RWTH Aachen

Department of Computer Science
Technical Report

Experimental Validation and Mathematical Analysis of Cooperative Vehicles in a Platoon

Hilal Diab

ISSN 0935–3232 · Aachener Informatik-Berichte · AIB-2015-07

RWTH Aachen · Department of Computer Science · April 2015

Experimental Validation and Mathematical Analysis of Cooperative Vehicles in a Platoon

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der
RWTH Aachen University zur Erlangung des akademischen Grades
eines Doktors der Ingenieurwissenschaften genehmigte Dissertation

vorgelegt von

Master of Science
Hilal Diab
aus Damaskus

Berichter: Universitätsprofessor Dr.-Ing. Stefan Kowalewski
Universitätsprofessor Dr.-Ing. Dirk Abel

Tag der mündlichen Prüfung: 09. Mai 2014

Hilal Diab
Lehrstuhl Informatik 11
diab@embedded.rwth-aachen.de

Aachener Informatik Bericht AIB-2015-07

Herausgeber: Fachgruppe Informatik
RWTH Aachen University
Ahornstr. 55
52074 Aachen
GERMANY

ISSN 0935-3232

Acknowledgement

First and foremost, I present my sincere gratitude to Prof. Dr.-Ing. Stefan Kowalewski for giving me this opportunity to write this thesis and providing me the interesting topic and his research guidance.

Also I would like to express my gratitude to Prof. Dr.-Ing. Dirk Abel for his support and for his role as the second reviewer of this thesis.

This work was partly supported by the German Research Foundation (DFG) within the priority program SPP1305, which I would like to thank for their financial support.

Without help and support, this thesis would not have been possible. Hence I would like to thank my colleagues at the department of the Embedded Software Laboratory RWTH Aachen, especially Ibtissem Ben Makhlouf, Eva Beckschulze and Paul Hänsch for the productive working atmosphere, Dr.-Ing. Jan Maschuw and Dr.-Ing. Martín Guillermo Chávez Grunewald from the Institute of Automatic Control RWTH Aachen for the good cooperation on the publications.

A special thank goes to Thomas Gatterdam for his contributions to the experimental setup, and the students Martin Lang, Jochen Schmücking, Ali Tarzan and Lars Tholen for their commitment.

I thank my family for their continuous support.

Jana, I thank you for the wonderful years.

Hilal Diab
Böblingen, April 2015

Abstract

The infrastructure of roads in many countries is congested because of the rapid increase of the amount of traffic flow in the past few years. A significant part of this is due to the increase of freight transport, which is continuing to grow. Nowadays, safety on the roads and saving fuel should also be taken into consideration while finding the solutions to road congestion. One suggested solution to the problem is to increase the capacity of the highways.

This can be realized by forming platoons of vehicles and automatically controlling the vehicles in order to maintain short, but at the same time, safe distances between the different vehicles within the platoon. The development of intelligent sensors and electronic control systems within vehicles make autonomous driving in platoons possible. Enabling vehicular wireless communication between the items of the platoon improves the performance of the controlled platoon significantly. Furthermore, enabling the platoon to communicate with other traffic members, such as other vehicles, intelligent traffic lights and infrastructure road units, allows the platoon to perform more complex driving scenarios for autonomous vehicles, such as crossing intersections.

The first part of this thesis investigates the behavior of the platoon when only communication between its items is possible. The safety of the vehicles within the platoon is investigated, taking the effects of the communication faults on the behavior into account. This analysis could be helpful in the process of controller design, where the developed controller should ensure stability despite network failures and should achieve an optimal performance in every situation. Therefore, the verification of the controller behavior was investigated by formal verification methods: a reachability analysis of a dynamic and hybrid system. The safety of practical relevant scenarios was checked.

In addition, a hardware platform was set up to test the platoon's behavior under the influence of hardware shortage, such as noises and time delays caused by hardware components. A 1:14 scaled platoon of four trucks equipped with sensors and WiFi modules was designed. This platform was used for testing different cooperative vehicle platoon controllers by examining their performance and influence on the safety in case of communication problems within the platoon.

