward control, the adaptive strategies use a feedback approach. One of the adaptive control system most known for its effectiveness is SCOOT (), developed by the British Government's Transport and Road Research Laboratory. SCOOT (Split Cycle Offset Optimization Technique)is also an example of on-line optimal control, as it entails the optimization of the signal control to reduced traffic delay and number of stops.

2.3.2. Proposed Strategies for Connected and Automated Vehicles

There is quite an extensive literature proposing traffic control strategies for connected vehicle and automated vehicle environments. The communication V2I brings new opportunities regarding input information $d(t)$ and measurement $y(t)$ the controller can receive in terms of new type of data and in terms of accuracy of such information. The automation of the driving tasks in connected vehicles allows to bring even more radical changes such as the usage of a different signal control $u(t)$, potentially leading to scenarios where traffic lights are obsolete. All these advantages are used to enhance the flexibility of the controller to the vehicles' arrival. Some research have chosen to do so within the limit of the traditional phase-based scheme, whereas others have proposed a new development stage of traffic control. Traffic is no longer considered collectively but it's individual vehicle-based. A taxonomy of the literature reviewed is provided by the Figure 2.3.

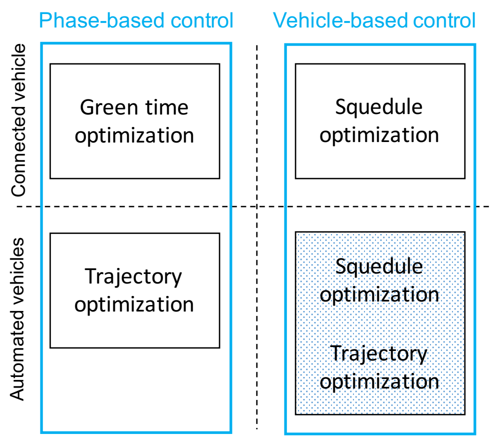


Figure 2.3: Taxonomy of proposed strategies for connected and automated vehicles

An example of the first category is given by (Feng et al., 2015), (Chang & Park, 2013). In this research connected vehicles are used to improve the performance of traditional stage-based control strategies. The vehicles function as dynamic sensors sending information about themselves (e.g. position,speed) or about surroundings vehicles. This information is used to calculate the state, usually queue length, and optimize the signal control based on this information. Some studies consider a traditional stage-sequence allocation and focus on optimizing the green time while others, like Priemer et al. (Priemer & Friedrich, 2009), allow for a more flexible allocation of phases. In this case restriction on minimum green light still hold. When automated vehicles are considered, researchers have improved the performance of traditional signal control by focusing on vehicle trajectory optimization (B. Xu et al., 2017), (Pandit, K; Ghosal, D; Zhang, H. M; Chuah, 2013). The aim is to make vehicles arrive timely at the green light with the minimal use of braking, while maintaining a safe distance between vehicles. In this case, the green light is a given variable. These trajectory-based algorithms benefit from the connectivity of automated vehicles to make preparations for the optimal departure timing and speed far ahead from the stop line.

The second category applies a new strategy that consider vehicles or groups of vehicles independently and schedule the control signal to be synchronized with their arrival. All connected vehicles-based control method still rely on the traffic lights system but loose the stage-based scheme and allow full flexibility of the phase allocation (Jia et al., 2007), (Pandit, K; Ghosal, D; Zhang, H. M; Chuah, 2013). Both phase allocation and green timing needs to be optimized. In this case vehicles are required to send more processed information, typically expected time of arrival. Studies based on the autonomous vehicle environment go even further and proposed a system without traffic light, relying on the timely execution of the crossing schedule by the vehicles. The first researchers to propose such system are Dresner and Stone (Dresner & Stone, 2008). Instead of phase-allocation and green time, their algorithm aim to coordinate the access to the crossing area by sorting the requests from upcoming vehicles. The vehicles are responsible to send the expected arrival time and compute the trajectory that allows them to arrive at the designated time. Under the same scenario, only few studies have faced both the scheduling problem and the trajectory control of vehicles (Ilgin Guler et al., 2014),(J. Li et al., 2016). Most of the research based on either connected or automated vehicles consider a full-penetration scenario. Only few connected vehicle-based studies consider different penetration rates (Chang & Park, 2013), (Ilgin Guler et al., 2014).

As mentioned in the Introduction chapter (Section 1.1), this thesis aims to develop a strategy that take full advantage of the communication and automation technologies, by relying purely on wireless communication (no traffic light). For this reason, this research falls into the category of vehicle-based control within an automated vehicle environment (blue square in Figure 2.3). Following this section, only literature pertinent to this category will be considered.

2.4. Optimization Problem

The relevant literature reviewed is mainly focused on optimization-based methods to solve both phase allocation and trajectory control. This section aims to gather more knowledge about the design of the optimization problem along with the techniques to solve it. In particular, it will focus on the optimization of the scheduling problem and trajectory control separately.

2.4.1. Signal Control Framework

In their review paper, Jing et al. (Jing et al., 2017) have attempt to capture the essence of optimizing the signal control within the connected and automated vehicle environment controlled by an adaptive traffic light scheme. An adaptation of such framework to fit a traffic light-absent environment is provided in Figure 2.4.

The framework is divided into three modules: input, optimization, and output. The first step of the optimization process is to calculate the traffic state. The inputs are the real-time coming from the vehicles, such as position and speed. The output of the first step is

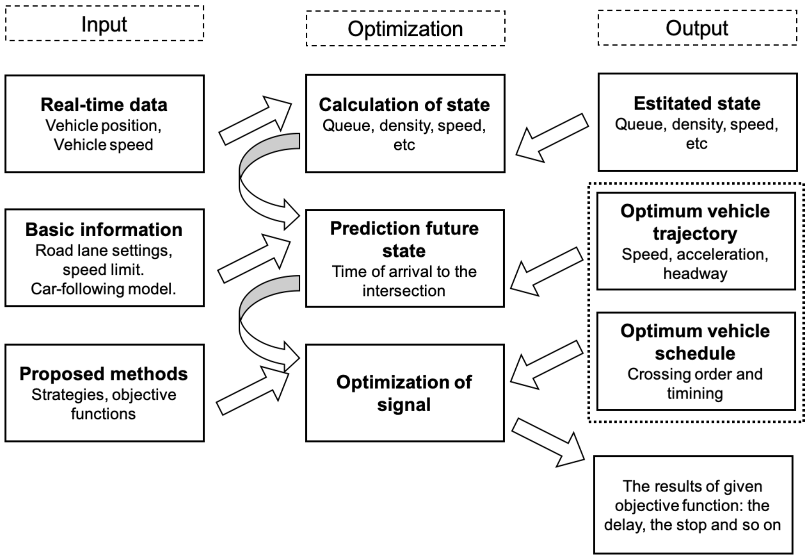


Figure 2.4: Framework of optimization of signal control for connected and automated vehicles

the estimated state. Following, the calculation of the future state is computed, in particular at the time when vehicles arrive at the crossing area. For this step additional information are required: basic intersection information, models for the prediction, such as car-following models or queuing models. For conventional vehicles the future state is predicted while for the automated vehicles it is computed. Thus, the ultimate output is the optimum vehicle trajectories. At last, the optimization of the control signal is executed based on defined strategies and objectives. As a result, the optimum vehicle schedule is obtained.

The optimization algorithms used in the reviewed literature aims either at minimizing the vehicle delay or at optimizing the vehicle queue.

2.4.2. Scheduling Problem

Several studies that aim to optimize the schedule of vehicles have considered the scheduling problem as a single machine scheduling problem. The isolated intersection is modelled as a single machine that can process parallel jobs. Each vehicle is modeled as a *job* and its arrival time and passing time are modeled as the job *release date* and *processing time*, respectively. Parallel jobs are vehicles that come from non-conflicting directions. The time each job should finish the traversing procedure without any delay is modeled as the job *due date*, which equals to the vehicle arrival time plus its passing time. Literature concerning this scheduling problem frequently propose two techniques to solve it: dynamic programming or branch and bound algorithms (Potts & Kovalyov, 2000).

Two of the main proprieties of the dynamic programming are overlapping sub problems and memorization. The overlapping means that the sub problems are solved by dividing into similar sub-sub problems until we reach the initialization condition. During the process, the same sub problems maybe appear several times. Instead of computing them every time, the solutions to sub problems are saved retrieved to solve the same problem in a later time. This is called memorization. Wu et al use dynamic programming to solve their scheduling problem (Jia et al., 2007). They allocate the right-of-way to different road directions according to the arrival time of vehicles. The passing time of all vehicles is assumed identical. The objective is to free the resource as soon as possible. By solving the scheduling problem with dynamic programming, time complexity is said to be $O(n^2)$), where n is the number of vehicles.

Yan et al. (Yan et al., 2008) has a similar problem than Wu et al. and they also considered to solve it with dynamic programming. Even though they recognize its advantages, they also point it out that this approach needs to consider all the possibilities before finding the optimal solution. If the number of the vehicles fluctuates to a high value and the situation of the intersection become congested, the algorithm will take a long time to find the optimal solution (the optimal sequence of the vehicles passing the intersection). Considering this, the authors opt for a technique able to narrow down the solution space so it is not necessary to calculate all the possibilities before obtaining the optimal solution. Such technique is the branch and bound algorithm. In this algorithm, the set of candidate solutions is thought of as forming a rooted tree with the full set at the root. The algorithm explores branches of this tree, which represent subsets of the solution set. Before enumerating the candidate solutions of a branch, the branch is checked against upper and lower estimated bounds on the optimal solution, and is discarded if it cannot produce a better solution than the best one found so far by the algorithm. The reducing rule that defines the lower bounds is essential to ensure efficient computational time. In a later study, Yan et al.(Yan et al., 2011) propose two algorithms to define the lower bound and initial upper bound along with a mathematical proof of their principles. Yan et al. (Yan et al., 2008) give also an extensive analysis on the computational effort of the branch and bound technique within the application of optimizing real-time traffic control. In the worst case, the search algorithm have to expand all the nodes in the search space. On the other hand, the complexity may be linear from the root node to a goal node if the lower-bound function is exact. However, these two extreme cases rarely occur. Several test are then carried to evaluate the average complexity of the branch and bound. All the computational experiments were run on a Core 2 computer with two 1.86GHz of CPU, 2GB of RAM and Linux system. For headway of 5 s, the branch and bound allows to solve more than 200 vehicles within 1s, this makes it possible to develop new dynamic control system in real-time.

2.4.3. Trajectory control

In general, the principle of trajectory control is to take advantage of the automated driving and integrate trajectory design into the signal control scheme. Some research let the vehicles optimize their own trajectories based on the scheduling signals received in advance through V2I communication. Other researchers assign the trajectory optimization to the intersection controller which send to the vehicles a speed or acceleration profile. In the latter case, the collision avoidance is either ensured by the constraint of minimal arrival time or is explicitly part of the objectives in the trajectory optimization. Vehicle trajectories can be designed to minimize the evacuation time (i.e. maximize arrival speed), number of stops or to minimize emissions.

Katsaros et al. (Katsaros, 2011) implement a GLOSA system to reduce traffic congestion by decreasing the average stop time behind traffic lights while reducing fuel consumption and CO2 emissions. The GLOSA application provides the advantage of timely information about traffic lights cycles through infrastructure-to-vehicle (I2V) communication, and provides drivers with speed advice guiding them with a more constant speed and with less stopping time through traffic lights. The speed advice is based on a very simple scheme. If the traffic light is green when the vehicle reaches it, then the vehicle continues its trip trying to reach the maximum speed limit of the road. If it is red, it calculates the speed that it should have in order to reach it in the next green phase. If it is yellow, depending on the remaining yellow time and the acceleration capabilities of the vehicle it could advice to accelerate or decelerate again within the permitted range. The calculation of arrival time and target speed is based on the current speed and acceleration of the vehicle using basic rule of motion.

In its research, Busse (Busse, n.d.) defines the optimal trajectory with the highest possible speed at the arrival. The speed profile is defined by five parameters of which four are known either from the vehicle or from the controller and only one parameter needs to calculated. As a result, each vehicle decelerate with their minimum acceleration until the calculated time when they proceed with constant speed. This method is very simple and computational effective, however the simulation results it is not clear if the strategy proved to be successful.

A similar approach on the trajectory control is given by (J. Li et al., 2016) J.Li et al. In this case, after receiving the arrival time from the controller, the optimal speed profile for each vehicle is found for the minimal fuel consumption. The speed profile is also in this case defined by a set of parameters some of which are given and some needs to be calculated. The calculations are based again on basic motion relationships. In respect to the research of Busse, Li et al. consider a more complex speed profile, where vehicles decelerate, cruse and then accelerate again to increase the speed at which the intersection is crossed. A case-based approach depending on the initial condition of the vehicles is used to provide the equations of the parameters.

In their research, Li et al (Z. Li et al., 2014) make interesting consideration regarding the parameterization of the trajectory. The optimized trajectory is defined by a acceleration profile. The more segment the profile has, the flexible it is. A single component trajectory can be controlled either by the final speed or by the travel time, but not both. In order to schedule the vehicle arrivals separated by a desired safe headway and to have them accelerate when they reach the intersection, both parameters must be controlled. Using two-component trajectories is the minimum required in this case. If the vehicle cannot be scheduled to reach the intersection with the maximum speed, it has to accelerate to the maximum speed in the downstream road (after the intersection). This results in three-component trajectory, which provides considerable flexibility for controlling vehicle arrival time and arrival speed.

2.5. Final Considerations

The literature study presented in this chapter has helped clarify some aspects of the thesis scope as well as providing a general understanding of the traffic control design.

Based on the investigation on communication and automation technologies, it was decided that this research will focus on the coexistence of two vehicle types: manually driven vehicles (Level 0-2), and connected automated vehicles (Levels 3-5).

Literature that contained studies with similar high-level control strategy as foreseen in this study, gave insights in the design possibilities. A common signal control framework for optimization-based methods has been constructed, providing a basis guideline for the design process.

Furthermore, literature found on the scheduling problem and trajectory control appear to be very useful. Two main approaches are highlighted to solve the scheduling problem; the dynamic programming and the bench-and-bound algorithms. Given the potential of solving the scheduling problem with a growing amount of traffic when applying the bench-and-bound algorithms and the complexity of the dynamic programming, it was concluded that the bench-and-bound algorithm was a proper fit for this study. Subsequently, among the literature about trajectory control it became clear that a common approach is to parameterize the trajectory, so that its computation occurs in a efficient way.

CHAPTER 3

Controller Design

Chapter 2 showed how the level of automation impacts the components of the intersection environment and how these components lead to certain design choices for the controller. Therefore, this chapter starts with an analysis on the characteristics of the intersection meant to be controlled in this research. The analysis resumes the first description given in Section 1.2 and delineate the intersection environment in scope in more detail. Following this description, the solution framework is presented, explaining how the traffic is regulated and how the coordination of the vehicles is designed. In order to evaluate the final performance of the controller, a simulation is performed. The description of the evaluation methodology is given in Chapter 4.

3.1. Intersection Environment

In the Introduction chapter, the research scope was defined (Section 1.2) providing a high-level description of the intersection environment. Such environment is depicted in Figure 3.1 and consists in an urban intersection that relies purely on wireless communication to coordinate right-of-ways of the traffic without the need of traffic lights. This section describes in detail the components of the intersection environment depicted in Figure 3.1. Some of the components of the proposed scenario are not yet fully implemented in real-life, namely the automated vehicles and the wireless communication between vehicles and traffic controllers. The interaction of these components with their environment is also not known and constitutes a research topic on its own. For these reasons, if the interaction is relevant for the controller design, some assumptions will be introduced.

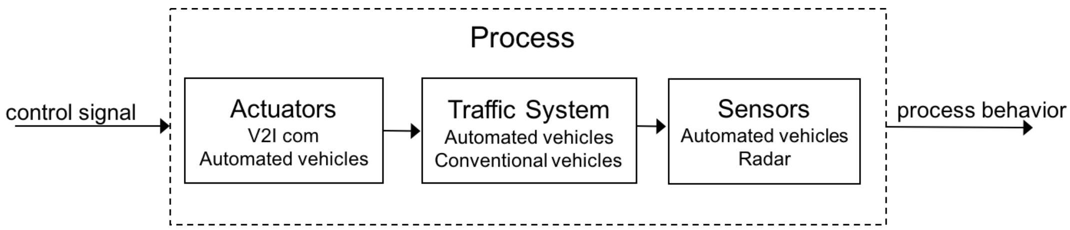


Figure 3.1: Intersection environment considered in this research

3.1.1. Actuators

No traffic lights are used to display the signal control at the intersection. V2I communication is used to send individually the signal control to the automated vehicles. Upon receiving the signal, the vehicle controller of the automated vehicle will use it as input to execute the next vehicle movement. This communication is assumed to be instantaneous and continuous in space while working under standard conditions in an urban environment. The intersection controller is the only authorized party to control the trajectory of the automated vehicles. It is assumed that this authorization can be overruled by the receiving vehicle itself if the execution of the signal control will jeopardize the safety of the vehicle or its surrounding environment. This assumption is the equivalent of a conventional car stopping in front of a green light when their preceding vehicles suddenly brake for no obvious reason.

The automated vehicles are in turn, used to *display* the signal control to the conventional vehicles in order to influence their movement. A common notion in traffic flow theory is that in free flow conditions, the speed of a conventional vehicle will fluctuate around the desired speed of the driver. This desired speed is hindered by the driving behavior of the preceding vehicle when the distance between the preceding vehicle and its follower (headway) is closing up. This happens when the preceding vehicle has a lower speed than its follower. In response, the follower will lower its own speed in order keep a safety headway. Once the leading vehicle is accelerating, the follower can chose to either match the leader's behavior with an acceleration of its own, or continue to drive with its current speed. Hence, the speed of a conventional driver is directly influenced when the preceding vehicle is driving within a threshold headway.

3.1.2. Traffic System

The traffic system is composed of conventional cars and automated vehicles. No pedestrian or cyclist are considered. Technically, automated vehicles are able to keep close distances to their preceding vehicles because the vehicle controller overcomes the faults of drivers, e.g.fatigue, lack of attention, late response. In practise, however, the coexistence of conventional vehicles and automated vehicles might limit the ability of the latter as their behavior can be perceived unsafe by the conventional vehicles. With respect to the ability of the automated vehicles, the worst case scenario will be assumed: the driving behavior of automated vehicles follows the same safety rules of conventional vehicles. The speed limit is set at 50 km/h.

Since traffic lights are removed from the intersection, conventional vehicles are expected to apply the right-of-way rule when passing the intersection at the crossing area.

3.1.3. Sensors

The intersection is equipped with detection technology able to provide accurate information about the traffic condition. Radar detection technology allows tracking vehicles within its range of sight and provides position, speed, acceleration, length and id of each object observed. The only information it cannot provide is the vehicle automation level. This is solved by merging the position information retrieved from the automated vehicle with the measured position of vehicles detected by radars. Within this set-up, the coverage, frequency and accuracy of vehicle measurements relies mainly on the radar technology capabilities. On average, a single radar can guarantee a minimum line of sight of 200 m for a fairly straight road with multiple lanes. New detection is supplied at least every 50 ms. Considering that speed limit is 50 km/h, between two measurements, vehicles driving at maximum speed traverse 12 m approximately. The precision of detection is considered acceptable and the measurement error is not taken into account.

3.1.4. Summary of Assumptions and Specifications

This paragraph summaries the assumptions and specifications chosen for the intersection environment of this research.

Environment assumptions:

- The automated vehicles overrules the control signal if the execution of the signal jeopardize the safety of the vehicle and its surrounding environment.
- The driving behavior of automated vehicles follows the same safety rules of conventional vehicles.
- The crossing behavior of conventional vehicles follows the right-of-way rule.
- The V2I communication is assumed to be instantaneous and continuous in space.
- The detection coverage and accuracy of vehicle measurements relies mainly on the radar technology capabilities.
- The measured vehicle data consist of position, speed, acceleration, length, id and automation level.
- The precision of detection is considered acceptable and the measurement errors not taken into account

Environment specifications:

- No traffic lights are needed at the intersection.
- No pedestrian or cyclists are considered in the intersection.
- Speed limit of 50 km/h.
- Detection of vehicles is possible within a range of 200 m from the stop-line.

3.2. Solution Framework

After reviewing the intersection environment at scope for this research, the rest of this chapter is dedicated to illustrate the control solution. This section introduces the underlining concept chosen to regulate traffic. First a high-level description of the control strategy is provided, followed by an introduction of the intersection model. The intersection model shows how the controller interprets the intersection environment and its description is divided in infrastructure model and traffic model.

3.2.1. Control Strategy

The control strategy relies on a combination of self-organization and individual coordination.

The controller receives a complete traffic image in real-time from the radar sensors. Based on this information, the incoming traffic is grouped in induced platoons. An induced platoon is formed between vehicles with a headway small enough to assume that the driving behavior of the vehicles is significantly influenced by their leader. Vehicles within the platoon drive along the road as single unit by unintentionally *induce* their own speed to their followers, starting from the leader of the platoon. For this reason, the platoons are referred as *induced platoons* (i-platoons). If the leader of an i-platoon is an automated vehicle, the i-platoon is defined *controlled*, *uncontrolled* otherwise. The controller interprets traffic in terms of i-platoon instead of vehicles. The controller uses the real-time information on all i-platoons to schedule the crossing of controlled i-platoons via V2I communication. The scheduling of controlled i-platoons aims to minimize the delay of all i-platoons. Once the optimum schedule is defined, a trajectory control reinforce this schedule by guiding controlled i-platoons through the intersection. Adjusting their movement to the advised trajectory, the controlled i-platoons can evacuate safely the intersection. The uncontrolled i-platoons enter the intersection on their own accord, following standard traffic rules at an unsignalized intersection. Since the schedule of controlled i-platoons is optimized based on delay of all i-platoons, also uncontrolled i-platoons can experience a benefit. For example, when a uncontrolled i-platoon is driving on a minor road and a controlled i-platoon is driving on the main road, the optimum schedule might lead to slow down the controlled i-platoon so that the uncontrolled i-platoon has enough time gap to cross the intersection first.

The formation of i-platoons, both controlled and uncontrolled, can variate over time. The condition that keeps an i-platoon together depends on several factors, including both on the behavior of the driver and on the current traffic state. The merging of two i-platoons can in fact happen due to an uncontrolled i-platoon closing up to its preceding i-platoon or due to an i-platoon encountering an i-platoon queuing at the stop line. The controller adapts its signal according to the dynamics of the induced i-platooning by introducing a feedback loop that periodically re-form i-platoons and re-optimize the crossing of current controlled platoons.

It is believed that a system making use of smart combination of self-organization and individual coordination performs better than failed self-organizing intersections solved by classic traffic light controlling. In case of fixed schedule, the advantage comes from the ability to adapt to the current situation while the fixed scheme cannot. In case of more advanced scheme, the combined strategy should perform better because it offer the possibility to adapt with more flexibility to the vehicle arrival of all direction considered, without following a preset cycle. In case of an actuated scheme, which is more flexible to the vehicle arrivals, the advantages of the proposed solution comes the lack of limitation in the crossing sequence. Actuated scheme have some flexibility in giving green time earlier or later than planned according to the real-time arrival of vehicles, however they are bound to the a cycle structure and some fixed parameters like minimum green time.

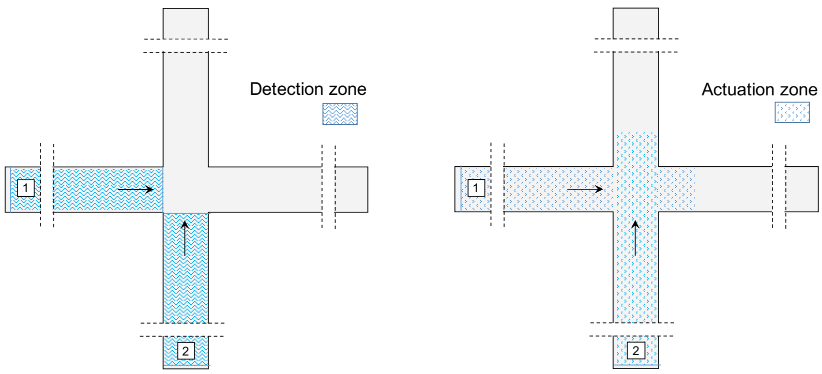
The success of this combined control strategy has a commensurate dependence on the percentage of controlled i-platoons in the traffic. The more controlled i-platoons there are, the more flexibility the system has to coordinate the crossing with the least total delay. The fraction of controlled i-platoons has also an effect on the efficiency of the trajectory guidance. Since uncontrolled i-platoons are allowed to increase and decrease their speed whenever they want, they are not as good predictable in behaviour as controlled i-platoons. For this reason, it is more likely that the time of entry at the intersection by an uncontrolled i-platoons is later than predicted, which causes on its turn an additional decrease in speed of the controlled i-platoons entering the intersection from a conflicting stream. Induced platooning was in fact introduced in order to increase the percentage of controlled traffic by allowing automated vehicle to function as actuators to conventional vehicles. Where traffic volume is high, the probability of making i-platoons increases, whereas a high penetration rate of automated vehicles increases the probability of forming controlled i-platoons. Moreover, when traffic volume is high there is less probability that i-platoon can be served without being hindered in their journey. Because more i-platoon have to be decelerated, the speed of the following i-platoon will also be affected and overall density increases. In this situation, the percentage of controlled i-platoon is essential in determining how quickly the situation is recovered. Therefore, it can be stated that the efficiency of this strategy is mainly depending on two external factors: the traffic volume and the penetration rate.

3.2.2. Intersection Model: Infrastructure

This section describes how the intersection's infrastructure is modelled. A simplified layout will be considered in order to reduce the complexity of the control design.

The intersection consists of two approaching road and the crossing area. All roads have only one lane, so overtaking is not possible. At the crossing, only the straight directions is allowed. The lane identification is done in respect to the origin and not to the destination because when vehicles are detected their desired direction is yet unknown. This consideration is obsolete for the layout just introduced, since each origin has only one destination possible respectively. However, if more allowed destinations would be considered, this labeling order becomes more meaningful as it suggests how to coordinate vehicles without knowing the destination. In this case, vehicles are processed sequentially by origin. Investigation of this case should be subjected to future research.

Each road is equipped with a radar sensor with an average detection range of 200 meters. Vehicles approaching the intersection are detected and tracked from 200 m upstream the crossing area until the beginning of the crossing area. The zone defined within these boundaries is referred as *detection zone*(Figure 3.2a)



(a) Detection zone *(b)* Actuation zone

Figure 3.2: Schematic representation of the intersection model

The actuation zone defines the area where automated vehicles are actuating the signal control received via V2I communications (Figure 3.2b). The signal control is sent to the automated vehicles for which the controller computed the signal, hence to the vehicles detected by the radar sensor. The boundaries of the actuation zone are defined by the range of the V2I communication, the range of the radar detection and the signal control itself. Since the first was assumed to be continuous (Section 3.1.1), and the detection and signal control are simultaneous, the start of the actuation zone concise with the start of the detection zone at 200 m upstream of the intersection. On the other hand, the end of the actuation zone is bounded only by the signal control itself, which is computed from the moment measurement are available to the time the platoons leaves the intersection. Hence, the end of the actuation zone varies for platoon and correspond to the end of the intersection plus the length of the platoon measured from the front of the leading vehicle to the rear of the last following vehicle.

Due to the layout simplification, the control design The simplification of the intersection layout affects the control design in two ways: the induced platooning only has to consider the longitudinal motion of vehicles and the scheduling problem only has to consider conflicting stream with singular known destinations.

3.2.3. Intersection Model: Traffic

As previously mentioned, the controller interprets traffic in terms of i-platoon instead of vehicles. This section provides a deeper insight on the concept of *induced platooning*, describing how the i-platoons are formed and most importantly how their driving behavior is modelled.

Definition

Induced platooning has the main purpose of grouping conventional vehicles with automated vehicles so that the latter can guide their followers through the intersection efficiently. In addition, considering batches of vehicles instead of considering vehicles individually reduces the computation effort for the prediction and coordination of the traffic.An induced platoon is defined as a cluster of vehicles with time headways h small enough to assume that the driving behavior of the vehicles is significantly influenced by their leader.

Formation

Three different conditions are set for the formation of i-platoons, considering their purpose in the control strategy: the headway condition, the length condition and the leader condition. The first one set the basic rule which is always followed expect in the two cases defined in the length and leader condition. Each condition is explained individually and illustrated by a practical example.

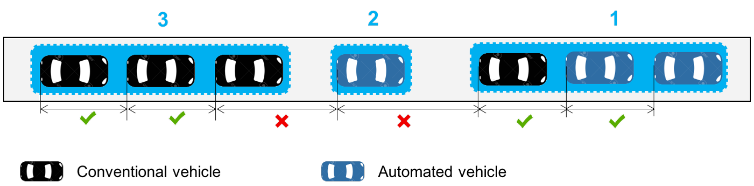


Figure 3.3: Practical examples of headway condition

The **headway condition** states that two following vehicles can be part of the same i-platoon if the follower have a headway within a threshold value. The headway is the time distance between the rear bumps of two successive vehicles. The threshold value of the headway is not an absolute value but varies within a range, according to several factors, e.g. the driving behavior of different drivers, vehicle's characteristics and variation within the behavior of a single driver. At same headway, it is possible that a vehicle is driving at its desired speed while another vehicle is driving at that speed as reaction to the behavior of the vehicle in front. In literature, many studies have been made on the headway distribution among the drivers population. One of the proposed models was given by Buckley (Buckley, 1968) where the total headway is obtain as the sum of two variables: the headway of vehicles that are following, and the headway of the vehicles that are driving freely. In (Hoogendoorn, Sergie P. ; Botma, 1997) an estimation of this headway function from headway observations can be found. Results show that the contribution of "following" headway ranges between 0.4 s to 4 s with a mean of 1.69 s. For the implementation of the control strategy in this research, a upper bound of 2.5 s is considered. In Figure 3.3, the i-platoon number 2 and number 3 are formed because the vehicles have headway longer than the threshold value. It should be noted that if vehicles arrive at a stand still, their headway is undefined, as the space headway is divided by a null speed. This means that stopped vehicles form individually a single-vehicle i-platoon.

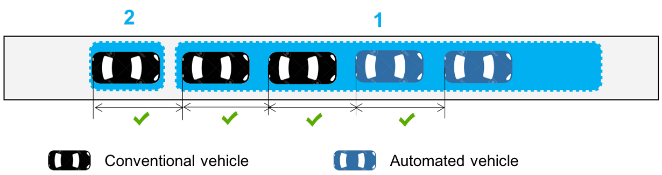


Figure 3.4: Practical examples of length condition

The **length condition** refers to the number of vehicle that can form a i-platoon. The length of an i-platoon influence the throughput of the intersection: the longer it is, the lower the throughput is. Thus, an experimental limit of 5 vehicles is introduced (Figure 3.4). This condition becomes important when density increases either due to increase of flow or decrease of average speed. If this condition didn't exist, the controller would consider traffic in the dense direction as a single inconveniently long i-platoon.

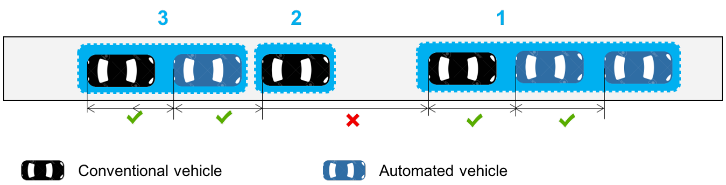


Figure 3.5: Practical examples of leader condition

The **leader condition** refers to which vehicle can be leader of an i-platoon. As mentioned in the Section 3.2.1, the performance of the control strategy is expected to be influenced by the fraction of controlled i-platoons present. In order to increase such value, additional condition in forming i-platoons is considered: a new i-platoon is always formed when an automated vehicle enter the detection zone, unless it meets the headway condition and its predecessor is an automated vehicles. Looking at the example in Figure 3.5, if only the headway condition would be applied, one controlled i-platoon (number 1) and one uncontrolled i-platoon (number 2) are formed. By introducing the leader condition, the automated vehicle following the leader of i-platoon 2 forms its own i-platoon (number 3), leading to a total of two controlled i-platoon and one uncontrolled i-platoon.

Behavior

The induced platoons are consider as a cluster of vehicles moving through the intersection as single unit lead by its leader. Therefore, given an induced platoon i formed of N vehicles with leader i^1 at time t, the characteristics of the i-platoon are defined as follows:

$$i = j \quad \text{with} \quad j = i^1 \tag{3.1}$$

$$d_i(t) = d_i^1(t) \tag{3.2}$$

$$l_i(t) = d_i^N(t) + l_i^N - d_i^1(t) + \tag{3.3}$$

$$v_i(t) = v_i^1(t) \tag{3.4}$$

$$tl_i(t) = \frac{l_i^N}{v_i(t)} + \sum_{n=2}^{n=N} h_i^n(t) \tag{3.5}$$

Each i-platoonis labelled with the id of its leader. The id of the leader is a ordinal number j, meaning that that vehicle is the $j-$th vehicle to enter the detection area. The annotation i^n is used to indicate vehicles that are part of the i-platoon i, $n \in [1,5]$ (length condition). The position d_i and speed v_i of the i-platoon i are determined by the leader i_1. The length l_i is the difference between the position of the head bump of the leader and the rear end of the last vehicle i^N. The variable tl_i is defined as *time length* and it corresponds to the sum of the headways of the vehicles part of the i-platoon plus the last vehicle length divided by its speed.

The time length is an important variable. When a i-platoon is composed by only automated vehicle the length tl_i is kept constant at all times. If the i-platoon is not formed only by automated vehicles, this statement does not hold at every time s with $s \in [t, t_{\text{exit}}]$. Under the assumption that vehicles within a i-platoon are voluntarily following their leaders, they adjust to the changes of speed and try to keep their headway constant. It is assumed that under certain speed condition the statement will hold within an acceptable degree of approximation. The assumption is formulated as follows:

$$tl_i(s) = const \quad \text{for} \quad v_i^1(s) \leq v_i^1(s) \quad \text{with } s \in [t, t_{\text{exit}}] \tag{3.6}$$

3.2.4. Summary of Design Assumptions

This paragraph summaries the assumptions made during the design process of the controller. Some have already discussed in the previous paragraphs of this section, other will be addressed later in this chapter.

Design assumptions:

- I-platoons have a time length constant while they drive through the actuation area with constant speed.
- All vehicles coming from a non-priority road perform a full stop at the start of the crossing area and their arrival speed at the crossing is zero.
- Uncontrolled i-platoon do not split up when they decide to drive through the crossing area.
- The probability of the gap acceptance is assumed to be deterministic.

3.3. Optimization Problem

This section describes the solution framework in mathematical terms, providing a formal overview of the problem to be solved. Following this introduction, the actual algorithms used to tackle the optimization problem are illustrated.

3.3.1. Problem description

The **disturbance** of the system is the arrival of vehicles upstream to the intersection, namely farther than 200 m to the crossing area.

The **measurements** $Y(t)$ consist of id, bound direction bd from which they are driving, position d, speed v, length l and automation level at of all vehicles detected within the detection zone. The position of the i-platoon is provided as distance to the stop line of the intersection. The automation level is a binary variable, 1 for automated vehicles and 0 for conventional vehicle.

$$Y(t) = \begin{bmatrix} \vdots \\ y_j(t) \\ \vdots \end{bmatrix} \quad \text{with} \quad y_j(t) = < j, \mathrm{bd}_j, d_j(t), v_j(t), l_j, \mathrm{at}_j > \tag{3.7}$$

The **state** of the system $X(t)$ is defined by the vector of all states of the i-platoons present in the detection area. The state of an induced platoon $x_i(t)$ with id i is defined by its position, speed, time length lt and control status c.

$$X(t) = \begin{bmatrix} \vdots \\ x_i(t) \\ \vdots \end{bmatrix} \quad \text{with} \quad x_i(t) = < i, \mathrm{bd}_i, d_i(t), v_i(t), tl_i, c_i > \tag{3.8}$$

The **control variable** is the crossing sequence O^{cp} of the controlled i-platoons. The crossing sequence is the order in which the i-platoons enter and clear the crossing area. According to this order, each i-platoon is assigned a cardinal number o_i representing the ordinal position of the i-platoon i in the sequence, e.g. $o_i = 1$ for the the first i-platoon to enter. The annotation $(\cdot)^{cp}$ is added to highlight the fact that the control variable refer only to the controlled i-platoons. The uncontrolled platoons will also be associated with a crossing order o_i, however these variables are not controllable but their knowledge is essential to the signal computation for the controlled i-platoons.

$$O^{cp} = < \ldots, o_i, \cdots > \qquad \text{with} \quad c_i = 1 \qquad o_i \in \left[1, \ldots, N^{cp}\right] \tag{3.9}$$

The **control signal** $U(t)$ is the acceleration profile $\bar{a}_i$ for the leader of the controlled i-platoon. The acceleration profile specifies the trajectory of the i-platoon from the moment the leader receives the signal t (assumed instantaneously) until the moment the i-platoon leaves the intersection.

$$U(t) = \begin{bmatrix} \vdots \\ u_i(t) \\ \vdots \end{bmatrix} \quad \text{with} \quad u_i(t) = \bar{a}_i \quad \text{and} \quad c_i = 1 \tag{3.10}$$

The **control objective** is to optimize the control signal $U(t)$ such that the resulting total delay of the i-platoon in the network is minimal. The total delay of the i-platoon is function of the current state $X(t)$, the estimated future state $\hat{X(t)}$, the control variable O^{cp} and of course the control signal $U(t)$. The optimization problem can therefore be formulated as:

$$\min_{U^{\text{cp}}} f\left(X(t), \hat{X}(t), O^{\text{cp}}, U^{\text{cp}}\right) \tag{3.11}$$

3.3.2. Optimization Layer

The formulation of the control objective provided by the equation 3.11, reveals the complicated nature of this optimization problem. On one side, the fact that i-platoon are considered individually brings a discrete aspect to the problem. The discreteness is related in particular to the order in which i-platoon can cross the intersection. Given the fact that there is no periodic phase plan determining the access order, any order of a given set of i-platoon coming from all streams is possible (within certain constraints). Moreover, a given crossing order is not associated with one control signal for controlled i-platoons. For each ordered sequence, there are multiple acceleration profiles that respect that order, for each controlled i-platoon involved. The optimization objective of minimizing the total delay would entail finding out, out of all associated control signals associated to all possible order sequences, the control signal that yields the minimal delay for all i-platoons present at the intersection. In order to solve this problem, the optimization is divided in two layers: an upper layer and a lower layer. The upper layer consists in finding the sequence with the least delay. In the upper layer, each sequence is associated with the control signal which generate the least delay. This optimization is solved in the lower level, where, given that sequence, the best acceleration profile is computed. An overview of the optimization of the controller is given in Figure 3.6. Vehicle

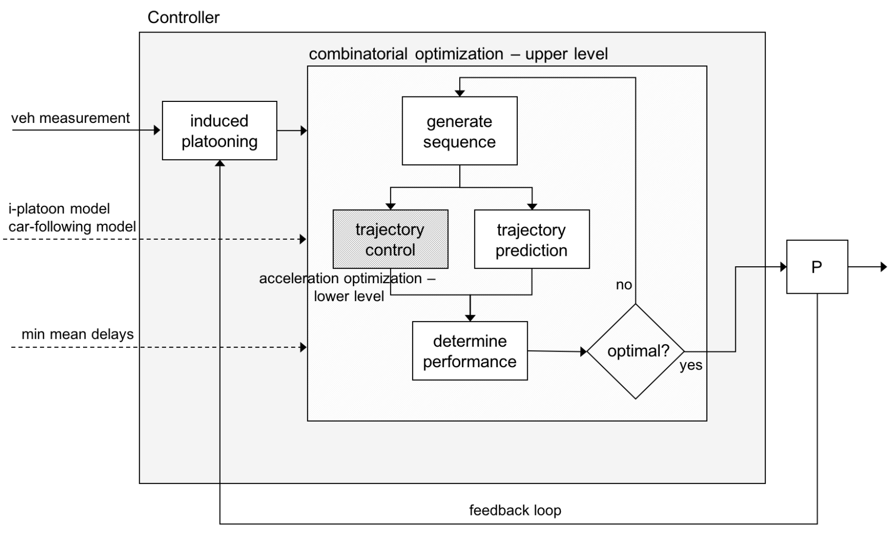


Figure 3.6: Schematic representation of the Control Optimization

measurement are used to form i-platoon, the state of the i-platoon is then sent to the optimization module. As mentioned before, the acceleration profile (lower layer) concerns only controlled i-platoons whereas the upper layer consider all i-platoons present. In order to be able to compute the total delay of a sequence, the necessary information from uncontrolled i-platoons also needs to be calculated. For this reason, the module *trajectory prediction* is added in the optimization scheme.