The second part of this thesis studies the behavior of the platoon when communication between the platoon and the intersection road unit is enabled. The following scenario has been considered: When a platoon of autonomous vehicles following a leader approaches an intersection, the platoon should change its highway mode to other modes in order to cross the intersection safely and efficiently. To realize that, information about the actual position of the platoon together with information of other vehicles in the intersection area are needed. Based on this information the platoon has to decide which mode should be performed. Therefore, we extended our platform with an indoor positioning system which is able to provide the position to the objects in a test environment independently. An intersection management system was implemented in order to test different scenarios related to different crossing modes of the platoon. Results showed that platoons can be controlled efficiently and safely while crossing the intersection.

Zusammenfassung

Die Infrastruktur der Straßen ist in vielen Ländern überlastet durch die rasche Zunahme des Verkehrsflusses in den letzten Jahren. Ein großer Teil davon kann auf den Anstieg des Lasten- und Güterverkehrs zurückgeführt werden. Ein Ende des Wachstums ist nicht zu erwarten. Die Sicherheit auf den Straßen und das Sparen von Kraftstoff sollte bei der Suche nach einer Lösung berücksichtigt werden. Ein weit verbreiteter Vorschlag basiert auf der Erhöhung der Kapazität der Autobahnen.

Eine Möglichkeit dies zu erreichen ist die Bildung von Kolonnen und das automatische Steuern der Fahrzeuge um einen möglichst geringen, aber gleichmäßigen Abstand und eine sichere Entfernung zwischen den verschiedenen Fahrzeugen innerhalb der Kolonne zu gewährleisten. Die Entwicklung intelligenter Sensoren und elektronischer Steuerungssysteme machen das autonome Fahren in der Kolonne möglich. Das Einführen einer drahtlosen Kommunikation zwischen den Fahrzeugen der Kolonne verbessert die Kontrolle des Zuges deutlich. Darüber hinaus ermöglicht die Kommunikation der Kolonne mit anderen Straßenteilnehmern, intelligenten Ampelsystemen und Infrastrukturkommunikationseinheiten das Ausführen komplexerer Fahrszenarien, z.B. das Überqueren von Kreuzungen.

Der erste Teil dieser Arbeit untersucht das Verhalten der Kolonne, wenn eine Kommunikation nur zwischen den Elementen der Kolonne möglich ist. Hierbei wird die Sicherheit der Fahrzeuge innerhalb des Zuges, unter Berücksichtigung der Auswirkungen von Kommunikationsfehler, untersucht. Diese Analyse könnte für die Entwicklung des Controllers hilfreich sein, da dieser trotz Netzwerkfehlern stabil laufen und eine optimale Leistung in jeder Situation erreichen sollte. Aus diesem Grund wurde das Verhalten des Controllers durch formale Verifikationsmethoden überprüft. Dafür wurde eine Erreichbarkeitsanalyse eines dynamischen und eines Hybrid-Systems durchgeführt.

Weiterhin wurde eine Hardware-Plattform eingerichtet um das Verhalten der Kolonne von Fahrzeugen unter dem Einfluss von Hardwareproblemen, wie Zeitverzögerungen und Störungen, zu untersuchen. Die Plattform besteht aus vier 1:14 skalierten LKWs, welche mit Sensoren und WiFi-Modulen ausgestattet sind. Die Plattform wurde benutzt um die verschiedenen Controller hinsichtlich ihrer Leistungsfähigkeit und ihrem Einfluss auf die Sicherheit im Falle von Kommunikationsproblemen innerhalb der Zuges zu überprüfen.

Der zweite Teil dieser Arbeit beschäftigt sich mit dem Verhalten der Kolonne, wenn eine Kommunikation zwischen der Kolonne und der Infrastrukturkommunikationseinheit stattfindet. Wenn sich eine Kolonne bestehend aus autonomen Fahrzeugen, die einem Führungsfahrzeug folgen, einer Kreuzung nähert, sollte der Autobahn-Modus in einen anderen Modus geändert werden, um die Kreuzung sicher und effizient zu überqueren. Um dies zu ermöglichen werden Informationen über die aktuelle Position des Zuges, sowie Angaben über das Vorhandensein anderer Fahrzeug im Kreuzungsbereich, benötigt. Anhand dieser Informationen muss die Kolonne entscheiden, welcher Modus ausgeführt werden soll. Aus diesem Grund wurde die Plattform um ein indoor positioning System erweitert, welches in der Lage ist die Positionsinformationen an die Objekte in einer Testumgebung unabhängig voneinander weiterzugeben. Weiterhin wurde ein Kreuzungsmanagementsystem eingeführt, um die verschiedenen Szenarien des Überquerens einer

Kreuzung durch eine Kolonne zu untersuchen. Die Ergebnisse zeigten, dass Kreuzungen, die von Kolonnen überquert werden, effizient und sicher gesteuert werden können.