3.3.3. Lower layer Optimization

The lower layer optimization aims to find the acceleration profile of each controlled i-platoon, given an order sequence O', that yields the minimum total delay. The total delay TD is defined as the sum of the individual delays of all the i-platoons P present at the detection area. The individual delay is defined as the difference between the time an i-platoon leaves the intersection $t_{i,\text{ex}}$ and time it would have left the intersection if it was driving undisturbed $t_{i,\text{ex,min}}$.

$$TD(O') = \sum_{i\in P} TD_i(O') = \sum_{i\in P} t_{i,\text{ex}}(O') - t_{i,\text{ex,min}} \tag{3.12}$$

For a given sequence, the minimum total delay equals to the sum of the minimum individual delays. In the upper layer optimization, when the total delays of different sequences are compared, the sum of the terms $t_{i,\text{ex,min}}$ is equal in all sequences because the term is independent of the order. The terms therefore becomes obsolete:

$$\min TD(O') = \sum_{i\in P} \min t_{i,\text{ex}}(O') \tag{3.13}$$

The exit time of an i-platoon can be considered as the sum of the entry time and the evacuation duration. The evacuation duration corresponds to the time i-platoon needs to pass through the crossing area from the moment it enters it to the time it leaves it with the rear end of its last vehicle. The evacuation duration is then a function of the state $\hat{x}_i(t)$ of the i-platoon when it enters the intersection (entering speed) and the acceleration profile during the crossing. The entry time is also a function of the acceleration profile from the current time t to time of entrance, together with the current state $x_i(t)$:

$$t_{i,\text{ex}} = t_{i,\text{en}} + T_{i,\text{evac}} = f(x_i(t), \hat{x}_i(t), \bar{a}_i) \tag{3.14}$$

Finally, the lower layer optimization can be formulated as:

$$\min TD(O') = \sum_{i\in P} \min_{\hat{a}_i} f(x_i(t), \hat{x}_i(t), \bar{a}_i) \tag{3.15}$$

The **constraints** for this optimization problem relates to safety requirements and operational requirements. The safety requirement consists in denying the occupation of the crossing area by more than one i-platoon at the time, coming from conflicting streams. Hence, the entry time of an i-platoon needs to be equal or higher of the exit time of the preceding i-platoon in the crossing sequence. A time buffer is introduced to strengthen the prevention of head to head collision. The entry time of platoons from the same stream should be separated at least with a safe headway, such that rear-front collisions are prevented. The safety constraints are formulated as follow:

$$t_{\text{en},i} \ge t_{\text{ex},m} + t_{\text{buffer}} \qquad \text{for} \quad o'_i \ge o'_j, \quad \text{bd}_i \ne \text{bd}_m \tag{3.16}$$

$$t_{\text{en},i} \ge t_{\text{en},j} + h_{\text{safe}} \qquad \text{for} \quad o'_i \ge o'_j, \quad \text{bd}_i = \text{bd}_m \tag{3.17}$$

The operational requirements interests the control signal $\hat{a}_i$, by considering the fact that the automated vehicle should be physically able to execute the speed within the time step. The acceleration profile is therefore bound by a maximum and minimum acceleration. The speed of any vehicle is of course also bounded by the speed limit on the road. The operational constraints then are formulated as follows:

$$a^{\min} \le \bar{a}_i \le a^{\max} \qquad \text{for} \quad c_i = 1 \tag{3.18}$$

3.3.4. Upper layer Optimization

The upper layer optimization is a combinatorial optimization that aims to find the sequence of i-platoon with the least delay. Formally, the optimization problem is defined by the quadruple $(P, f(P), TD, \min)$. P is the set of i-platoons present at the intersection, $f(P)$ is the set of feasible crossing sequences. Given a feasible solution $O \in f(P(t))$, $TD = (P, O)$ is the total delay of the i-platoons following the sequence O and denotes the measure of the solution sequence O. Finally, min is the goal function, which is a minimization. The aim of the optimization is then to find for the i-platoons P an optimal solution, that is, a feasible solution O with:

$$TD\big(P, O) = \min\{TD\big(P, O' \mid S' \in f(P)\big)\} \tag{3.19}$$

where the $TD(P, O')$ equals the $TD(O')$ found in the lower lever optimization.

The **constraints** for the optimization problem relates to the assumptions made on the traffic behavior at the intersection, i.e. no overtaking. This constraint define what is considered a feasible crossing order by limiting the possible combinations of the sequencing. In order to be compliant to the constraint, i-platoons on the same stream should enter the intersection respecting the order they entered the detection zone of that stream

$$O' =< \dots, o_i, o_m, \dots >\in f(P) \quad \text{if} \quad d_i < d_m, \quad \text{bd}_i = \text{bd}_m \tag{3.20}$$

3.4. Solution of Upper Level Optimization

At the upper level, the optimization problem aims to find the sequence that yields the minimum delay. The algorithm applies the branching technique to systematically enumerate candidate solutions. The calculation of the cost (delay) of each candidate solution uses input from the lower level optimization. Branching alone would amount to brute-force enumeration of candidate solution and testing them all. To improve on the performance of the brute-force search, the algorithm keeps track of lower and upper bounds on the minimum that is trying to find and uses these bounds to reduce the search space, eliminating candidate solutions that it can prove will not contain an optimal solution. This technique is called branch and bound.

This section presents the algorithm used to solve the optimization at the upper level. The description is threefold: first the branching of candidate solutions is explained, secondly the cost of each candidate solution is formulated and thirdly the search algorithm is presented.

3.4.1. Branching candidate solutions

A crossing sequence O is defined as a set of i-platoon, ordered by their time of arrival at the intersection. Considering the fact that i-platoon are not allowed to overtake within the same lane (equation 3.3.4), i-platoon from the same bound direction should enter the intersection respecting the order they entered the detection zone. Any sequence O that satisfies this requirements is defined as candidate solution. Let's consider the subsets $P_b d$ that group the

i-platoon based on the bound direction they are driving from. In order to enumerate a candidate solution as stated above, the first i-platoon of any possible sequence is one of the first i-platoon of the subsets. Given the first i-platoon, the following i-platoon is either the second i-platoon of the same subset, or the first i-platoon of any of the other subsets. At each choice step, a partial sequence is formed and the step is repeated until all i-platoons have been selected once, hence all subsets are empty. If the process is done iteratively, exhausting all possible choice making, all feasible solutions are enumerated.

In order to compute this process, the set of candidate solutions is structured as a rooted tree and by exploring the branches of this tree, a systematic enumeration of all solutions is achieved. A rooted tree has a starting node (root) that is partitioned in children nodes. Each child node is then systematically partitioned in children nodes of its own. At the end of the branching, the nodes without any children are called leaf nodes. The paths connecting the root node to the leaf nodes constitute the candidate solutions. In this structure, each node represents an i-platoon assigned to a specific order, with the first children nodes having the first position and the leaf nodes having the last position in the possible sequences. The root node represents the last i-platoon of the previous optimal sequence. Starting from the root node, each node is partitioned into n branches: one branch indicates that the last i-platoon added to the partial sequence should be followed by an i-platoon from the same subset P_{bd_i}; other $n-1$ branches indicate that the partial sequence should be expanded with an i-platoon from other subsets where there are still i-platoons to be selected. The branching rule is repeated until it reaches a the leaf node. When a leaf nodes is reached, the branching is stopped and backtracking is initiated to the first node that was created but not branched out yet. The process ends when there are no more nodes to branch, only leaf nodes.

A concrete example of the branching logic is presented in Figure 3.7. The examples considers five i-platoon and two bound directions: $P_1 = \{1,3,4\}$, $P_2 = \{2,5\}$. The rooted tree is fully expanded and includes ten candidate solutions.

3.4.2. Cost of candidate solutions

Each candidate solution is associated with a cost. The solution with the least cost is the optimal solution. The cost is defined by the sum of individual weighted delay of the i-platoons forming the sequence. The individual delay is defined as the difference between the minimum exit time $t_{\mathrm{ex,min}}$ and the exit time the i-platoon has if it is scheduled after the preceding i-platoon. The minimum exit time is specific for each i-platoon because equals the standard minimum entry time plus the evacuation time a i-platoon needs to cross the intersection with the rear end of its last vehicle. The weighted delay is achieved by multiplying the delay with the number of vehicles that forms the i-platoon. The weighted aspect of the delay is introduced in order consider a fairness for the individual vehicles. In this paragraph, the words preceding and following will refer to the order of i-platoon in the sequence, not to be confused with the physically preceding or following i-platoon within the same lane.

The cost of each candidate solution O for N i-platoon is calculated at the leaf node

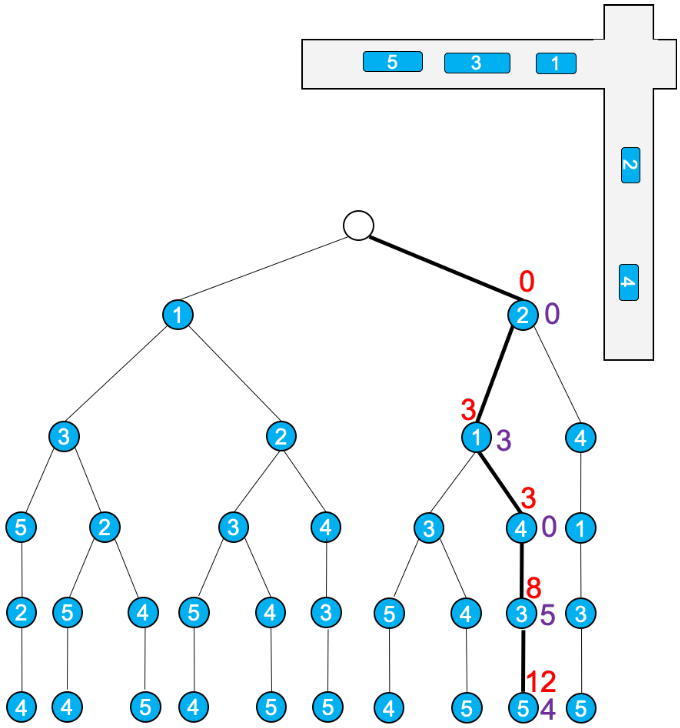


Figure 3.7: Example of branching candidate solutions

with the following equation:

$$\mathrm{cost}(O) = \mathrm{cost}(n_{\mathrm{leaf}}) = \sum_{i=1}^{N} (t_{i,\mathrm{ex,min}} - t_{i,\mathrm{ex}}) \cdot \mathrm{veh}_i \tag{3.21}$$

$$\text{where} \tag{3.22}$$

$$t_{i,\mathrm{ex,min}} = \frac{d_i(t) + w_{\mathrm{bd}_i}}{v_i(t)} + tl_i \tag{3.23}$$

The cost of a partial sequence O_1^n, (candidate solution at an intermediate state) can also be calculated at any node n, using the same formula and summing the individual delay of the i-platoon scheduled so far including the i-platoon of the node n.

Considering the candidate solution highlighted in the Figure 3.7, $O = (2,1,4,3,5)$, the fictional individual delay of each ordered i-platoon is reported with a purple number. The red numbers correspond to the cost of the nodes. The cost of the leaf node represented by the i-platoon 5. The cost of O equals the sum of all the purple numbers: $12 = 0+3+0+5+4$.

3.4.3. Search algorithm

In the algorithm described so far, all solutions must be enumerated completely in order to compute their costs and compare them. With the branch and bound algorithm, before enumerating the candidate sub-solution (partial sequence) of a node, the node's cost is checked against an upper and lower bound and it is discarded if it cannot produce a better solution than the best one found so far. With this approach, the feasible solution space is narrowed down and the search for the optimal solution can be faster. The efficiency of this algorithm lies on the definition of the upper and lower bound which define the criteria for a node to be discarded. In the worst case scenario, if the definitions are not efficient, the searching algorithm will result in the enumeration of all possible candidate solutions.

The upper UB and lower LB bound are inspired by the definition provided by Yan et al. (Yan et al., 2011). In this paper, the authors offer a mathematical proof of how any partial-sequence candidate that respects the defined lower and upper bounds are indeed part of the optimal complete solution. In their definition, the lower bound is composed by two elements, the cost of the partial-sequence $\mathrm{cost}(n)$ and sum of the minimum cost of the i-platoon not yet selected $\min \mathrm{cost}(P \cap n)$. The upper bound is the cost of the best complete solution found so far. At the beginning, the upper bound is set to infinite.

$$LB(n) = \mathrm{cost}(n) + \min \mathrm{cost}(P \cap n) \tag{3.24}$$

$$UB(n) = \mathrm{cost}(n_{\mathrm{leaf}}) \qquad \text{with} \quad UB(1) = \infty \tag{3.25}$$

The definitions of $\mathrm{cost}(n)$ and of $\mathrm{cost}(n_{\mathrm{leaf}})$ were given in the previous section. By considering $n-1$ the parent node of the child node n, in order to avoid repetitiveness in calculations, the cost of the node n can also be computed as follows:

$$cost(n) = cost(n-1) + (t_{n,\mathrm{ex,min}} - t_{n,\mathrm{ex}}) * \mathrm{veh}_n \tag{3.26}$$

This is easy to understand by looking at the Figure 3.7, where each node cost (red number) is equal to the sum of the previous node cost and the individual delay (purple number). For example, the node cost of i-platoon 4 is 3 = 3 + 0.

The sum of the minimum cost of the i-platoon not yet selected $\min \mathrm{cost}(N \cap n)$ is formulated as follows:

$$\min cost(N \cap n) = \sum_{i \in N \cap O_1^n} t_{n,\mathrm{ex}} - t_{i,\mathrm{en,min}} \tag{3.27}$$

The exit time $t_{i,\mathrm{ex}}$) is calculated for each branching that add a new i-platoon to the sub-sequence candidates. For controlled i-platoon this computation corresponds to the second layer optimization where the exit time is minimized (Section 3.5). For uncontrolled i-platoon the computation is an estimation of the i-platoon behavior approaching and crossing the intersection (Sections 3.6).

After the branching of a node, the lower bound of the branches is compared, and the branching is continued from the node which has the smaller lower bound. After a number of iterations, a leaf node is reached, the complete solution is named as *best so far* solution and the lower bound value of that node is the updated upper bound. Backtracking to the nodes with still partial solution, the upper bound is compared to their lower bound. If a lower bound is bigger than the upper bound, no branching is required from this node because there the final solution will always have a higher cost than the best so far solution. If the lower bounds is smaller than the upper bound, the branching is computed. During this iterative process, if a new complete solution is found with a lower bound smaller than the upper bound, than this solution is named the new best so far solution and its costs is the new upper bound.

The following pseudo-code in Algorithm 1 illustrates how the branching and bound technique is implemented to calculate the optimal sequence.

Algorithm 1 Optimal Sequence

Input: $\{p_{L1}, \ldots, p_{Li}, \ldots, p_{Ll}\}$(all subsets)
  Initialization:
  add root node r
  $LB(r) \leftarrow 0$
  $UP \leftarrow \infty$
  $Q = r$ ▷ queue of node to branch
  while $Q \neq \varnothing$ **do**
    if $G = (r, \varnothing)$ **then**
      $v \leftarrow r$
    end if
    Branch node v:
    $I = \{i | p_{Li} \neq \varnothing\}$
    for $i, i \in I$ **do**
      add edge $e = (v, m)$
      compute $LB(m)$
      if $LB(m) < UP$ **then** ▷ eliminate future node branching
        add m to Q
      end if
      $m \leftarrow m + 1$
    end for
    $v = \min\{LB(m), \ldots, LB(m+i)\}$ ▷ min lower bound node
    if v is leaf node **then**
      Backtrack $v = Q\{1\}$
      if $LB(v) < UP$ **then**
        $UP = LB(v)$ ▷ new upper bound
      end if
    end if
    delete v from Q
  end while
  $SEQ = n \mid LB(n) = UP$ ▷ leaf node with min cost
Output: SEQ (optimal sequence)

3.5. Solution of Lower Level Optimization

This section presents how the lower level optimization is solved. The lower level optimization aims to compute the optimal acceleration profile for controlled i-platoon given a candidate sequence. Given the order in which an i-platoon should enter the intersection, a controlled i-platoon should exit the intersection in the minimum time possible within a defined set of constraints. The exit time is considered as the sum of the entry time and the evacuation time, which is the time the i-platoon needs to cross the intersection. Thanks to the assumption made regarding the time length tl of an i-platoon, the exit time can be calculated as follows:

$$t_{\text{ex}} = t_{\text{en}} + \frac{w_{\text{dir}}}{v(t_{\text{en}})} + tl \tag{3.28}$$

where w_{dir} is the width of the intersection coming from the direction dir. The problem of generating a control sequence able to minimize the exit time can be formulated as a multi-objective optimization problem: among all possible sequences of control signals that control the vehicle to enter an intersection, find one such that arrival time is the smallest possible and the arrival speed is the highest. Considering the nature of the problem and the fact that this optimization has to be computed iteratively at each branching, a series of considerations are introduced in order to simplify the problem and its solution.

Let's consider a time-speed diagram, where $v(t_0)$ is the current speed of the vehicle, t_0 is the current time, d is the distance from the intersection, $a^{\max}$ is the maximum acceleration and $a^{\min}$ is the minimum deceleration, $v^{\lim}$ is the speed limit of the road. Any function $v(\cdot)$ in the time-speed diagram that satisfied the following constraints is a feasible speed schedule for reaching the intersection and the derivative of $v(\cdot)$ is an acceleration sequence for the acceleration-based controller.

$$0 \leq v(t) \leq v^{\lim} \tag{3.29}$$

$$a^{\min} \leq \frac{d}{dt} v(t) \leq a^{\max} \tag{3.30}$$

$$D = \int_{t_0}^{t_{\text{en}}} v(t) dt \tag{3.31}$$

Any feasible solution derived from this problem needs to respect also the constraints defined in Section 3.2.3, concerning the nature of the platoon and safety requirements. The equation 3.30 can than be re-written as follows:

$$a^{\min} \leq \frac{d}{dt} v(t) \leq 0 \tag{3.32}$$

$$t_{i,\text{en}} \geq t^{\text{safe}} = t_{m,\text{ex}} + t_{\text{buffer}} \quad \text{with} \quad (m, i) \text{ partial sequence} \tag{3.33}$$

The objective is to find a function $v(\cdot)$ such that v_{en} is as high as possible while t_{en} is as small as possible. Only piece-wise linear functions with slopes of the line segments equal to either $a^{\min}$ or 0 (eq. 3.32) should be used because for any non-piecewise linear function that satisfies the constraints, we can always find a piecewise linear function with a smaller t_{en} and/or a larger v_{en}. Considering the constrains 3.29 - 3.33, two cases can be identified. Case 1, the vehicle can reach the intersection by cruising to the entry with its current speed which correspond to the highest speed allowed (cond. 3.32). In this case, the acceleration profile equals to $< (a^{\max}, t_{\text{ex}} - t_0) >$. Case 2, the vehicle has to decelerate and then cruise at the lower speed to enter the intersection at a later time. The deceleration is equal to $a^{\min}$ and time of entry t_{en} equals the safe time t^{safe}. The acceleration profile is given by $< (a^{\min}, t_1 - t_0), (a^{\max}, t_{\text{ex}} - t_1) >$, where t_1 correspond to the time the vehicle stops decelerating and begins to cruise. In both cases, the piece-wise function allows for a numerical integration of the condition 3.31 and the computation of the solution can be generalized as follows:

$$\begin{cases} \frac{1}{2}(v(t_0) + v(t_1)) \cdot (t_1 - t_0) + v(t_0) \cdot (t^{\text{en}} - t_1) = D \\ v(t_1) = v(t_0) + a^{\min} \cdot (t_1 - t_0) \end{cases} \tag{3.34}$$

where, in Case 1 t_1 equals t_0 and $v(t_{\text{en}})$ equals $v(t_0)$. In Case 2, $v(t_{\text{en}})$ equals v^{t_1}. Figure 3.8 represents graphically the speed profiles of vehicles falling in Case 1 and Case 2, respectively. If the leader of an controlled i-platoon slows down to a very low speed, it has to continue its journey with the same speed, crawling its way out of the intersection. In the worse scenario, when it comes at a full stop it will never start to drive again as the future speed can only be equal or slower than the current speed. In order to prevent this, the trajectory control is computed for controlled i-platoon with a minimum speed of 5 km/h.

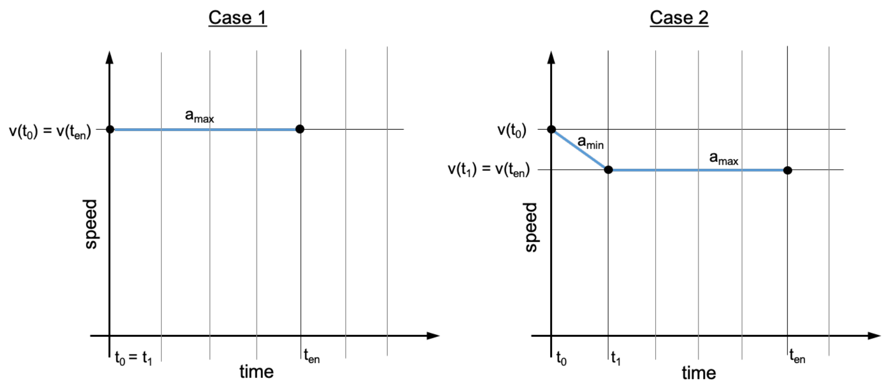


Figure 3.8: Speed piece-wise functions: Case 1 and Case 2

In summary, the algorithm for the trajectory control is shown in the following Algorithm 2.

Algorithm 2 Trajectory Control

Input: $v_i(t_0)$ (current speed), d_i (current distance to intersection), t_0 (current time), t^{safe} (minimum safe entry time)

Initialization:
if $c_i = 1 \&\& v_i \geq 5 km/h$ **then**
    $t = \frac{d}{v_i(t_0)}$
    if $t \geq t^{\text{safe}}$ **then** ▷ Case 1
        $t_{i,\text{en}} \leftarrow t$
        $t_1 = t_0$
        $v_i(t_{\text{en}}) \leftarrow v_i(t_0)$
    else ▷ Case 2
        $t_{i,\text{en}} \leftarrow t^{\text{safe}}$
        compute $v_i(t_1)$ and t_1
        $v_i(t_{\text{en}}) \leftarrow v_i(t_1)$
    end if
    compute $t_{i,\text{ex}}$
end if

Output: $< (a^{\min}, t_0 - t_1), (a^{\max}, t_{\text{ex}} - t_1) >$ (acceleration profile), t_{ex} (exit time)

3.6. Estimation of Uncontrolled *I*-platoons

Uncontrolled i-platoons drive through the intersection on their own accord. The time at which they exit the intersection cannot be controlled but can be predicted. In addition, the knowledge of their movement is essential in order to avoid collisions at the crossing area between conflicting streams with controlled i-platoons. The prediction of the exit time of uncontrolled i-platoons is divided in two steps: the arrival time prediction and the crossing prediction. The arrival time prediction refers to the moment the i-platoon reaches the start of the intersection and it is achieved by applying a modified car-following model. The crossing prediction determine when the i-platoon is able to drive through the crossing area, according to a gap acceptance model. The clear distinction is made due to the distinct nature of the models used to predict each component of the exit time.

3.6.1. Arrival Prediction

A trajectory prediction is introduced in order to estimate the arrival time of an uncontrolled platoon. The behavior of conventional vehicles at the intersection is assumed to follow the right-of-way rule for which vehicles will always give priority to the vehicles coming from their right. According to the layout of the intersection, the uncontrolled i-platoons can than be divided in two categories: i-platoons with priority and i-platoon without priority. I-platoons with priority have always the right to cross the intersection while i-platoon without priority can only cross the intersection if the crossing is free to do so without conflicts. In the first case, the i-platoon will approach the intersection unhindered, hence its motion is only bounded by its preceding i-platoon. In the second case, while approaching the intersection, the i-platoon has to engage in a deceleration strategy in order to stop at the crossing. For simplicity sake, it will be assumed that all vehicles perform a full stop and their arrival speed at the crossing is zero. The described behavior will be modelled by a car following model. Among the well-known car-following models, the Gipps model (Gipps, 1981) is chosen. In the Gipps model, vehicles are classified either as free or as constrained by the vehicle in front. When constrained by the vehicle in front, the follower tries to adjust its speed in order to obtain safe space headway to its leader. When free, the vehicle's speed is constrained by its desired speed and its maximum acceleration. The model is collision-free and doesn't require a calibration because it has no parameters. The simulation step equals the reaction time which make the numerical computation faster. The governing equation is as follows:

$$v_i(t+\tau) = \min\left(v_i^a(t+\tau), v_i^b(t+\tau)\right) \tag{3.35}$$

where τ is the reaction time which equals the simulation step, v_i^a is the maximum speed the vehicle can accelerate to during the time step, v_i^b is the maximum safe speed the vehicle can have in respect to the preceding vehicles.

$$\begin{cases} v_i^a(t+\tau) = v_i(t) + 2.5 \cdot a_i^{\max} \cdot \tau \cdot \left(1 - \frac{v_i(t)}{v_i^{\text{desired}}}\right) \cdot \sqrt{0.025 + \frac{v_i(t)}{v_i^{\text{desired}}}} \\ v_i^b(t+\tau) = a_i^{\min} \cdot \tau + \sqrt{(a_i^{\min} \cdot \tau)^2 - a_i^{\min} \cdot \left[2\big(s_m(t) - l_m - h - s_i(t)\big) - v_i(t) \cdot \tau - \frac{v_m^2(t)}{a_i^{\min}}\right]} \end{cases} \tag{3.36}$$

where $v(t)$ is the speed at time t, $s(t)$ is the position at time t, h is the net space headway at standing still, i is the follower i-platoon and m is the leader i-platoon. The model presented does not yet simulate the full stop of i-platoon without priority at the crossing. To achieve that, the stop line at the crossing is modeled as a standing still vehicle with zero speed and zero length. The algorithm 3 shows how the stop line vehicle is integrated in the Gipps model.

Algorithm 3 Trajectory Prediction 1 - Arrival

Input: (m, i) (partial sequence), $\hat{v_m}$ (speed profile of leader), t (current time step), s_{stop} (position of crossing), $a_i^{\min}, a_i^{\max}, h, v_i^{\text{desired}}, \tau$ (model parameters),

1: **Initialization:**
2: **if** $c_i = 0$ **then**
3: $\quad x_{\text{stop}} = \begin{cases} v_{\text{stop}} = 0 \\ s_{\text{stop}} = s_{\text{stop}} \\ l_{\text{stop}} = 0 \end{cases}$
4: $\quad$ **while** $s_i(t) - s_{\text{stop}} >= 0$ **do**
5: $\quad\quad$ **if** dir_i = non-priority **then**
6: $\quad\quad\quad$ **if** $\nexists\, x_m(t)$ or $s_m(t) - s_{\text{stop}} >= 0$ **then**
7: $\quad\quad\quad\quad x_m = x_{\text{stop}}$
8: $\quad\quad\quad$ **end if**
9: $\quad\quad$ **end if**
10: $\quad\quad$ compute $v_i(t+\tau)$
11: $\quad\quad$ compute $s(t+\tau)$ for i,m
12: $\quad\quad t = t + \tau$
13: $\quad$ **end while**
14: $\quad t = t_{i,\text{ar}}$
15: **end if**

Output: $t_{i,ar}$ (arrival time), $\hat{v_i}$ (speed profile of follower)

The algorithm is run until the i-platoon i has arrived at the crossing (line 3). If the i-platoon doesn't have an i-platoon leader in the detected zone or the leader has exited the detected zone during the simulation, the leader will be the stop-line vehicle. As an example, Figure 3.9 shows the trajectory predicted for two fictional i-platoons without priority. The i-platoon i has no leader. The green line represents the safe gap time within which the vehicle is expected to cross the intersection. The red line represents the time during which a conflicting vehicle is occupying the crossing.

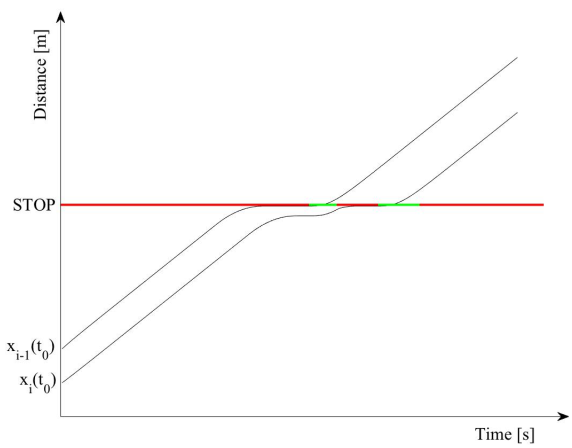


Figure 3.9: Example of the modified car-following model

3.6.2. Crossing Prediction

Once an uncontrolled i-platoon has reached the crossing area, if it doesn't have priority, it is standing still at the stop sign. At this point, the leader vehicle observes the gaps in the conflicting road and determines whether the gaps are adequate to cross the intersection. A*gap* is defined as the time headway between the stopped i-platoon and the leading i-platoon approaching from the conflicting stream. For a i-platoon i, the gap observed g_i is calculated as the difference between the estimated arrival times of the i-platoon i and the i-platoon m that is approaching from the conflicting road.

$$g_i = t_{m,\mathrm{ar}} - t_{i,\mathrm{ar}} \tag{3.37}$$

The evaluation of available gaps and the decision to carry out the maneuver defines the concept of gap acceptance. A gap acceptance model can help describe how a driver decide whether to accept it or not. In the gap acceptance model, the critical gap $\hat{g}$ is an important parameter and is defined as the minimum gap that a vehicle is willing to accept to enter and drive through the crossing area of the intersection.

For sake of simplicity, the gap acceptance model used in this research consists in a deterministic function: if the gap is lower than the a given critical gap, the gap is considered rejected thus the probability of acceptance is 0; if the gap is higher than the critical gap, the gap is accepted with 100% probability.

When the gap is accepted, the vehicle is expected to enter the crossing area at the time of the arrival. If the gap is rejected, the arrival time equals the exit time of the i-platoon which has just passed across. In this way, the updated arrival time can be used to evaluate the following gap, until a gap has been accepted.

Finally, the exit time can be computed. The equation used for calculated the exit time for controlled i-platoon (equation 3.31), is modified to consider that uncontrolled i-platoons are crossing the intersection from standstill:

$$t_{\mathrm{ex}} = t_{\mathrm{ar}} + \frac{w_L}{v(t_{\mathrm{en}})} + tl \tag{3.38}$$

The gap acceptance model is not only used to estimate the exit time of an uncontrolled i-platoon. It is also needed to estimate whether the partial sequence generated by the Algorithm 1 is realistic or not. Let's consider two i-platoons, i and m, approaching the crossing from the non-priority and priority roads respectively. The i-platoon i can either cross before or after the crossing of i-platoon m, depending on whether i accepts the gap g_i. Assuming it does accept it, the actual sequence is (i, m) and the sequence (m, i) will never happen. If at some point of the sequence optimization algorithm, the sequence (m, i) is considered as candidate solution, the arrival time of the i-platoon i is set to infinite so the solution will not be branched any further and it will never be chosen as optimal solution.

The pseudo-code summarizing the crossing prediction is presented in the Algorithm 4.

Algorithm 4 Trajectory Prediction 2 - Crossing

Input: (m, i) (partial sequence), $t_{i,\mathrm{ar}}, t_{m,\mathrm{en}}$ (arrival times), $t_{m,\mathrm{ex}}$ (exit time) , $\hat{g}$ (critical gap) , $a^{\max}$ (avg. acceleration from standstill)

```
Initialization:
if c_i = 0 && dir_i = non-priority then
    compute g_i
    if g_i < 0 then                                   ▷ m arrives first
        t_{i,ar} ← max(t_{m,ex}, t_{i,ar})
        ElseIf 0 ≤ g_j < ĝ                            ▷ i arrives first but does not accept the gap
        t_{i,ar} ← t_{m,en}
        ElseIf g_j > ĝ                                ▷ i arrives first and accepts the gap
        t_{i,ar} ← ∞
    end if
    compute t_{i,ex}
end if
```

Output: $t_{i,\mathrm{ex}}$ (exit time),

CHAPTER 4

Evaluation and Implementation Methodology

Chapter 3 covered the design of the controller, describing the control approach applied as well as providing its mathematical formulation. As described in the Research Approach (Section 1.4), the next step of this study is to evaluate the effectiveness of such design with respect to the research objectives. This chapter provides a description of the methodology used to plan and carry out this investigation. The evaluation plan introduces the Key Performance Indicators (KPI) and the scenarios selected to test the implementation of controller in different traffic environments. Following this description, the simulation platform is illustrated in detail specifying how the traffic environment and the controller are modelled. The results of the evaluation will be presented in the following chapter.

4.1. Evaluation Plan

The performance of the proposed traffic controller is tested within a simulation environment. The investigation aims to assess how well the controller is able to coordinate incoming traffic at a simulated intersection, under different circumstances. This analysis is then compared to the performance of the same intersection controlled by a standard traffic light controller with a fixed signal program. This section provides the description of the criteria chosen to measure the intersection performance and the description of the scenarios designed to measure the influence that external factors have on the intersection performance.

4.1.1. Evaluation Criteria

The research objectives regarding the intersection performance aim to investigate the efficiency of the controller as well as the qualitative driving comfort of drivers (Section 1.3, subquestion 5). Minimizing the travel delay is the objective of the controller, therefore it stands reason that travel delay can be used as indicator of the intersection performance. In addition, travel delay is one of the typical performance measures of intersection control, in literature as well as in practise, (J. Li et al., 2016), (Florin & Olariu, 2015). Qualitative driving comfort is not so straightforward to identify, as it can have different connotations and there is no standard definition. In relation to approaching an intersection, it can be stated that a common frustration of drivers is having to decelerate to a full stop and having to accelerate again to continue the journey. The discomfort increases even more so if this occurs more than once before being able to cross the intersection. For this reason, the number of stops per vehicle will be considered as KPI for qualitative driving comfort.

Travel delay

As mentioned, travel delay is one of the most used indicator to describe the performance of an intersection. Both total travel delay (veh.s) and vehicle delay (s) are commonly used. In this evaluation, vehicle delay will be considered as it allows to compare scenarios with different traffic volumes. The vehicle travel delay is defined as the difference between the average time needed to pass the road section and the time that a vehicle remains in the road section. Given the availability of individual vehicle records from the simulation, the average vehicle delay can be calculated as follows:

$$TD_{\text{mean}} = \frac{1}{N}\sum_{n=1}^{N} \max(0, TT_{\text{mean}} - TT_n) \tag{4.1}$$

where

$$TT_{\text{mean}} = \frac{\text{length road section}}{\text{speed limit}} \tag{4.2}$$

$$TT_n = t_{\text{exit road section}} - t_{\text{entry road section}} \tag{4.3}$$

In the equation 4.1, the enforcement of a positive delay is introduced so that vehicles that are driving faster than speed limit (negative delay) do not decrease the average delay. In order to analyze further the intersection performance and understand how it differently impacts vehicles, the average vehicle delay will be computed for the following relevant groups of vehicles:

1. $TD_{\text{mean,route}}$: average vehicle delay for vehicles travelling in the same route (origin-destination);

2. $TD_{\text{mean,cv}}$, $TD_{\text{mean,av}}$: average vehicle delay for conventional vehicles (cv) and automated vehicles (av);

In the first case, the comparison of average delay per route $TD_{\text{mean,route}}$ can be used as indicator of the fairness of the controller. In the latter case, investigating separately the delay of conventional vehicles and automated vehicles can give more insight on the effectiveness of the control strategy.

Number of stops

The number of stops is considered as proxy of the driving comfort. The number of stops corresponds to the number of times a vehicle has to come to full stop during its journey through the intersection area. This indicator is calculated per vehicle by the simulator software and it is retrieved directly from the output of the simulation runs. These values are then averaged to obtain the average number of stops per vehicle. As it is done for the travel delay, the average number of stops will be also defined for the aforementioned vehicles categories.

Additional traffic information

Aside of the performance indicators just described, additional traffic information will be considered in order to interpret better the results. In particular, the space mean speed is calculated. The space mean speed is defined as the average speed of vehicles traveling a given segment of road during a specified period of time. Different to the time mean speed which is an average of individual vehicle speeds, the space mean speeds weight the speed of slower vehicles more heavily, as the slower vehicles are within the segment of interest for a longer period of time. For this reason, it proves an interesting insight on the speed of vehicles travelling under the control of the V2I communication in opposition to vehicles reacting to a traffic light with fixed signal program. The space mean speed $V_{\text{mean,space}}$ is easily obtained from the output data of the simulation. Speed of vehicles are not reported at cross-section but at time instant. The space mean speed is therefore the mean of all the instantaneous speed of all the vehicles recorded. The segment of interest is defined between 190 m and -1 m distance from the crossing area.

$$V_{\text{mean,space}} = \frac{\sum_{n=1}^{N} \sum \text{instantaneous speed of vehicle n}}{\text{total number of observation}} \tag{4.4}$$

4.1.2. Evaluation Scenarios

In the simulation environment, the intersection system is influenced by four external factors. These external factors are traffic volume, demand ratio, penetration rate and the vehicle arrival pattern. Each factor has a different impact on the output of the control system, which is the performance of the intersection. In the next few paragraphs, an explanation of the relationship between the external factors and the system's performance is given. Subsequent of this explanation, the chosen evaluation scenarios are defined and visualized.

An increment of traffic volumes equals lower average headways. A lower average headways increases the probability of vehicles being grouped in the same i-platoons, thus the increase of volume doesn't necessary lead to an increase of i-platoon. Nevertheless, a higher number of i-platoon can be expected as the number of vehicles entering the intersection area increases. The concurrent presence of a high number of i-platoon at the detection area implies a multiplication of the total amount of possible sequence combinations causing a serious increase in computation time.

The second external factor is the demand ratio between the roads crossing at the intersection. When there is no traffic lights the capacity of the intersection is denoted by the priority rule. While the capacity of the road with priority is not affected as the journey of the vehicles is unhindered, the road without priority has a decrease in capacity. The capacity of the non-priority road is determined by the gaps in the priority flow, thus its volume. If the traffic volume reaches the decreased capacity, the intersection will become congested. When the congested condition sets in, the effectiveness of the controller is expect to reduce drastically.

The next external factor is the penetration rate, i.e. the percentage of automatic vehicles present in the network. A higher share of automatic vehicles obviously impacts the controller positively, since there is more controllable traffic that approaches the intersection. This factor is, especially in combination with traffic volume, crucial for a frequent formation of i-platoons in order to optimize the intersection's performance.

At last, the relationship between the arrival pattern and the performance of an intersection is described. The arrival pattern correspond to the distribution of vehicles' arrivals during the simulation. It affects the formation of i-platoons when vehicles are approaching the intersection with a short headway such that a i-platoon can be formed. In reality the arrival pattern is a random factor and as such, it cannot be predefined in the simulation, but it is randomly determined by the software. A traffic controller should be robust to the variation of arrival patterns, since is something it cannot be estimated.

The relationship between the performance of intersection and the external factors have now been clarified. It must be argued which and how external factors are used to form evaluation scenarios. To provide an answer to the first research question "Which is the range of penetration rate of automated vehicle and traffic volumes that is necessary to affect correctly the human driving?" the penetration rate and traffic volume are already required. Since it is too time consuming to construct sensible evaluation scenarios combining all three factors, it is chosen to focus on the penetration rate and the traffic volume. The considered range of traffic volume goes from 400 veh/h to 1600 veh/h of the total volume at the intersection. The full spectrum of penetration rates is tested, from 0% to 100% with increments of 20%. The demand ratio is set to 1:1 for all the scenarios. Even though the demand ratio is constant, by increasing the volume, the effects of reduced capacity will be experienced. The proposed controller will be referred as Control with Communication Only (CCO) and the comparing counterpart Fixed Traffic Light (FTL).