Contents

List of Tables

List of Figures

1 Introduction

The mobility of persons is very important nowadays. Although there exist other ways of transportation, vehicles on the roads are still the main mode of transport. Due to that, the number of manufactured and used vehicles has increased significantly over the last years. This growth has its direct effect on the transportation and road infrastructure network. In addition, freight transport continues to grow, blocking large parts of road areas; thus, avoiding congestion is becoming an important issue. In several countries worldwide, the maximum road capacity has nearly been reached. The average speed travel of vehicles on highways, especially near big cities, has been reduced to almost $50km/h$ and on some roads to a total stop during the rush hours. The results are many hours of delay and a vast amount of wasted fuel due to these traffic jams. The economical effect caused by congestion is also rising, based on the time spent on the road [86]. Furthermore, a lot of unused energy is wasted due to the road jams which contribute strongly to atmospheric pollution. Another important factor in transport, which should be investigated more, is road safety. Although the number of fatal injuries on the road is decreasing, large numbers of persons are involved in accidents each year. The number of people who are killed in road accidents per year is about 1.2 million people and greater numbers are injured [4]. In addition to the human losses, economic losses of over 160 billion Euros per year are recorded.

Therefore there is an urgent need to find solutions for these problems mentioned previously, taking into consideration many factors that affect transportation in order to make traffic on the road efficient and safe. Many governments have shown significant interest where they initiated many laws in order to improve the safety factors and standards on the roads and reduce the pollution effect of vehicular transportation. To solve congestion, initial ideas included building new and larger roadways or replacing the entire traffic management system which could solve many of the transport problems. This however could involve high financial costs and would raise a lot of political and environmental issues. In addition, building new roads is not the optimal choice, especially for areas with limit spaces. Thus, better utilization of the existing traffic variables would be the practical solution. That would entail efficient traffic flow where there would be an increase in the number of vehicles on the road while avoiding congestion and accidents.

Incorporating recent technologies in transport could be the optimal solution. Reducing road congestion and fatality rates in road traffic could be possible by making the vehicle and the road network intelligent. The human driver has a slower reaction time than a machine and could lead to many undesired driving behaviors. For example, unnecessary braking from one vehicle could cause many vehicles to slow their speed massively, or even to stop, since other drivers need some time to react and they would maybe brake harder than necessary. The human factor also has a significant share in causing accidents. It

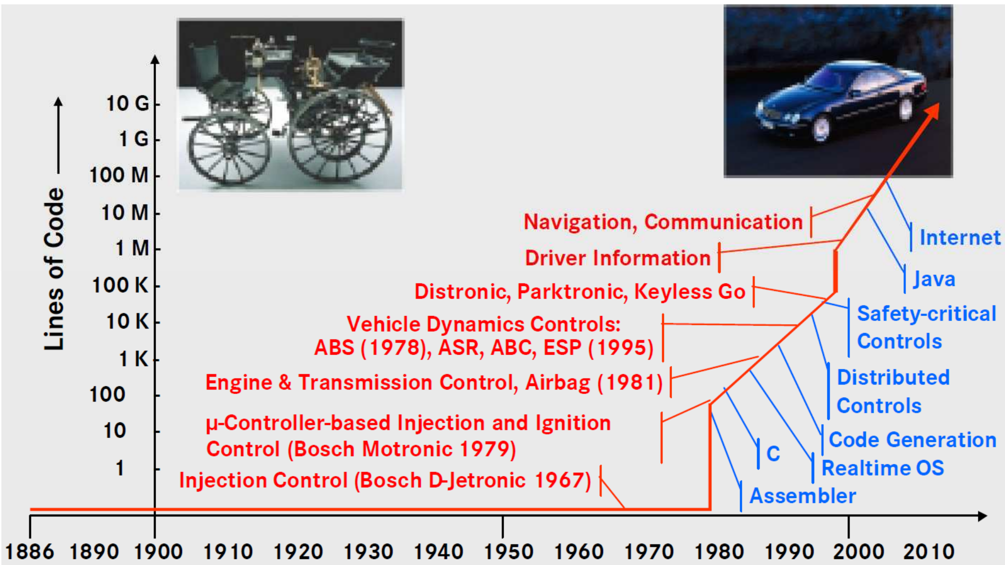


Figure 1.1: History of software-based features in vehicles [26]

was reported that human error is responsible for about 80-90% of all incidents which took place in the USA [5]. Thus, one approach which is slowly becoming reality is making the vehicle intelligent by equipping it with up-to-date electronic technologies to reach a better estimation of the surroundings.