	Penetration Rate [%] - Control Type					
	0	20	40	60	80	100
Volume [veh/h]	CCO | FTL	CCO | FTL	CCO | FTL	CCO | FTL	CCO | FTL	CCO | FTL
400	1.1	1.2	1.3	1.4	1.5	1.6
800	2.1	2.2	2.3	2.4	2.5	2.6
1200	3.1	3.2	3.3	3.4	3.5	3.6
1600	4.1	4.2	4.3	4.4	4.5	4.6

Table 4.1: Evaluation Scenario

A quintuplet repetitions of each scenario is executed. Due to the characteristics of the simulation software, each run has a stochastic representation of the driving behavior, e.g desired speeds, arrival pattern. In this way, the performance of each scenario is accounted for the changes of driving behavior that reflects reality.

Each repetition is run for 15 minutes, which is considered a period long enough to capture the different dynamics of the traffic in response to the control signal.

Table below summarizes the scenarios that are considered in the evaluation plan. In total, 24 scenarios are be tested. These scenarios are tested first with the proposed controller and later with a standard fixed-time signal control.

4.2. Simulation Platform

The intersection environment described in Section 3.1 is reproduced using the simulation platform Vissim-COM. Vissim is a microscopic road traffic simulator based on individual behavior of the vehicles. Its mature simulation module made it the preferred platform for researches to simulate intersection traffic control scenarios (Jing et al., 2017). Vissim offers a user-friendly graphical interface (GUI) through which one can design the geometry of road networks and set up vehicles characteristics and vehicles driving behavior in a simple way, prior the start of the simulation. In order to simulate the communications sensors-to-controller and controller-to-actuators, an external application needs to able to access and manipulate the network objects during the simulation dynamically. For this end, Vissim offers an additional interface based on the COM (Component Object Model) which is a technology designed to enable inter-process communication between software. The COM Interface defines a hierarchical model in which the functions and parameters of the simulator, originally provided by the GUI, can be manipulated by programming. The traffic controller is modelled and run in the programming environment of MATLAB. Via the COM interface, the input of the controller is obtained by requesting vehicle information and the output of the controller is sent to the desired vehicles by editing appropriate attributes.

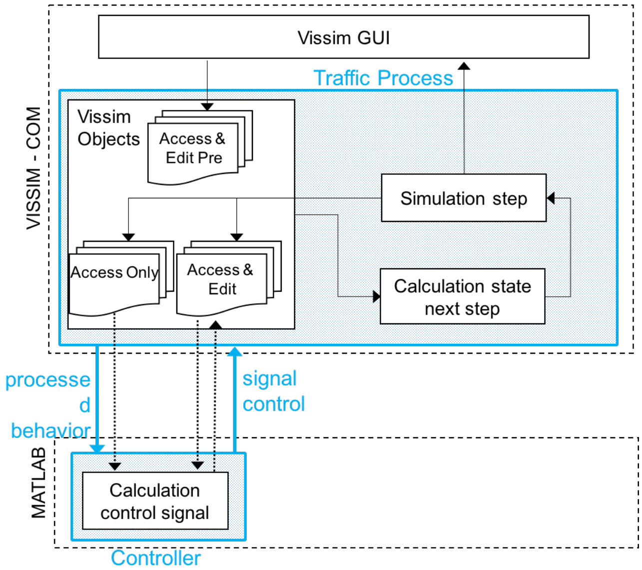


Figure 4.1: Representation of the Simulation Platform

Figure 4.1 depicts the analogy between the intersection environment and the simulation platform. The traffic system, is modelled in Vissim, the controller is modelled in MATLAB and the control process it simulated via the COM Interface which allows the dynamic communication between system and controller. The figure also includes a schematic overview of the time-step based simulation sequence in Vissim. From the start of the simulation, at every time step, the simulator calculates the state of all vehicles based on their attributes stored in a library. After the calculation, the simulator actuates the directive and simulate accordingly the objects in the network. The new state is stored, and a new calculation can be initiated. Within this cycle, the COM Interface gives access to the stored information at the end of each simulation, providing information on the processed (simulated) behavior. Information can also be edited at this point, allowing to determine and send the control signal, right before the calculations for the next step take place. As shown in the picture, not all the information available can be also edited during the simulation. This set up poses some restriction in what the user can manipulate, in particular regarding the ability to send the control signal to the automated vehicles. The issue is discussed and confronted in Section 4.4.2.

The described platform is required only for the simulation of the CCO Controller. Vissim offers the possibility to model traffic control with a fixed signal program, thus the simulation platform for the FTL controller is programmed internally in the Vissim software.

4.3. Vissim-COM Simulation

This section describes how the traffic system is modelled in the Vissim program. The description is threefold: vehicle behavior, infrastructure and traffic control. It covers all steps required to build a working simulation, namely the infrastructure and traffic behavior, the vehicle behavior, the traffic sensors and finally, the control signal.

4.3.1. Vehicle Behavior

This section focuses on the modelling of vehicle behavior, in particular aiming to show how the conventional and automated behavior has been implemented in the Vissim simulation. From the assumptions made in Section 3.1, the automated vehicles are expected to behave like conventional cars except for the following aspects:

1. Following behavior: the automated vehicles are not affected by the subjectivity and limitation of the human drivers, their perception of the environment is the result of their sensors' measurements.

2. Desired speed: the automated vehicles are expected to have always a desired speed equals to the speed limit. On the contrary, the conventional vehicles are expected to have a desired speed that fluctuates around the speed limit.

Driving Behavior

The traffic flow model in Vissim contains a psycho-physical car following model for longitudinal vehicle movement and a rule-based algorithm for lateral vehicle movement. The models deployed are based on Wiedemann's research work. Wiedemann's traffic flow model is based on the assumption that there are four different driving states, between which the drivers switch, depending on their current situation. For each of the four driving states, acceleration is described as a result of current speed, speed difference, distance to the preceding vehicle as well as of individual driver and vehicle characteristics. Drivers switch from one state to another as soon as they reach a certain threshold that can be described as a function of speed difference and distance. For instance, small differences in speed can only be perceived at short distances. Whereas large differences in speed already force drivers to react at large distances.

Vissim offers the possibility to change several parameters for following behavior, lane change and lateral behavior models. The road layout of this simulation consists of single-lanes, therefore the specification of lane change and lateral behavior is obsolete. On the other hand, the customization of the following behavior is important in defining the behavioral difference between conventional and automated vehicles. A detail description of the parameters available to the user is presented below. Following, Table 4.2 provides the chosen values of these parameters.

1. **Look ahead distance**: minimum and maximum distance that a vehicle can see forward in order to react to other vehicles either in front or to the side of it (within the same link). It is important when modeling the lateral behavior of vehicles.

2. **Look back distances**: minimum and maximum distance that a vehicle can see backwards in order to react to other vehicles behind (within the same link). The minimum look-back distance is important when modeling lateral vehicle behavior.

3. **Temporary lack of attention**: duration and probability. The duration is the period of time when vehicles may not react to a preceding vehicle. They do react however to emergency braking. The probability is the frequency of the lack of attention.

4. **Smooth closeup behavior**: If this option is checked, vehicles slow down more evenly when approaching a stationary obstacle. At the maximum look-ahead distance from the stationary obstacle, a following vehicle can plan to stop.

5. **Car following model**: Wiedemann 74 for urban environment, Wiedemann 99 for highway environment. For each car-following model, several parameters are available to influence the calculation of the vehicles' acceleration. For the Wiedemann 74, it is possible to define the parameters determining the desired distance d.

$$d = ax + (bx_{add} + bx_{mult} \cdot z) \cdot \sqrt{v},$$
$$\text{where} \quad v = \text{vehicle speed m/s}, \quad z = \text{value or range } 0.1$$

   (a) Standstill distance to static obstacles (ax): Defines the average desired distance between two cars. The tolerance lies from –1.0 m to +1.0 m which is normally distributed at around 0.0 m, with a standard deviation of 0.3 m.

   (b) Additive part of safety distance (bx_{add}): Value used for the computation of the desired safety distance d. Allows to adjust the time requirement values. Default 2.0

   (c) Multiplicative part of safety distance (bx_{mult}): Value used for the computation of the desired safety distance d. Allows to adjust the time requirement values. Greater value equals greater distribution (standard deviation) of safety distance

The values of the following behavior parameters for conventional vehicle are the default values proposed by Vissim. Both look ahead and look back distances are set to 250 m for automated vehicles, considering the technical capabilities of their front and rear radar sensors. Numerous publications have in fact associated automated vehicles with a 77Ghz long range radar with view range up to 250 m. Temporary lack of attention is set to zero both in duration and in frequency. According to the parameters related to the desired distance, in case of conventional drivers such distance ranges from 2 m during standstill to 20 m while driving at speed limit. In case of automated vehicles, the desired distance ranges from 1 2 m during standstill to 10 m while driving at speed limit. The parameters regarding the desired distance for automated vehicles are chosen following the suggestion given by PTV in a presentation regarding the simulation of automated vehicles in Vissim (PTV, 2017).

Desired Speed

During the simulation, if not hindered by other vehicles or network objects, e.g. signal controls, a driver travels at his desired speed. The desired speed varies per driver but it is possible

Parameter Name	Driving Behavior	
	Urban Conventional	Urban Automated
Look ahead distance [min max]	[0m 250m]	[0m 250m]
Look back distances [min max]	[0m 150m]	[0m 250m]
Temporary lack of attention [dur prob]	[1s 2%]	[0s 0%]
Smooth closeup behavior	Checked	Checked
Standstill distance to static obstacles (ax)	2 m	2 m
Additive part of safety distance (bx_{add})	2	2
Multiplicative part of safety distance (bx_{mult})	3	0

Table 4.2: Parameters of the Following Behavior Model

to influence the ranges of this variation. Vissim assigns the desired speed through a desired speed distribution, defined as fractiles. Fractiles are the cut-off points where the distribution reaches a certain probability. The function creates a probability distribution treating the elements as fractiles with equal probability spacing. It requires a minimum of two elements, one defining the zero fractile (f_{min}) and one defining the 1 fractile (f_{max}). If it has $n+1$ elements, the i-th value is in the i/n fractile. The distribution is piecewise uniform, that is it assumes a uniform distribution between each pair of adjacent fractiles. The standard distribution for the conventional vehicles is a function defined by <(0,48 km/h) (1, 58 km/h)>, meaning that the 100% of driver population has a desired speed between 48-58 km/h, with average of 53 km/h. Such distribution is defined by Vissim to simulate the driving behavior at roads with 50 km/h, reflecting the fact that some people usually drive slightly above the speed limit. Each vehicle is assigned with a random value x between 0 and 1 from which the desired speed is defined as follow:

$$\text{des\_speed}(x) = f_{min} + (f_{max} - f_{min}) \cdot x \tag{4.5}$$

The automated vehicles drives aim to drive at a speed equal to the speed limit. The desired speed distribution is a function whose minimum and maximum values equal to 50 km/h.

Desired Speed Distribution	Elements
± 50 km/h	<(0,48 km/h) (1, 58 km/h)>
Const. 50 km/h	<(0,50 km/h) (1, 50.01 km/h)>

Table 4.3: Configuration of desired speed distribution

Vehicle Category, Class and Type

In Vissim, there are three types of classification of vehicles: *vehicle category*, *vehicle class* and *vehicle types*. According to this classification, different attributes describing the vehicle behavior are assigned to either one of this three categories.

The vehicle category specifies the basic behavior in traffic and covers the differences between car, truck, public transport, bike or pedestrian. In the scope of this research, only the category *Car* is considered. Within this category, vehicles can be differentiated based on technical driving characteristics (e.g. different speed, acceleration) via vehicle types. Three vehicles types are defined: *Conventional*, *Automated Not connected* and *Automated Connected*. Automated vehicles enter the intersection as *Automated Not connected* and if they become leader of a controlled i-platoon their vehicle type is changed to *Automated Not connected*. The difference between connected and not connected type relates to the parameters of the acceleration functions assigned to these vehicles. The need for this distinction is explained in the description of the control signal and its simulation (Section 4.4.2). Vehicle classes provide the basis for speed data, evaluations, path selection behavior and other network objects. Most importantly, vehicle classes are used to assign vehicles to a specific driving behavior. Vehicles under the vehicle type *Conventional* are assigned to the class *Conventional* whereas vehicles assigned to either and two *Automated Not connected* and *Automated Connected* are assigned to the class *Automated*. Figures 4.2 give a comprehensive description of how this classification affects the implementation of conventional and automated behavior.

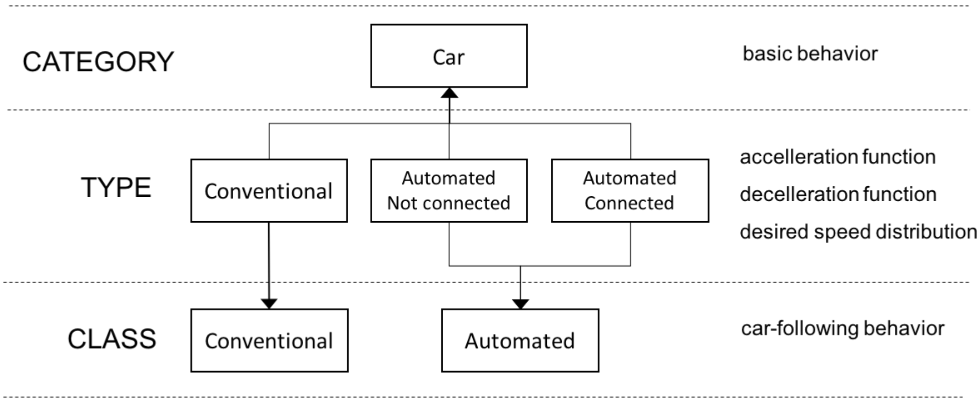


Figure 4.2: Summary of the vehicle classification and its attribute in Vissim

4.3.2. Infrastructure

This sections concerns the modelling of the infrastructure and the expected behavior of traffic driving in this infrastructure. Via the Vissim GUI, using the off-the-shelf objects, it is possible to create the road layout and the traffic rules. Consequently, the link behavior and traffic assignment can be defined.

Road Layout

The road infrastructure consists in the crossing of two single-lane roads, West-bound and South-Bound. No pedestrian or bike crossing are present. The two conflicting traffic flows are not regulated by a traffic light system and therefore drivers cross the intersection according to the right-of-way rule. In this case, it means that drivers on the south-bound always have priority to cross with respect to the drivers approaching from the west-bound road. The south-bound road is refereed as *priority road* and the west-bound road as *yielding road*. Each road is modelled by a link. At this point, the modelled roads represent two links over-crossing each other. Vehicles driving on different link have no interaction with each other. In order to made them aware of the crossing area, traffic rules needs to be introduced.

Traffic Rules

The yielding behavior in a non-signalized intersection can be modelled by using either priority rules or conflict areas. Depending on the choice made, the driving behavior of the drivers on both the yielding and priority roads differs.

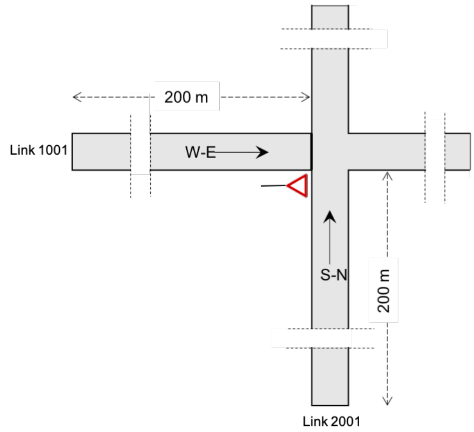
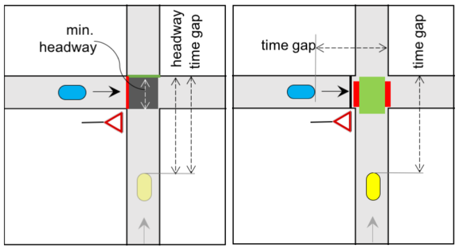


Figure 4.3: Representation of the Simulated Intersection



(a) Priority Rule *(b)* Conflict Area

Figure 4.4: Modelling Yielding Behavior

With priority rule, the vehicles on priority road drive unhindered towards and through the crossing area. When vehicles on the yielding road travel to the stop line, the crossing decision is taken considering two criteria: the gap time and the (space) headway. The gap time is the time that the first upstream vehicle in the priority road will require to reach the end of the crossing area in its present speed. The headway states the distance between the end of the crossing area and the first vehicle which is moving towards the crossing area. Depending on the traffic situation, either the headway or the time gap is more important. In a normal traffic flow, it is mainly the time gap which is relevant. In the case of slow-moving traffic and congestion, the gap time is often large enough to pass the prerequisites and so the headway becomes the decision factor. In summary, under priority rules, a yielding vehicle does not need to stop and wait at the stop line if all conditions of gap time and headway are fulfilled. If either one of them is not met, the vehicle waits at the stop line until both gaps are sufficiently long. In this case the priority rule is treated as a preceding vehicle with null speed. In the Vissim documentation (PTV, 2013), it is not specified when and how often the criteria for the yielding vehicle are checked.

In conflict areas, a vehicle in a yielding road will calculate whether it will be able to cross the priority road with every time step while approaching the conflict area. If the driver feels there is a large enough gap in the main traffic stream, he will simply continue to drive. If the gap is too small, the vehicle will decelerate as if it had to stop in front of the conflict area. This calculation is repeated with the next time step. The braking is then either cancelled or the driver continues driving and might even accelerate when finding a gap in the traffic stream to enter safely. The acceleration allows him to pass and clear as soon as possible the conflict area. A vehicle in the minor stream will not enter a conflict area if it assumes that it will not be able to leave it before the next vehicle of the main stream arrives. Vehicles in the main traffic stream also react to conflict areas. If a vehicle does not manage to cross the entire conflict area because the driver has misjudged the situation, the vehicle in the main traffic stream will brake or even stop. The drivers that have the right of way carry out a comparable decision-making process for crossing the conflict area as the drivers who are supposed to yield.

The fact that vehicles in both direction are aware of each other, makes the conflict area a much more suitable choice for the simulation regarding the movement of automated vehicles in the yielding road. In this way, it can be simulated how conventional vehicles on the priority road would react to the crossing of automated vehicles. While the timing is technically safe, their behavior can be perceived otherwise by the human driver used to a more precautions behavior.

Link Behavior

The following step is to define and assign a link behavior to the road links, based on the environment they represent, e.g. whether a link represents a highway road or an urban road. By defining a link behavior is possible to distinguish different driving behaviors by vehicle classes. For this simulation, the link behavior *Urban Future* is created by associating the conventional and automated classes to the their respective driving behaviors. This link behavior is the assigned to the links 1001 and 2001.

Link Behavior	Vehicle Class	Driving Behavior
Urban Future	Conventional	Urban Conventional
	Automated	Urban Automated

Table 4.4: Configuration of link behavior

Link	Link Behavior
1001	Urban Future
2001	Urban Future

Table 4.5: Assignment of link behavior

Traffic Assignment

Traffic assignment consists in appointing the origin and destination links within the network and define the available paths linking origin and destination. Given the simplicity of the intersection, this step is quite straightforward. There are two origins, also referred as vehicle inputs, positioned at the beginning of the links 1001 and 2001, respectively. From each origin, there is only one path available leading to end of the link, which correspond to the destination. The routing decision is therefore static.

Vehicle Input	Link	Volume	Vehicle Composition
1	1001	*200*	*Mixed*
2	2001	*200*	*Mixed*

Table 4.6: Configuration of vehicle input

The traffic volume entering the network is defined per origins. The attributes of each vehicle input consist in volume (km/h) and vehicle composition. The vehicle composition is the element where the penetration rate of automated vehicles can be introduced. The vehicle composition defines the composition of the traffic per vehicle type. It also assigns a desired speed distribution to each vehicle type. Once a vehicle type is assigned to vehicle composition, its minimum relative flow is 0.001 and can't be set to 0. Therefore, to simulate all required scenarios, three vehicle composition are created: one with only human drivers, one with mixed penetration and one with only automated vehicles. The *Mixed* vehicle composition is used for the scenarios of 20%, 40%, 60% and 80% of penetration rate. For this reason, in Table 4.7, the values written in *italic* are values that change according to the evaluation scenario. The desired speed distribution described in Section 4.3.1 is here assigned to the vehicle types. The vehicle type representing the automated vehicles is the Automated Not Connected, as this is the mode in which the vehicles enter the network.