Vehicles are no longer pure mechanical systems, but are nowadays provided with different electronic systems. The use of electronic components in vehicles has increased in recent years and modern vehicles are now equipped with computers, on-board electronic sensors which add and improve already existing functions in the vehicle. The developed functions, which are technically programmed in software, are implemented through embedded electronic control devices in combination with sensors and actuators. Figure 1.1 shows the development history of such functions and the software development. These improvements affect the chassis and body systems for passive safety and comfort, as well as systems that assist the driver and provide active safety.

These systems are called Advanced Driver Assistance Systems (ADAS). Rapid advances in sensor technology are giving the development of such systems a real push in the way of achieving more powerful systems. Excluding the ability of controlling the vehicle through the actuators, these systems support the driver by informing him in critical situations or guiding him to avoid an accident. Allowing these systems to control the movement of the vehicle (longitudinal and the lateral control), these systems should estimate the situation and take the necessary action to perform the correct assisting maneuvers.

Adding wireless communication and exact ways to determine the position, this idea of intelligent vehicle opens new opportunities of a more sophisticated level of driving: namely, collaborative driving. One of these technologies, which could be used in solving the problem of congestion and safety on the road, can be found in automated platooning technology.

A Platoon is a string of electronically connected vehicles driving together on the roads. There is always one vehicle driven manually, called the leader, while other vehicles are controlled autonomously and are called followers. Important technology enabling automatic driving in platoon formation is the ability of the vehicles to communicate between themselves, which is called vehicle-to-vehicle (V2V) communication. The communication allows the platooning format of reaching safer and more efficient cooperative driving. Driving in this formation utilizes the road optimally, keeping close spacing between the vehicles. In addition, this formation has proven to save fuel consumption due to the slipstream effect.

However, this kind of autonomous driving can be safety-critical, especially if communication disturbances such as delays or loss of data have occurred. It is therefore necessary to design controllers that are robust against these communication faults. Furthermore, hardware problems like a breakdown of a sensor could cause hazardous accidents. Thus, it is very important to test platoon controllers by means of simulation and under the effects of the real environment. In addition, once this technology is on the market it should be possible for a platoon of vehicles to drive on different parts of the roads, and not only on highways. For example, the platoon should be able to face driving on important parts of the road network, such as the intersection.

Intersections are the main cause of accidents on the roads. Improving intersection driving is thus important and will improve the safety level of the entire road network. For a vehicle which is driven autonomously, intersections are considered to be challenging since a lot of factors should be taken into account. For automated platoon driving this is also the case. Enabling the platoon to communicate with such intersection management units using vehicle-to-infrastructure (V2I) communication allows the platoon to perform more complex driving scenarios such as driving at intersections.

1.1 Objectives and Contributions

In this thesis we took the example of the platoon to investigate the safety in autonomous driving. Verification methods based on reachability analysis were developed in order to detect dangerous situations and undesired behavior. In addition, a platform of scaled vehicles was developed to capture the problems which could happen in a real environment. Since intersections are an important part of the roads we investigate the behavior of the platoon when it approaches two kinds of intersections (ones signalized with traffic lights and unsignalized ones).

The main contributions of this thesis are as follows:

- We provide a survey of other work in the field of autonomous control methods used in vehicle applications.
- We have developed a testing platform consisting of scaled vehicles driving in platoon formation.
- We have investigated the controller behavior by performing safety analysis and verification by means of the reachability analysis method.

- An indoor positioning system was introduced and evaluated for finding the position for platoon members.
- We have proposed a method and system architecture for enabling the platoon to cross an intersection safely.
- An intersection management system was tested in order to enable the platoon to cross the intersection efficiently.

The thesis presented here is part of the project ”Control of Platoons with Topologically Varying Communication Networks Based on Energy Functions” of the Priority Program 1305, the DFG-Priority Programme 1305 ”Control Theory of Digitally Networked Dynamical Systems”. The project was in Aachen with a cooperation from the Institute of Automatic Control RWTH Aachen (”Institut für Regelungstechnik (IRT)”) and the Embedded Software Laboratory RWTH Aachen (”Lehrstuhl Informatik 11 (i11)- Software für eingebettete Systeme”). In this project the Institute of Automatic Control (IRT) has designed a control strategy based on energy methods control, while the Embedded Software Laboratory (i11) investigated the safety analysis of the designed controller on the basis of hybrid automata. Taking into account the failure of communication in the platoon, a safety analysis should be considered. Here, methods for modeling and analyzing hybrid systems are applied. Constraints regarding the application and the robustness of these methods should be also examined.

Figure 1.2 shows the outline of this work. The platoon controller is written as state-space model. Adding the communication problems and losses we get a hybrid model. The behavior of the resulting hybrid system should be investigated. Safety requirements are mainly defined as having no collisions between vehicles. Then methods of hybrid verification are used to check the safety of the system. On the hardware side the controller model is designed in Simulink. Adding some blocks from other tools (target support package and later as part of the Embedded Coder tool) the control algorithm can be programmed on an MPC555 processor on which the controller as well as the incoming data were processed. Graphical User Interfaces (GUIs) have been developed to monitor the dynamic behavior of the system under full communication as well as under disturbances and to modify the parameters of the system online. The platform and the safety analysis have then been extended in order to test different test cases when the platoon approaches an intersection.

1.2 Thesis Structure

The rest of this thesis is organized as follows: Chapter 2 gives an overview of the intelligent vehicle technologies and concentrates on the platooning systems. In addition, it points out the intersection as a vital part of the road, introducing the latest technologies used to improve the efficiency of the intersection. Chapter 3 gives a description of a developed testbed used for testing platoon technologies. This includes the hardware and software description, in addition to the description of the control algorithms used to control

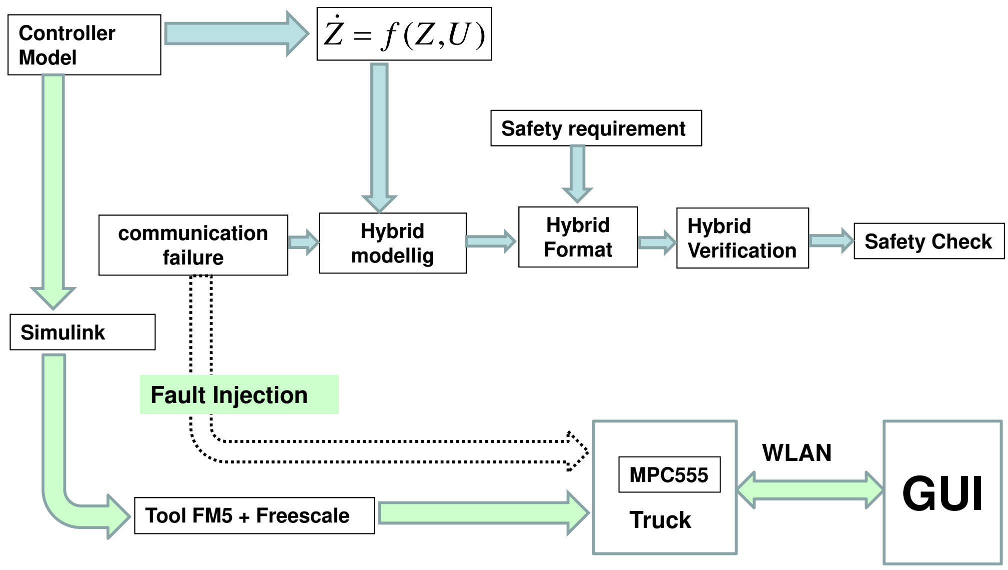


Figure 1.2: Outline of the first part of the thesis

the scaled vehicles. In Chapter 4 using formal methods in the safety verification of the platoon's behavior is introduced. Here we concentrate on the reachability analysis techniques in dynamic and hybrid systems during the verification of the platoon behavior. Chapter 5 shows how the used platform were extended to investigate the behavior of the platoon at signalized and unsiganlized intersections. The simulation environment for intersections is introduced in Chapter 6, explaining some energy saving issues. Finally, Chapter 7 summarizes the thesis and discusses possible issues for future works.