Vehicle Composition	Vehicle Type	Desired Speed Dist.	Relative Flow
Conventional	Conventional	± 50 km/h	1
Mixed	Conventional	± 50 km/h	*0.2*
	Automated Not connected	Fixed 50 km/h	*0.8*
Automated	Automated Not conected	Fixed 50km/h	1

Table 4.7: Configuration of vehicle computation

4.3.3. Traffic Control

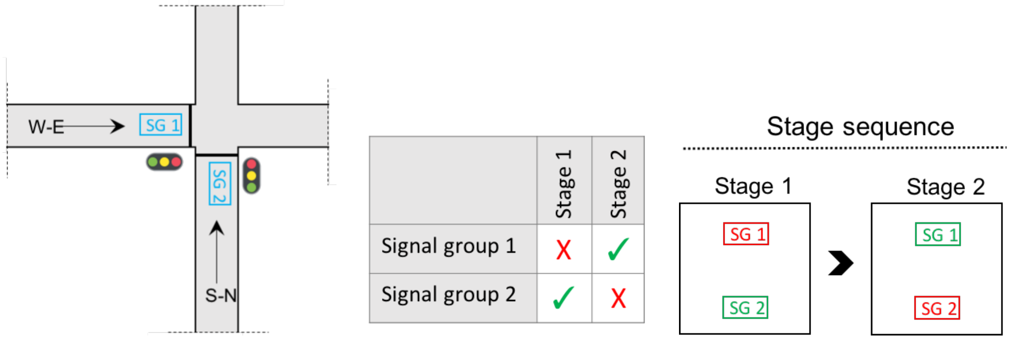
The evaluation plan entails the simulation of the (Control with Communication Only) CCO Controller as well as the simulation of a standard traffic control system with a fixed signal program. While the CCO Controller cannot be programmed internally in Vissim, the software does offer the ability to control an intersection via a fixed traffic light program. This type of control set off-line the greet time for each direction and the order in which it is distributed, according to historical traffic information.

Fixed Traffic Light Control

In order to introduce the signal control, traffic lights should be first added to the infrastructure via the network objects *signal head* (Figure 4.5). Each traffic light is assigned to a signal group. The signal control program needs to define the following elements:

- **Stage**: A phase defines which signal group receive green time at the same time and by default which ones received red time.
- **Stage sequence**: The sequence of the phases at each cycle
- **Signal sequence**: The sequence of the signal for each phase, e.g. red - green - amber.
- **Cycle time**: Duration of one cycle
- **Signal duration**: Duration of each signal per phase.

The stages and their sequence are depicted in Figure 4.5. The configuration is straight forward: once the W-E direction has green, the conflicting direction has red and vice-versa. The signal sequence is made of red-green-amber. The amber time lasts always 3 seconds, during



(a) FTL Infrastructure *(b)* FTL Stages *(c)* FTL Stage sequence

Figure 4.5: Modelling Fixed Traffic Control

which vehicles are still allowed to pass through. In all scenarios, the volume ratio between the two direction is 1:1. Thus, the amount of green time should be distributed equally among the signal groups. This leads to assigning 27 seconds to green time and 30 seconds of red time. The resulting signal program is synthesized in the following Figure 4.6.

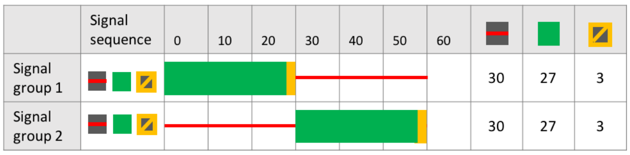


Figure 4.6: Representation of the Simulated Intersection

4.4. COM-Matlab Simulation

The COM Interface allows to initiate the simulation of Vissim from a Matlab script and automate the run of multiple simulation. The pseudo code in Algorithm 5 shows how the simulation platform is implemented in practise. After setting up the simulation settings to match the evaluation scenario, the simulation is initiated as a for loop. At each loop, Vissim simulate a single step equal to 0.1 seconds and receives the signal control. Every 10 simulation steps (1 second), the sensors' measurements are collected and a new signal control is calculated.

The highlighted modules define the three main components that allows the implementation of the controller within the simulation. Chapter 3 already provided an extensive description of how the optimal control signal is calculated. The following sections will focus on the description of the input sensor and the communication of the control signal.

4.4.1. Input Sensor

The intersection is equipped with sensors able to provide accurate information about individual vehicles. The sensors consist in traffic radars and in the automated vehicles themselves. The traffic radar technology allows to track vehicles within a range of sight and provides position, speed, acceleration, length and id of each vehicle observed. The same type of information is available from the automated vehicles about their own vehicles. Whereas the radar is supposed to provide information over all traffic, the automated vehicles are sup-

Algorithm 5 Main - Simulation

```
Initiation of Server:
Connect to the COM Server
Load Vissim file

Simulation:
for volume = 200:200:800 do
    for penetration_rate = 0:0.2:1 do
        for seed = randi([1 100],1,5) do
            set random seeds
            set volume
            set penetration rate
            for s=1:End_of_simulation do
                Run single step s                ▷ Simulation step in Vissim
                t=s/10
                if mod(t,1) == 0 then
                    Input Sensor
                    Calculate Signal Control
                end if
                Send Signal Control
            end for
        end for
    end for
end for
```

posed to identify themselves as 'automated' and send the information to allow for a mapping of the automated vehicles in the traffic radar detection. Eventually, the information that the controller is supposed to receive is a list of all vehicles with their proprieties.

In general, Vissim produces by default a set of sensing data, defined as attributes of each vehicle. The vehicle data is stored in a database, which is updated after every simulation step. This database is accessible via the COM Interface by accessing the object *Vissim.Net.Vehicles* and collect the required attributes. The list of attributes collected in the simulation are showed in Table 4.8. The output frequency of such information is 1 s and no measurement error is taken into account.

Radar Measurement	Vehicle Attribute	[Short Name]
Vehicle Id	Vehicle Number	[No]
Distance to Stop Line	Distance Travelled	[DisTravTot]
Vehicle Length	Vehicle Length	[Length]
Vehicle Speed	Vehicle Speed	[Speed]
Vehicle Type	Vehicle Type	[VehTyNo]
Radar Id	Lane	[Lane]

Table 4.8: Specification of the Input Information received by the CCO Controller

4.4.2. Control signal - Communication

The control signal consist in a parameterized acceleration profile for leaders of controlled i-platoons. The acceleration profile is a uniform piecewise function defined by two segments. In the first segment the vehicle decelerate with -3 m/s^2 for $t_1 - t_0$ time, in the second segment the vehicle cruise with constant speed (0 m/s^2) until its i-platoon exits the crossing area at t_{ex}.

Once the vehicles receive the signal at t_0, they should execute their movement accordingly between t_0 and $t_0 + 1$ seconds. Ideally, Vissim should use the acceleration provided by the controller to calculate the movement of the vehicles. However, the vehicles' acceleration is an attribute internally calculated by the software and cannot be manipulated by the user. Moreover, as depicted in Figure 4.3 (Section 4.2), the ability to access and edit objects doesn't occur between the calculation step and the simulation step but between the simulation step and the calculation for the next step. This means that is only possible to alter the input for the calculations and not the results of the calculations themselves. In order to be able to implement the control signal an indirect solution is needed.

One of the attributes available for editing during the simulation is the speed of the vehicles. Let's consider the following situation. At the end of the simulation step s, the vehicle i has a speed $v_i(s)$. $\hat{a}_i(s)$ is the acceleration that vehicle i should use according to the control signal. $a_i(s)$ is the acceleration calculated by the software. Assuming that the software uses the kinematic equation to calculate the next movement, the new speed of vehicle i at simulation step $s+1$ for the software $(\cdot)$ and for the controller $(\hat{\cdot})$ are the following:

$$v_i(s+1) = v_i(s) + a_i(s) \cdot ds$$
$$\hat{v}_i(s+1) = v_i(s) + \hat{a}_i(s) \cdot ds$$

If speed $v_i(s)$ is set equal to $\hat{v}_i(s+1)$ and $a_i(s)$ is forced to 0 m/s^2, $v_i(s+1)$ equals the desired $\hat{v}_i(s+1)$. The acceleration of vehicles can be forced to be zero by manipulating the maximum and desired functions of acceleration and deceleration. To account for differences in the driving behavior of several drivers and different vehicle properties during acceleration and deceleration, Vissim uses functions instead of individual acceleration or deceleration data. Maximum acceleration and deceleration functions outline the limits of what the vehicle is technically capable of executing. The desired acceleration and deceleration functions are used for all situations in which the maximum functions are not required. According to the interaction state of the vehicle, the car-following model decides to either decelerate or accelerate. If all functions related to the acceleration and deceleration are set to 0 m/s^2, no matter the state in which the vehicle is, it will always keep its current speed constant. This is the reason why there are two vehicles types representing the automated vehicles (Section 4.3.1). The vehicle type Automated Non Connected has the same acceleration and deceleration functions of the Conventional type. The vehicle type Automated Connected has instead all functions equal to the constant value 0 m/s^2. Ultimately, instead of an acceleration profile, a speed profile is used as control signal and every 0.1 seconds, the control speed is sent to the vehicles.

This alternate solution comes with two disadvantages. Firstly, since the vehicles cannot break, the execution of the signal as speed does not guarantee that the vehicles will keep a safety distance to the preceding vehicles. Secondly, while the speed $v_i(s+1)$ equals $\hat{v}_i(s+1)$, the space movement will differ. The latter case does not significantly impact the execution of the control signal nor affects the reliability of the simulation. At each simulation step (0.1

seconds), the difference between the actual simulated movement and the movement the controller expects is on the range of centimeters.

The first inconvenience, regarding the lack of assurance on safety distance, is not acceptable because it can lead to the absurd situation where vehicles overrun each other in the simulation. In order to solve the issue, a safety speed needs to be calculated that defines the maximum speed the vehicle can drive while respecting a safe distance to the preceding vehicles. In order to relieve the Matlab computations to a minimum, the safety speed is introduced in Vissim as user-defined attribute of vehicles, so that its value will be computed at each simulation step by the software. User-defined attributes are created using either Vissim data or a formula as data source. In case of formula, the equation can be defined using attributes values of any network object as input. The limitation lies only on the available attributes Vissim provides, e.g. is not possible to known the speed of a preceding vehicle but it is possible to know the speed difference with the preceding vehicle.

The formula of the safety limit speed is based upon the IDM (Intelligent Driving Model) () which is used to simulate the automated driving behavior. The desired minimum gap between two vehicles under IDM, is given by the following expression:

$$\text{hdwy}_{i,\text{safe}}(s+1) = \text{hdwy}_0 + v_i(s \cdot T) + \frac{v_i(s) * \delta v_i(s)}{\sqrt{a \cdot b}} \tag{4.6}$$

where T denotes the safety time gap, $\delta v(s) = v_i(s) - v_{i-1}(s)$, hdwy_0 is the minimum distance for congested traffic, a the maximum acceleration and b the comfortable deceleration. Applying the kinematic equations to the equation 4.6, the safety distance is formulated as function of the $a_{i,\text{safe}}(s)$.

The Algorithm 6 provides the pseudo-code regarding the communication of the control signal.

Algorithm 6 Send Control Signal

Input: Signal_Id (Id of the leader of controlled i-platoons), Signal_T1 (Time until which the vehicles have to break),Signal_Tex (Time of exit of controlled i-platoons)

```
for vehicle ∈ Signal_Id do
    if t ≤ Signal_T1(vehicle) then                          ▷ Braking phase
        Speed_new = min{Speed safe, Speed − 3 ∗ dt}
        set Speed_new to vehicle
        set Type Automated Connected
    end if
    if Signal_T1(vehicle) < t ≤ Signal_Tex(vehicle) then    ▷ Cruising phase
        Speed_new = min{Speed safe, Speed}
        set Speed_new to vehicle
        set Type Automated Connected
    end if
    if t > Signal_Tex(vehicle) then                         ▷ End of control
        set Type Automated Not Connected
    end if
end for
```

CHAPTER 5

Results

The previous chapter introduced the simulation environment used to evaluate the performance of the controller. Following the different evaluation scenarios, the results are here illustrated and analyzed. As a consequence of the final discussion over the overall results, the research questions will be answered in the following Chapter.

5.1. Results Overview

The results of the intersection performance are summarized in Tables 5.1, 5.2, 5.3. In total, 48 different scenarios have been tested and evaluated according to the criteria described in the previous chapter. The results shown in the tables are the average values of the multiple runs executed per each scenario. In almost all scenarios, the proposed controller CCO (Control with Communication Only) has achieved positive results, besting the performance of its Fixed Traffic Light (FTL) counterpart. In the cases where the CCO performs better, the difference in performance is substantial, ranging from 10 to 2 times higher in average vehicle delay. The number of stops reflects the same behavior. The contrast might be due to the fact that a fixed signal program uses a blunt logic to regulates traffic that lacks the ability to adapt the signal to the vehicle arrival. In fact, the space mean speed for the FTL stands around 30 km/h, showing that no matter the traffic demand, vehicles will be subjected to the same delay. On the other hand, the space mean speed for the CCO shows how the control signal slows down vehicles only when it is necessary (i.e. avoid collision). The scenarios where the CCO performs worse than the FTL are 4.1 and 4.2, where the total traffic volume is 1600 veh/h and the penetration rate is null and 20% respectively. During the running simulations, it can be seen that in these scenarios, the queue forming on the minor road start to spill back considerably before it starts to resolve. The scenario with 1600 veh/h is in fact, the only scenario where traffic regulates by the FTL reaches saturated condition, i.e. the queue is not resolved within the green time.

As expected, the performance of the fixed traffic light slightly worsens with the increase of traffic volume. A higher input causes longer queues and higher waiting time. This is supported by the results of the space mean speed, which decreases significantly in the scenario with the highest volume. The number of stops remains stable throughout the different scenarios. On average, one in two vehicles has to come to a full stop during its journey. The increase in penetration rate has a small impact, possibility correlated to the fact that automated vehicles travel with the same desired speed of 50 km/h.

The performance of CCO is obviously more affected by the penetration rate as well as by the traffic volume increase. The behaviour of the performance will be more in depth analyzed in the following sections of this chapter.

Average Vehicle Delay [s]

Volume [veh/h]	0 CCO	0 FTL	20 CCO	20 FTL	40 CCO	40 FTL	60 CCO	60 FTL	80 CCO	80 FTL	100 CCO	100 FTL
400	0.31	9.84	0.28	9.08	0.26	8.89	0.21	8.32	0.11	8.03	0.05	7.52
800	0.73	11.43	0.74	11.04	0.69	10.45	0.66	10.11	0.45	10.00	0.07	9.96
1200	4.62	12.13	2.70	11.84	1.5	11.65	1.70	11.11	0.87	10.77	0.31	10.80
1600	26.61	17.72	23.72	14.62	18.80	13.70	2.33	12.06	0.35	10.91	0.70	10.78

Penetration Rate [%] - Control Type

Table 5.1: Results Overview - Delays. Rounding at the second decimal

Average Number of Stops per Vehicle

Volume [veh/h]	0 CCO	0 FTL	20 CCO	20 FTL	40 CCO	40 FTL	60 CCO	60 FTL	80 CCO	80 FTL	100 CCO	100 FTL
400	0.01	0.44	0	0.43	0	0.44	0	0.44	0	0.43	0	0.44
800	0.08	0.51	0.06	0.51	0.01	0.50	0	0.50	0	0.49	0	0.49
1200	0.76	0.56	0.24	0.53	0.06	0.53	0.01	0.52	0	0.49	0	0.51
1600	3.91	0.61	3.23	0.58	2.60	0.58	0	0.57	0	0.56	0	0.56

Penetration Rate [%] - Control Type

Table 5.2: Results Overview - Stops. Rounding at the second decimal

Average Space Speed [km/h]

Volume [veh/h]	0 CCO	0 FTL	20 CCO	20 FTL	40 CCO	40 FTL	60 CCO	60 FTL	80 CCO	80 FTL	100 CCO	100 FTL
400	51.01	31.75	50.63	30.85	50.48	30.86	50.40	30.34	50.46	30.54	50.14	30.20
800	44.29	27.59	50.03	28.29	49.81	28.83	49.51	29.25	49.35	29.59	49.86	30.12
1200	42.23	27.06	44.46	27.75	46.32	27.15	47.31	27.98	48.05	30.54	49.27	28.61
1600	19.32	23.15	21.67	25.15	23.48	27.27	48.40	27.97	49.46	28.62	49.14	28.96

Penetration Rate [%] - Control Type

Table 5.3: Results Overview - Space Speed. Rounding at the second decimal

In correlation to this analysis, the total number of i-platoons formed during the simulation are considered. The results are shown in Figure 5.1. The number of i-platoons are calculated by counting the number of vehicles that have led an i-platoon at any time during their journey through the detection area. This means that this result reflects the time when i-platoons split, but it does not account for the time when i-platoons join each other because the new joined i-platoon does not have a new leader. A split can happen when a human driver drifts from its preceding vehicle, when two i-platoons join each other but their cumulative length is longer than the limit allowed or more likely, when an vehicles of an i-platoon come to full stop.

In the figure, the data is distinguished by controlled and uncontrolled i-platoons. As expected, the number of controlled i-platoons (blue bars) grows with the increase of penetration rate and it does so almost at a linear peace. In respect of the volume, the number of controlled i-platoon follows almost proportionally its increase. It seems that the increase of average headway didn't really impacted the aggregation of vehicles in i-platoon. For example, when the penetration rate is null, the number of uncontrolled i-platoon is almost equal to the

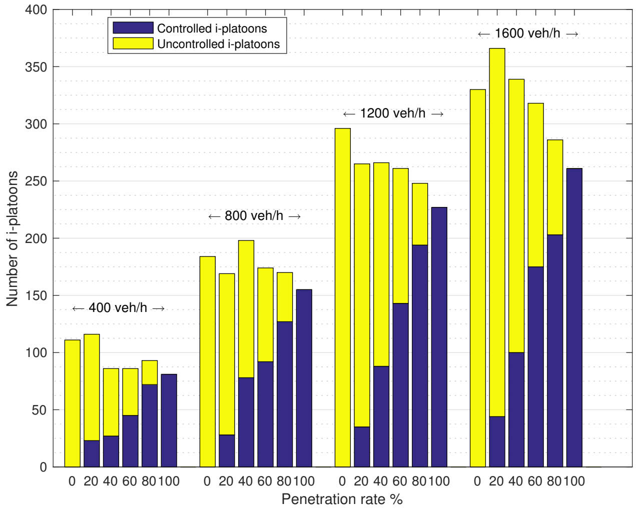


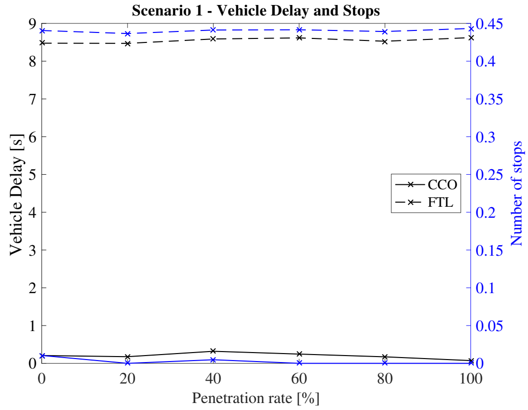
Figure 5.1: Representation of the simulated i-platoons

number of vehicles for all different volume scenarios. As previously explained, this outcome could also be the result of uncontrolled i-platoon splitting when they stand in a queue. It is challenging to relate this data directly to the performance of the intersection, as multiple factors concur in the formation of the i-platoons. Nevertheless, this results will be used to support the answer to one of the sub-research question, investigating the ability to influence the conventional vehicles.

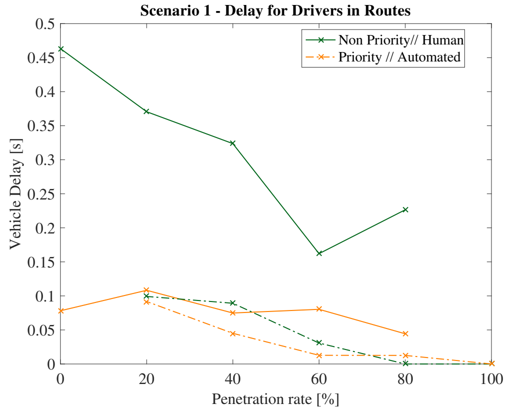
The following section, the results are grouped by the traffic volume, allowing to focus the analysis on the impact of the penetration rate on the intersection performance.