1.3 Bibliographic Notes

In this thesis three different bibliographies were used, own publications, supervised graduation thesis and other literature. A bibliographic reference that refers to an own publication of the author is composed of the initial letters of the author's surnames and the year of publication, for example [DCGBM$^+$10]. Supervised bachelor thesis are referred to with the first three letters of the students last name and the year of publication, such as [Tho12]. All other references are numbered serially. Quotation of pictures are labeled in the caption. The results presented in this thesis are partly already published in various publications and base partly on supervised graduation thesis, which will be illustrated in the following:

Details about the platform, its structure and the sensor specifications can be found in [DCGBM$^+$10]. This reference, in addition to the reference [MDAK11], gives a more detailed description of the used control methods and their characteristics. These references are cooperation publications with the IRT. In order to investigate the scenario of a platoon approaching an intersection, the publication [DBMK12] presented the necessary extension of the platform by equipping the vehicles with an indoor positioning system and by introducing an intersection management system. The modification of the positioning system, a part of the intersection management system and the hardware implementation of the collision detection was introduced in the supervised bachelor thesis of Ali Tarzan [Tar12]. The supervised bachelor thesis of Lars Tholen [Tho12] dealt with simulations and test cases for a platoon crossing signalized intersections.

In [BMMH+11] we introduce a method to verify a longitudinal vehicle control algorithm. Together with the publication in [HDMK13] the verification method was based on the reachability analysis of continuous linear systems with uncertain inputs. In [BMDK12] we consider the worst case scenario corresponding to the transition from a fully functioning communication between the vehicles to a total loss of communication. In [BMDK13] the verification using reachability analysis was also used to consider the management of a platoon crossing an intersection.

2 Cooperative Vehicle Systems: State of the Art

There has been an increasing number of embedded systems in automobiles. Nowadays, vehicles are equipped with several Electronic Control Units (ECUs). Each vehicular ECU is devoted to a selected vehicle function. Modern vehicles can have about 80 ECUs which control many functions in the vehicle, ranging from comfort to safety. Some of these systems assist the driver or even take over the control of the vehicle. The main goal of such systems is to ensure safety while the coordination of the vehicle's movement is performed. This is done by supporting the driver in some complex scenarios. Examples of such systems that can prevent dangerous situations include ABS (Anti-lock Braking System) and ESC (Electronic Stability Control).

There is currently a trend of developing systems which could partly or completely take over the driver mission. These kind of systems are called Advanced Driving Assistance Systems (ADAS). Sensors, actuators and control methods are the components for such systems. Besides, by adding communication systems, cooperative vehicle scenarios become possible in which the driving status of other neighboring vehicles could be observed. Interest in such control systems grows with the increase in traffic congestion, exhaust emissions and the safety risks of freight transport. These systems can be seen as potential solutions for a number of road traffic problems.

In this chapter the recent trend of making the vehicle more intelligent is reviewed. The enabling technologies from sensors and positioning technologies to communication systems are presented. Afterwards the platoon system will be introduced. At the end of the chapter intelligent driving on important parts of the road, such as intersections will be discussed.

2.1 Intelligent Vehicle Systems

In order to solve the transportation problem the concept of Intelligent Transportation Systems (ITS) was developed. ITS systems can be seen as using the recent technologies and trends in systems engineering concepts in order to enhance the transportation process. The intelligent vehicle systems are a smaller part of the bigger concept of the ITS. In this thesis we define intelligent vehicle systems as a vehicle which has a number of intelligent systems that have the ability to gather information about the driving environment and can either inform the driver or control the vehicle to perform complex driving operation optimally. A similar definition of the intelligent vehicle can be found in the literature such as in [18]. In this section we review the senor technologies which are needed for intelligent vehicles, including the positioning and the communication. In Addition we examine the intelligent systems with regard to the aspects of safety and traffic efficiency.

2.1.1 Sensor Technologies

There have been a good number of big automobile industries which have equipped their vehicles with the recent technologies in the field of sensors. These technologies get information about the surrounding environment and then later are used for the implementation of the developed driver assistance functions. A wise choice of these sensors is important since it has a significant influence on the quality of the whole system. Therefore the characteristics of the sensors must be carefully investigated in the development phase in order to be integrated to do the proper function.