5.2. Scenario 1 - Low Traffic

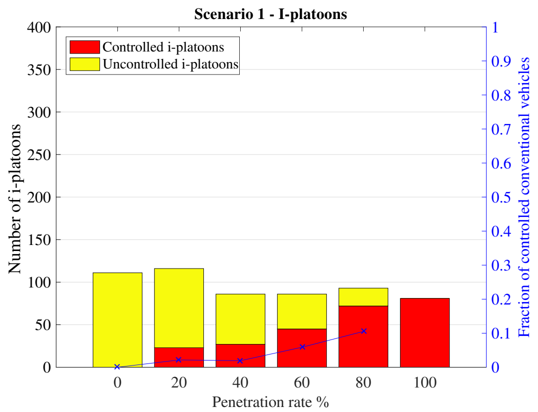
Scenario 1 is characterized by a total volume of 400 veh/h, 200 veh/h each route. During the simulation, approximately 100 vehicles have been simulated. As expected, the average delay of all vehicles in the CCO simulation is lower than the delay caused by the fixed traffic light control (Figure 5.2a). The difference in delay is significant for all penetration rates, as traffic coordinated by the CCO Controller has essentially no delay. The same can be said for the average number of stops per vehicle. The fact that full conventional traffic already experiences no delay shows how vehicles on both approaching directions drive on free flow and barely interact with one-another. The number of induced i-platoons is ranges between 85 and 112, indicating that the formed i-platoons are mostly constituted by a single vehicle. The potential improvement of the CCO Controller cannot be really tested in this situation. However, the results prove that the trajectory control is successful in coordinating the i-platoons across the intersection.



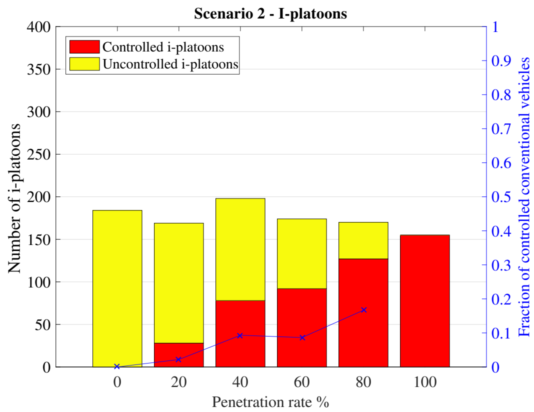
(a) Delay and Stops in Scenario 1



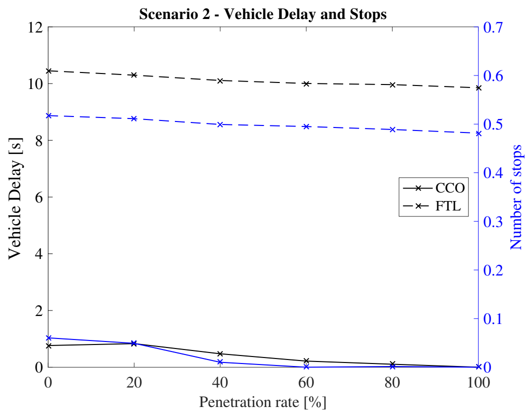
(b) Delay for Drivers in Routes in Scenario 1



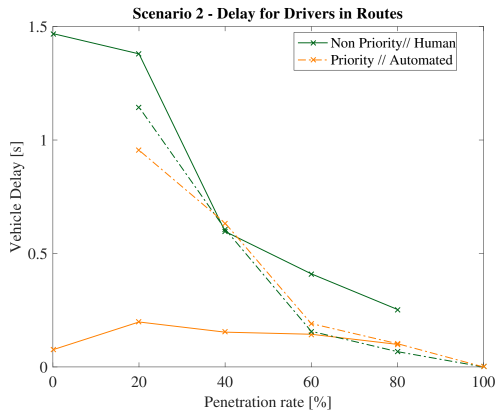
(c) Number of *i*-platoons in Scenario 1



(d) Number of *i*-platoons in Scenario 2



(e) Delay and Stops in Scenario 2



(f) Delay for Drivers in Routes in Scenario 2

Figure 5.2: Results for Scenario 1 and Scenario 2

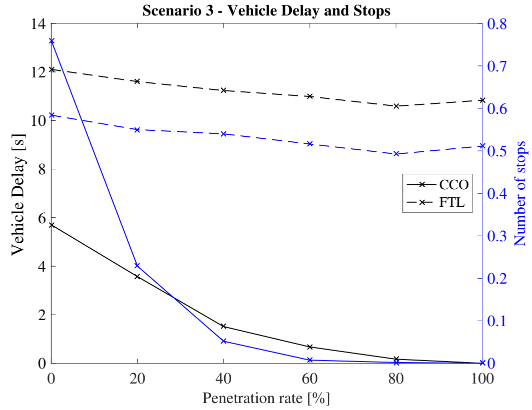
5.3. Scenario 2 - Medium-Low Traffic

Scenario 2 is characterized by a total volume of 800 veh/h, 400 veh/h per each route. During the simulation, approximately 200 vehicles are simulated. Scenario 2 is in some aspects similar to Scenario 1 because the flow is still not high enough to cause a significant delay with self-regulating traffic. As expected, the average vehicle delay with the CCO simulation keeps being lower than the delay caused by the fixed traffic light control(Figure 5.2e). However, in this scenario, the presence of automated traffic starts to have a bigger impact on the delay caused by the CCO Controller. From 40% penetration rate, the increase of automated vehicles follows a linear decrease of the delay. The stop data confirms that, in this scenario, the application of CCO Controller is most effective with at least 40% where delays are minimal and vehicles avoid stopping altogether.

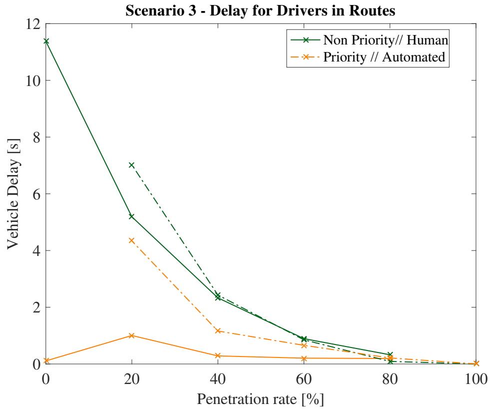
Given the relevance of the CCO Controller's impact, Figure 5.2f further investigate the average vehicle delay of 4 different vehicle groups: conventional driving on the non priority road (green solid), conventional driving on the priority road (orange solid), automated driving on the non priority road (green dashed) and automated driving on priority road (orange dashed). The results shown are remarkable in picturing what the algorithm aims to achieve. Looking at the human delays, one can see how the difference in routing is closing up with the penetration rate, meaning that the control is able to redistribute the delay more evenly. This goal is fully achieved in the results of the automated vehicles which are always controlled. At 20% penetration rate the automated vehicles on the priority rule have a higher delay than the conventional counterpart because the algorithm coordinates the crossing of controlled i-platoons independently of their routing. The fact that the decrease of delay for automated vehicles is also affected by the penetration rate indicates that the ability to efficiently control the traffic increases with the number of i-platoons to control. Considering the Figure 5.2d, between 40% and 80% of penetration rate, less than 20% of conventional traffic is part of a controlled i-platoon.

5.4. Scenario 3 - Medium-High Traffic

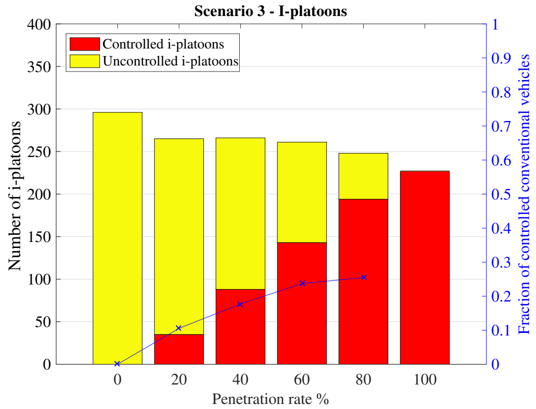
Scenario 3 is characterized by a total volume of 1200 veh/h, 600 veh/h each route. During the simulation, approximately 300 vehicles are simulated. With the increase of volume, the traffic on the non-priority road is approaching its capacity. As result, the average vehicle delay at self-organization has significantly increased in this scenario along with the number of stops which surpass the number of stops caused by the FTL Controller. Despite the higher number of stops, the average vehicle delay with the CCO Controller is still lower then the one from its counterpart FTL. This might be due to the fact that once vehicles come to full stop they do so for a very short time. Overall, Figure 5.3a shows that the CCO Controller still performs better than the FTL Controller in terms of average vehicle delay for all penetration rate and in terms of number of stops from the penetration rate of 20%. The linear decrease profile glimpsed at Scenario 2 has become prominent in the current scenario. By the introduction of 20% of automated vehicles, the vehicle delay is almost halved. The trend continues with the addition of automated vehicles at 40%, after which, the decrease of delays slows down. Figure 5.3b offers a closer look at how the average delays affect the vehicles. The drop of delay at 20% penetration rate is experienced by the conventional vehicles driving on the non-priority road. At this point, only 10% of this category is part of a controlled i-platoon and the rest is



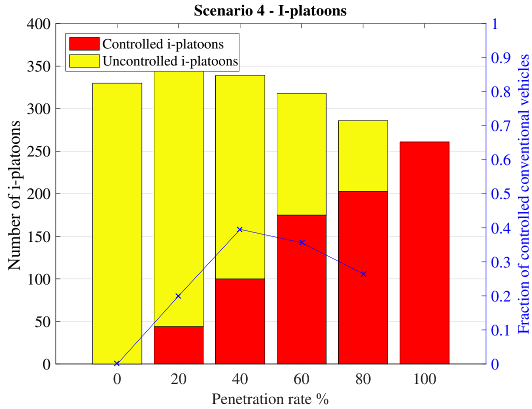
(a) Delay and Stops in Scenario 3



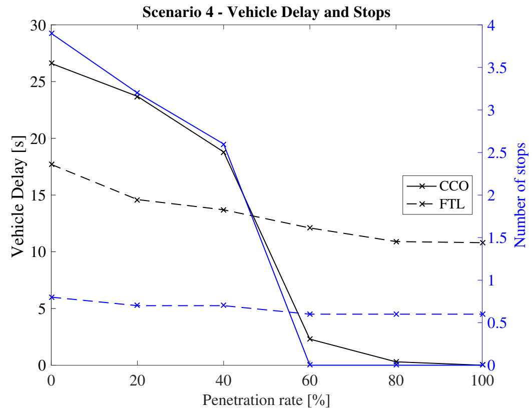
(b) Delay for Drivers in Routes in Scenario 3



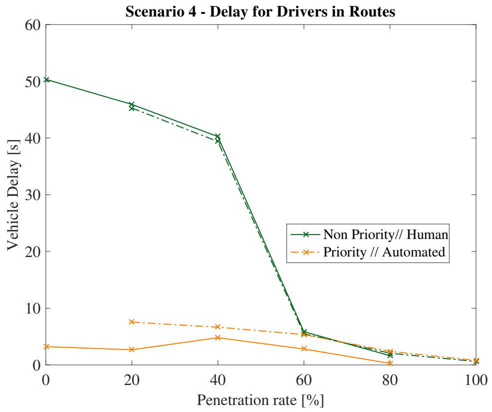
(c) Number of *i*-platoons in Scenario 3



(d) Number of *i*-platoons in Scenario 4



(e) Delay and Stops in Scenario 4



(f) Delay for Drivers in Routes in Scenario 4

Figure 5.3: Results for Scenario 3 and Scenario 4

following the right-of-way.

5.5. Scenario 4 - High Traffic

Scenario 4 is characterized by a total volume of 1600 veh/h, 800 veh/h each route. During the simulation, approximately 400 vehicles are simulated. In this final scenario, the amount of traffic volume driving the non-priority road reaches its near capacity. Unlike other scenarios, this time the self-regulated traffic at 0% penetration rate has a higher delay than the fixed traffic control program. The introduction of automated vehicles does not immediately improve the situation. At 20 % the performance of the CCO controller is still worse than the FTL simulation. The reason for this has to do with the fact that the trajectory control, in this case, decreases the average speed and increases the queue. The more the queue increases the less controlled i-platoons are available, which doesn't help to regain control over the traffic. This happens both in the priority road and the non-priority with obvious greater impact for the latter.

CHAPTER 6

Discussion

Chapter 5 presented the results of the evaluation of the controller in different scenarios. The evaluation was aimed at investigating the intersection performance as result of the control strategy applied. In this chapter, the results of this investigation will be discussed. In light of this discussion, a reflection on the model and its real implementation will also be included. Finally, the real implementation of the control strategy is considered.

6.1. Discussion on Results

This research aim to propose a new traffic control concept that would allow to regulate traffic with conventional and automated vehicles only by wireless communication. The proposed strategy is a combination of self-organization and individual trajectory control. The efficiency of this strategy is enhanced by the concept of induced platoons, which allow to indirectly extend the trajectory control feature to conventional vehicles. Once under trajectory control, the movement of platoons were regulated so that the crossing schedule of all platoons present would cause the minimum delay. The underlying principle was that controlling just a portion of the traffic with the aforementioned strategy would have yield better results than controlling the whole traffic with traditional strategies. In order to investigate in which scenarios this principle would hold, a traffic simulation was executed, comparing the performance of the proposed strategy and a traditional fixed- control signal scheme.

The results shown that the traffic control strategy is able to improve the efficiency of the traffic control under unsaturated conditions (scenario 1,2,3). In this case the coordination of vehicles lowers the chances where stops are needed and the make the journey of vehicle hindered-free. This results is achieved already with low penetration rate of 20%. Along with a reduction of delay, the strategy achieves a higher fairness in the delay distribution, as vehicles (in form of i-platoon) are sorted based on their exit time.

During saturated condition (scenario 4), the control strategy shows a performance drop. At the lowest penetration rate 20%, the performance is even worst than traditional strategies and overturns only with higher penetration rate, in particular from the rate 60%. When uncontrolled platoons start queuing on the minor road and there is not enough platoons on the major road to slow down in favor of the minor road, the traffic condition cannot be improved. This results consist in a drawback of the strategy because it relates to combination with self-organizing rules which do not allow for control over conventional vehicles.

It can be noted that the scenario with penetration rate at 0% running the CCO controller equals to a non-signalized intersection regulated only the right-of-way rule. The fact this scenario performs better than the scenario with traffic light, indicates that the intersection with that traffic flow is actually better off without any control in the first place, which defeats the purpose of replacing the exiting traffic control with the I2V control. Another cause for such a difference could be the fact that the simulated intersection model is quite straightforward and its simplicity doesn't call for a traffic light. Nevertheless, it can also be argued that in some situations traffic lights are not installed only to improve the traffic delay but also to improve the traffic safety. In this case, it is reasonable to offer the translation to CCO when such control performs better.

6.2. Reflection on the Methodology

Considering the performance of the CCO controller alone, the results reported match expectation and they give some insights of the potential application of the proposed concept strategy. Within this context, reflections on the methodology used in terms of designing choices and their evaluation are provided in this section.

6.2.1. Reflection on the Model

In the early stages of the design process, the idea to use induced platoon lead to differentiate two scenarios, one with only controlled induced platoons and one with uncontrolled i-platoons. Given the dynamic feature of the induced platoon, these scenarios cannot be differentiated in time, i.e. they coexist in the intersection. In order to provide a universal approach while remaining coherent to the intersection environment, the self-regulating approach was chosen as ruler of uncontrolled i-platoons. This choice narrowed down the potential adaptability of the control signal to different intersection layouts. Another solution could have been the usage of a traffic sign that display go/no go to the approaching uncontrolled i-platoons. Using known traffic rules for uncontrolled i-platoons instead of introducing new traffic sign is believed to be a better option despite the consequential restrictions.

Following the findings of the literature review, the control signal was computed with an optimization-based method. The challenging goal to optimize both the crossing schedule and the trajectory control was solved by a bi-level optimization. The lower level was assigned to the determination of the i-platoon trajectory, optimized for controlled i-platoons and predicted for uncontrolled i-platoons. Given the constraints, the trajectory control was reduced to a very simple and straightforward optimization. The simplicity of such control is fundamental because the trajectory of the same i-platoon is recursively computed in different sequences. The prediction of uncontrolled i-platoons was computed using a car-following model. Even though the introduction of induced platoons helps to reduced the computation effort of this prediction (only the leader movement is predicted), the repetitive application of car following model require still considerable effort. Considering the possible detection inaccuracies of the radar technology, the usage of such high-accuracy but high-effort model might not be worth it. Using a simpler model to detect the conventional vehicle could have been a more efficient choice. Within this line of though, it could even be proposed to allocate a time frame to conventional vehicle based on best and worst case instead of assuming an exact entry time. The upper level of optimization determine the optimal feasible sequence.

Within the solution space of feasible sequences, the branch and bound technique was used a search algorithm to find the optimal solution as quickly as possible. The solution space was represented by decision tree enumerating all feasible sequences. This scheme allows to easily implement special strategy control routing-specific as well as vehicle-specific such as priority to ambulances or buses.

6.2.2. Reflection on the Simulation

In order to test the performance of the proposed control strategy, a simulation platform based on the integration of Vissim and Matlab was used. The choice of using a well developed traffic simulator with user-friendly interface was appropriate, considering the major computing effort that went into building the code for the controller and correctly integrating the whole simulation. For the same reason, early on during this process, it was decided that building an external car-following model for the simulation of automated vehicles was not worth the effort and that the ability to change speed within Vissim-COM would have sufficed. However, this decision did not lead to expected results. First of all, only later in the development phase it was discovered that the changed speed was implemented prior to the next-step computation and not after the computation. The fact that the speed that determines the actual movement cannot be changed let to a undesired situation that called for workarounds measures. Secondly, the implementation of the trajectory control in the simulation through speed control required to send the speed signal every simulation step 0.1 in order to ensure the vehicles were moving according to the control signal and not according to the driving behavior modelled by Vissim. The required constant communication between Matlab and Vissim slowed down the simulation time significantly.

The test-intersection used for the simulation was quite basic. Only one lane per direction was chosen and only two directions were included. While this set-up fails to investigate the performance of the controller in a more realistic environment, its simplicity is essential to first focus the computation effort on the correct implementation of the strategy. Using a simple intersection in the simulation is actually a recurring practise in researches that propose a new strategy concept (Jia et al., 2007),(Yan et al., 2008).

6.3. Real Implementation

Practical trials of connected and autonomous vehicle technology are underway on public roads in different cities of Europe, with public funding of the European Commission. The trials are being conducted in daily traffic conditions within a test-bed areas (Kernstock, 2017). The control strategy of this research could be a possible candidate for such testing since it was shown that already a low penetration rate of automated vehicles can deliver tangible results. It is very likely that if that were to happen, the test-area would be a unsignalized intersection, rather than a signalized intersection with shut down traffic lights. In fact, as already stated in Section 3.2.1, the best case-studies for implementation of this strategy would be simple intersections with low/medium traffic volume where right-of-way rules are easy to apply.

However, the investigation of this research did not yet considered several factors that would be crucial in a the live-test implementation of the control strategy. Namely, the effective merging of *I'm-here* messages from automotive vehicles with the radar detection data;

the dependency of the model to measurement accuracy and possible miss-detection of vehicles; latency of the V2I communications.

Another aspect that is relevant to the real implementation is the computational time required by the controller to optimize the control signal at every measurement update. In this research the latency of V2I is not considered and the signal update was sent every second, meaning that the signal computational should take no later than 1 s. In order to give an impression on whether this is feasible, the computational time of the Matlab code during the simulations is here reported. The computation time includes all the algorithms run to compute a new signal, from the moment the system state is calculated (traffic modelled in i-platoon) to the moment the optimal sequence with optimal entry times is computed. The computational times are reported in relation to the number of i-platoon the controller has to optimize at once. Five simulations from the scenario 4.3, total volume 1600 veh/h and 20-100% penetration rate are considered. The simulation was run on a Windows laptop with CORE i5, 2.60 GHz of CPU, 4GB RAM. The results are shown in Figure 6.1 and Figure 6.2.

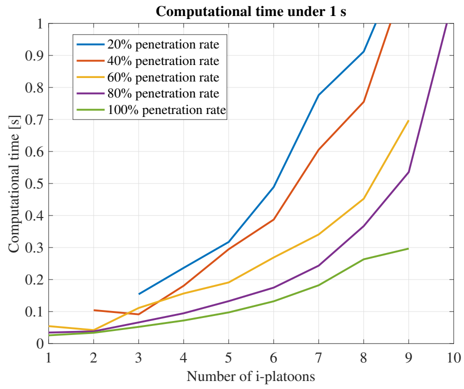


Figure 6.1: Computational time under 1 s

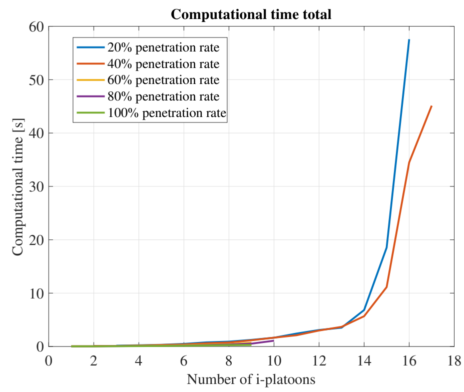


Figure 6.2: Maximum computational time used

The first figure shows how many i-platoon the controller is able to optimize under 1 second of computational time. In general, no more than 9 i-platoon can be optimized at the same time. It can be seen that the same amount of i-platoon require less computational times as the penetration rate increases. This result is due to the high effort in computing the predicted trajectories of uncontrolled i-platoon.

The second figure shows the maximum computational time required during the simulation. For the penetration rate 20% and 40%, the increase of i-platoon to optimize at the same time exponentially increases the computational time to a maximum of 60 s.

It should be noted that real implementation of traffic control is done with higher computational power than the laptop used in this simulation. In addition, the code used to model the controller can definitely be improved and written more efficiently. All things considered, it can be stated that the solution presented in this research shows potential feasibility for its implementation in a real case study.

CHAPTER 7

Conclusions and Recommendations

This last chapter builds on the previous chapters by drawing conclusions from the results and by formulating recommendations for future work. In the first section every research sub-questions is answered leading to main research question, where the main findings of this project are presented. Following, the second section includes recommendations for further research.

7.1. Conclusions

The ultimate goal of the research consisted in designing a traffic controller able to safely and efficiently coordinate a mixed flow of human drivers and automated vehicles, by relying purely on wireless communication.

The scope of the research was to investigate the problem from a traffic engineering perspective. Thus, more focus was given on the methodology choices and techniques needed to design a control algorithm that is be able to perform according to the stated objectives. A set of sub-question were formulated in order to support this investigation. The answer to these sub-questions reads as follow:

1. **What control approach is the most suited to deal with the traffic environment with a mixed flow of conventional and automated vehicles**?

   This research aimed to propose a new traffic control concept that would allow to regulate traffic with conventional and automated vehicles only by wireless communication. Within this framework, the proposed control strategy is a combination of self-organization of conventional vehicles and the individual trajectory control of automated vehicles. The efficiency of this strategy is enhanced by the concept of induced platoons, which allow to extend the trajectory control feature to a portion of conventional vehicles. The presence of conventional vehicles makes necessary a prediction module able to predict the future state of the vehicles since it cannot be controlled. The control signal is based on this prediction, thus a feedback loop is essential to check whether the traffic is moving according to the prediction and whether the control signal needs to be recalculated. This is particularly important when the traffic lights are taken out of the environment and the trajectory control should account also for collision avoidance. The choice of consider individually the conventional vehicles (in form of uncontrolled *i*-platoon) is enabled by the presence of traffic radar and its tracking technology. Without these sensors, the control approach would have not been feasible.

2. **How the coordination of the right-of-ways can be assigned?**

   The coordination of the right-of-way can be computed by a bi-level optimization. The upper layer consider all feasible sequences of vehicles according to the FIFO rule applied within each stream. For each feasible sequence, at the lower layer, the trajectory of the vehicles are optimized with the constraint of minimum entry times, in order to avoid collisions. According to the resulting delay, the sequence with least delay is chosen. In addition, the delay assigned to the sequence takes also in consideration the probability of conventional vehicle to follow the chosen sequence in order to improve the accuracy of the coordination. The optimal sequence is chosen to be the one that yields the least delay for all i-platoon considered. This means that the implementation of the control signal doesn't necessary mean that the journey of vehicles is individually optimized. The optimal sequence might lead to an increase of delay for some vehicles while it decreases it for others. In particular, when the penetration rate is low, the results have shown that automated vehicles have a higher delay than conventional vehicles.

3. **How can the speed profile of the automated vehicles be parameterized in order to optimize them?**

   The speed profile can be parameterized by considering the speed as function of acceleration and time. The resulting profile consist of piece-wise linear functions with slopes of the line segments equal to acceleration values. The number of parameters (acceleration and time) can vary depending on the designed level of flexibility and computation time available. Given the constraint on speed given by the induced platoon, in this research a two parameters trajectory has been considered. Vehicles can either cruise to the intersection or slow down and then cruise in order to arrive at the allocated entry time while driving continuously. Results show that this control does not impact negatively the speed of vehicles, in fact the space mean speed for high penetration rate (above 60%) does not decrease more than 2 km/h on average. At the same time, vehicles in these scenario avoids to come to full stop altogether.

In order to quantify the effectiveness of the solution proposed and investigate further the condition for its hypothetical deployment, additional sub-questions were formulated. The answer to these sub-questions reads as follow:

4. **Which is the range of penetration rate for automated vehicles that is necessary to affect effectively the human driving?**

   The presence of automated vehicles can directly or indirectly affect the human driving. If the conventional vehicle is part of a controlled i-platoon, the movement of the vehicle is directly controlled as result of the trajectory control of the i-platoon leader. When the conventional vehicles is part of a uncontrolled i-platoon, its movement can still be influenced if a controlled i-platoon is driving in the conflicting stream. From the results presented in Chapter 5, only a very low fraction of conventional vehicle was ever part of a controlled i-platoon. On average, in all traffic scenarios this fraction did not exceed 0.3 for any penetration rate. Nevertheless, judging from the average vehicle delay of conventional vehicles, the combination of both direct and indirect control leads to a delay reduction of almost 50%. This happens with a penetration rate of 40% in all considered traffic volumes. From the perspective of a control strategy that specifically aims to control a mixed traffic of conventional and automated vehicles, a minimum threshold value of 40% is quite high. Especially in terms of timing application, the implementation of such penetration rate is really long term in the future. The controller does perform well with lower penetration rate but this happens only with low traffic volume.

5. **What is the improvement in terms of efficiency and qualitative driving comfort for different traffic flows compared to traditional signal control?**

   The improvement of the proposed strategy is inversely proportional to the increase of volume and is directly proportional to the increase of penetration rate, both in terms of efficiency and qualitative driving comfort. With low traffic, average vehicle delay is decreased by a range from 93% to 100% compared to the delay of traditional signal control. At the same time, no stops have been reported for vehicles driving in the intersection controlled by the proposed strategy. This means that their journey is unhindered, thus providing a much more comfortable drive compared to traditional control for which one every two vehicles always has to stop at the red light. With higher traffic, average vehicle delay has an improvement ranging from -47% to 100%. The number of stops registered indicates that in these scenarios vehicles might stop more than once during their journey which is considered very uncomfortable. The highest improvement is achieved when the penetration is higher than 60%.

6. **What are the implications of applying the control scheme to a more extended network of multiple intersections?**

   The proposed control strategy has been designed to coordinate traffic at an isolated intersection. If this control strategy were to be applied over a network of intersection, the resulting network-based control would take the shape of a decentralized, bottom-up approach. Considering a network of intersection within the framework of connected vehicles, comes with the assumption that the road-side units at the intersections can communicate with each other. Information over vehicles and signal control decision can be shared within the surrounding intersections. Depending on the distance between the intersections, this means that the controller gets information of approaching vehicles even earlier than the defined *detection zone*. The availability of such information can bring an improvement to the performance of the controller because vehicles have more time to adjust their speed to the designated entry time, leading to a possibly higher entry speed. In the isolated intersection, the control signal was terminated after the i-platoon had crossed the intersection area. If the i-platoon leader would continue to execute its control signal until reaching the *communication zone* of the following intersection, the formation of i-platoons would also be preserved. Considering the drawbacks of this strategy, (Section 5.1) the independent control of the intersection can lead in some scenarios to grid lock. Therefore it might be appropriate to have a higher scheme that define priority of roads of the network based on the traffic density. This improvement would be easily implemented within the current control scheme by adding a weight to the cost function in scheduling optimization (Section 3.4.2)

Finally, the main research question can be answered:

How can the performance of an intersection be improved based on a certain objective by using a speed and sequence control strategy for traffic of mixed level of automation?

A new traffic control concept has been introduced to allow the regulation of traffic with conventional and automated vehicles only by wireless communication. The proposed strategy is a combination of self-organization and individual trajectory control. The efficiency of this strategy is enhanced by the concept of induced platoons, which allow to indirectly extend the trajectory control feature to conventional vehicles. Once under trajectory control, the movement of i-platoons were regulated so that the crossing schedule of all i-platoons present would yield the minimum delay possible. The underlying principle was that controlling just a portion of the traffic with the aforementioned strategy would have yield better results than controlling the whole traffic with traditional strategy. Results have shown that the traffic control strategy has been able to improve the efficiency of the traffic control mainly unsaturated conditions with already low penetration rates of 20%. With saturated conditions, a higher penetration rate of 60% is required.

7.2. Recommendations

During the development of this research, different ideas for additional working directions arose. Some consider expansion of the proposed control strategy to a more complex environment in order to investigate further its applicability. Others ideas offer possible improvements on the control strategy itself that might be worth to consider. These ideas are here listed as suggestions to future research.

The first field of investigation concern expansion of the intersection layout, and the number of intersections considered. Adding new lanes and route decisions (not only straight direction) implies that the controller design needs to deal with an increase of vehicle and new implication on the scheduling of vehicles. It might be needed to add an extra module that is able to translate this new situations in the terms that the current design can respond to. The profit of such investigation is been able to assess whether the design choices made for the simple layout are able to cope with a more complex environment. Considering additional intersections controlled by their own controllers will be able to investigate the consequences of individual coordination of intersection within the network performance. A theoretical discussion on the implications of a network-based control was already proposed in the answer to the last sub-question. With conventional traffic control approaches, it is known that cooperative traffic control between intersection leads to better results. However, within the new control approach few characteristics might lead to a different conclusion on network performance.

The improvements of the control design are considered given the simple layout of the intersection. These improvements would lead to a better performance of the controller and ultimately of the intersection. They include the following aspects:

- **Allowing induced platoons to speed up**. One-vehicle controlled i-platoons or i-platoons formed only by automated vehicles do not need to enforce the speed constraint ($v(t) \leq v(t-1)$) because their formation is constant in time and space. If possible, allowing them to speed up to the speed limit reduces the minimum entry time as well as the exit time. This would lead to a more efficient use of the crossing area, as evacuation time is reduced.
- **Considering connected vehicles with automation level 1-2**. Connected vehicles with

at least 1-2 level of automation refer to vehicles able to perform adaptive cruise control. The adaptive cruise control could be used by the controlled to assign connected vehicles to follow their preceding vehicles at the defined headway. This would mean that the i-platoon time length would be more constant and smaller, so also in this case the evacuation time is reduced.

- **Allowing an optimization of the deceleration**. At this time the maximum deceleration is constant for all vehicles. Given the objective to maximize the speed given the entry time, the maximum deceleration is always chosen. If in the order sequence, the second i-platoon to enter can only enter a minute after the first vehicle has passed with its minimum entry time, than the first vehicle could have entered later with a smother deceleration.
- **Use constant inter-platoon headways directly as constraint**. In order to keep the formulation of induced platoon constant, a speed constraint was used under the assumption that acceleration can cause the i-platoon to split. The constraint is applied in the design of the trajectory. By defining the constant inter-platoon headways as constraint, small accelerations would be allowed as long as it is not predicted that the headway would increase over the splitting threshold. In doing so, the headway behavior is not just assumed constant anymore but it would be predicted. The accuracy of the exit time would than improve.

References

Abboud, K., Omar, H., transactions On, W. Z. I., & 2016, U. (2014). Interworking of DSRC and cellular network technologies for V2X communications: A survey. *Ieeexplore.Ieee.Org*.

Al-Sultan, S., Al-Doori, M. M., Al-Bayatti, A. H., & Zedan, H. (2014). A comprehensive survey on vehicular Ad Hoc network. *Journal of Network and Computer Applications*, *37*(1), 380–392. doi: 10.1016/j.jnca.2013.02.036

Buckley, D. J. (1968). A Semi-Poisson Model of Traffic Flow. (May 2018).

Busse, K. T. (n.d.). A functional combination of platooning and traffic-adaptive intersection control.

Chang, H.-J., & Park, G.-T. (2013). A study on traffic signal control at signalized intersections in vehicular ad hoc networks. *Ad Hoc Networks*, *11*(7), 2115–2124. doi: 10.1016/j.adhoc.2012.02.013

Dresner, K., & Stone, P. (2008). A multiagent approach to autonomous intersection management. *Journal of Artificial Intelligence Research*, *31*, 591–656. doi: 10.1613/jair.2502

European Commission. (2018). On the road to automated mobility: An EU strategy for mobility of the future. *COMMUNICATION FROM THE COMMISSION TO THE EUROPEAN PARLIAMENT, THE COUNCIL, THE EUROPEAN ECONOMIC AND SOCIAL COMMITTEE, THE COMMITTEE OF THE REGIONS*, *09*(02), 1750022. doi: 10.1142/S1793830917500227

Feng, Y., Head, K. L., Khoshmagham, S., & Zamanipour, M. (2015). A real-time adaptive signal control in a connected vehicle environment. *Transportation Research Part C: Emerging Technologies*, *55*, 460–473. doi: 10.1016/j.trc.2015.01.007

Florin, R., & Olariu, S. (2015). A survey of vehicular communications for traffic signal optimization. *Vehicular Communications*, *2*(2), 70–79. doi: 10.1016/j.vehcom.2015.03.002

Gipps, P. G. (1981). A behavioural car-following model for computer simulation. *Transportation Research Board*, *15*(2).

Hegyi, A. (2012). *A design methodology for traffic control systems* (Traffic Ma ed.). TU Delft, Internal publication.

Hoogendoorn, V., Serge; Knoop. (2016). *Traffic flow theory and modelling.* Internal TU Delft publication.

Hoogendoorn, Sergie P. ; Botma, H. (1997). Modeling and estimation of headway distributions. *Transportation Research Record: Journal of the Transportation Research Board*(1591), 14–22.

Ilgin Guler, S., Menendez, M., & Meier, L. (2014). Using connected vehicle technology to improve the efficiency of intersections. *Transportation Research Part C: Emerging Technologies.* doi: 10.1016/j.trc.2014.05.008

Jia, W., Abbas-Turki, A., Corréïa, A., & El Moudni, A. (2007). Discrete intersection signal control. *2007 IEEE International Conference on Service Operations and Logistics, and Informatics, SOLI*(ii). doi: 10.1109/SOLI.2007.4383891

Jing, P., Huang, H., & Chen, L. (2017). An Adaptive Traffic Signal Control in a Connected Vehicle Environment: A Systematic Review. *Information*, *8*(3), 101. doi: 10.3390/info8030101

Katsaros, K. (2011). Performance study of a Green Light Optimal Speed Advisory (GLOSA) Application Using an Integrated Cooperative ITS Simulation Platform. *Proceedings of Wireless Communications and Mobile Computing Conference (IWCMC)*, 918–923. doi: 10.1109/IWCMC.2011.5982524

Kernstock, W. (2017). *Detailed pilot overview report. C-Roads* (Tech. Rep. No. December). Vienna: ATE.

Kooijman, R. (2016). *Adaptive traffic management and C-ITS challenges in Rotterdam* (Tech. Rep. No. December). Rotterdam: Gemeente Rotterdam.

Krikke, R. (2017). *Eindrapport PPA Noord Definitief Vastgesteld door PPA Stuurgroep* (Tech. Rep. No. April). Amsterdam: Provincie Noord-Holland.

Li, J., Dridi, M., & El-Moudni, A. (2016). A cooperative traffic control of Vehicle–Intersection (CTCVI) for the reduction of traffic delays and fuel consumption. *Sensors (Switzerland)*, *16*(12). doi: 10.3390/s16122175

Li, Z., Elefteriadou, L., & Ranka, S. (2014). Signal control optimization for automated vehicles at isolated signalized intersections. *Transportation Research Part C: Emerging Technologies*, *49*, 1–18. doi: 10.1016/j.trc.2014.10.001

Lyons, S., & Babbar, S. (2017). *Market Forecast for Connected and Automated Vehicles* (Tech. Rep. No. July). London, UK: Transport Systems Catapult.

National Transportation Operations Coalition. (2012). 2012 National Traffic Signal Report Card Technical Report. *National Transportation Operations Coalition (NTOC)*.

Pandit, K; Ghosal, D; Zhang, H. M; Chuah, C. (2013). Adaptive Traffic Signal Control With Vehicular Ad hoc Networks. *IEEE Transactions on Vehicular Technology*, *62*(4), 1459–1471.

Potts, C. N., & Kovalyov, M. Y. (2000). Scheduling with batching: a review. *European Journal of Operational Research*, *120*(2), 228–249. doi: 10.1016/S0377-2217(99)00153-8

Priemer, C., & Friedrich, B. (2009). A decentralized adaptive traffic signal control using V2I communication data. *Intelligent Transportation Systems, 2009. ITSC'09. 12th International IEEE Conference on*, 1–6.

PTV. (2013). PTV Vissim 6 User Manual. *Karlsruhe, Germany*.

PTV. (2017). Connected Autonomous Vehicles Context / Overview. , 1–44.

SAE. (2016). *Surface Vehicle Recommended Practice* (Vol. J3016 SEP2; Tech. Rep.). SAE INTERNATIONAL. doi: 10.4271/2012-01-0107.

SMMT. (2017). Connected and Autonomous Vehicles: Position Paper. , 1–46.

The MathWorks Inc. (2018). *MATLAB 8.0 and Statistics Toolbox 8.1.* Natick, Massachusetts, United States.

Weil, T. (2008). *Securing Wireless Access in Vehicular Environments (WAVE)* (Tech. Rep. No. December). New Orleans: IEEE.

Xu, B., Ban, X. J., Bian, Y., Wang, J., & Li, K. (2017). V2I based cooperation between traffic signal and approaching automated vehicles. In *Ieee intelligent vehicles symposium, proceedings.* doi: 10.1109/IVS.2017.7995947

Xu, Z., Li, X., Zhao, X., Zhang, M. H., & Wang, Z. (2017). DSRC versus 4G-LTE for connected vehicle applications: A study on field experiments of vehicular communication performance. *Journal of Advanced Transportation*, *2017*. doi: 10.1155/2017/2750452

Yan, F., Dridi, M., & El Moudni, A. (2008). Control of traffic lights in intersection: A new branch and bound approach. *5th International Conference Service Systems and Service Management - Exploring Service Dynamics with Science and Innovative Technology, ICSSSM'08*, 1–6. doi: 10.1109/ICSSSM.2008.4598496

Yan, F., Dridi, M., & Moudni, A. E. (2011). A scheduling approach for autonomous vehicle sequencing problem at multi-intersections. *Int. J. Oper. Res*, *8*(1), 57–68.

Yang, K., Ilgin Guler, S., & Menendez, M. (2016). Isolated intersection control for various levels of vehicle technology: Conventional, connected, and automated vehicles. *Transportation Research Part C*, *72*, 109–129. doi: 10.1016/j.trc.2016.08.009

Zhang, S., Chen, J., Lyu, F., Cheng, N., Shi, W., & Shen, X. S. (2018). Vehicular Communication Networks in the Automated Driving Era. *IEEE Communications Magazine*, *56*(9), 26–32. doi: 10.1109/MCOM.2018.1701171