Object Detection Systems

Finding a free spot for vehicle movement is important to avoid crashes with other objects. For the human driver, one's own senses are responsible for that, particularly his eyes and with smaller contribution his ears. In a vehicle, different sensor technologies can be used for detecting the objects or at least for measuring the distance to the next object near the vehicle.

1. Ultrasonic Sensors:

   These sensors have been used in underwater applications such as sonar before they were used in automotive industry. The sensors are mainly used in the vehicle for lateral safety and for applications with small range detection. Ultrasonic sensors are inexpensive and have a wide detection range. The distance to a reflecting object is measured by calculating the time difference between the start of transmitting an ultrasonic wave and the arrival of the reflected echo signal. The distance is then calculated by multiplying this value by the speed of sound in air. Air pressure, the temperature, and the frequency of the echo can affect the accuracy of the distance measurements.

2. Radar Sensors:

   Radar (Radio Detection and Ranging) is widely used in collision warning and collision avoidance systems. The radar uses radio waves in order to detect and calculate the distance to an object. It transmits a very short, high-intensity burst of radio waves at a high frequency. The reflected signal is then picked up by a receiver. From this reflected signal, the distance to the reflected object as well as information about the relative speed and the angular position can be calculated. The detection range can be from 2 to 120 meters with 5% accuracy.

3. Lidar Scanner:

   Lidar (Light Detection and Ranging) is an optical measurement device used for the detection and measurement of distances to the objects. This technology has been used in many different areas in physics and astronomy. These sensors have now been installed in automobiles as well. The time from the emission of pulses of light (ultraviolet, infrared) and the reception of their reflection from a potential object is measured and thus the distance to the object is determined. The detection range of the Lidar sensors could have the range of 0 to 130 m. It should be noted, however, that the atmosphere (such as dust particles and water droplets) can reduce the performance of the sensor system.

4. Vision Systems:

   Camera sensors are used in the automobile in order to provide comprehensive information of the road by analyzing the captured images. This can be used later for achieving several functions such as object detection, initiating a collision warning, identifying the lane area, and detecting traffic signs and pedestrians. The detection range of a camera sensor can vary from 5 to 80 m depending on the resolution of the digital camera lens. However, this can be affected by poor weather conditions. Camera sensors have, in contrast to other sensor technologies, the ability of better object classification and a better angular resolution. The technologies use camera systems positioned in the front window that can warn the driver if the vehicle shifts off the lane. This system has been further developed, allowing the vehicle to react and thereby keep the vehicle in the middle of its lane.

Accelerometers and Gyroscopes

These sensors are inertial MEMS (Micro-Electro-Mechanical Systems) which are implemented in order to measure the vehicle dynamics. An accelerometer measures the acceleration and the gyroscope is a device for measuring orientation. Together, these sensors have been used in a number of safety systems, especially in Electronic Stability Control (ESC). Knowing the values of the acceleration and the yaw rate could be essential for systems which stabilize the vehicle in case of skidding by applying differential braking force to the separate wheels, which in consequence leads to reducing the engine torque. Recently, this combination is used Inertial Navigation Systems (INS), in order to support the navigation of vehicles [82].

2.1.2 Positioning Technologies

Navigation objects have been deployed in many application areas such as maritime, aviation and military. This includes finding the absolute position and the orientation of the moved object. Nowadays, navigation is commonly used for vehicles on the road. There are a number of different techniques which are used to locate and navigate the vehicles on the road.

The most commonly used system for outdoor localization is the Global Positioning System (GPS) [30]. The system consists of many satellites, each with already known orbits. A GPS receiver receives many signals from different satellites, then measures the time shift between them and uses the time shifts to calculate the position. Use of the GPS is widespread and has become practical because it is seemingly cheap, but the GPS has its limitations. The accuracy of the absolute position using civilian GPS signals is about 10 meters. This is in general not enough to perform some intelligent drive operations and to assure safety requirements expected from the control algorithms. As an enhancement for GPS, the Differential Global Positioning System (DGPS) is introduced to increase the accuracy of positioning. To implement this system at least one additional reference station is needed in order to estimate the error caused by different characteristics of the atmosphere between the satellites and the receiver. The reference station broadcasts correction data to other GPS receivers, achieving accuracy in centimeter intervals. Although these methods provide a very good way to solve the accuracy problem, the cost factor for building such reference towers is the major drawback of these methods.

Dead-reckoning [51] is also a technique for guessing the vehicle's new position, which depends on information about the exact previous position and the vehicle's dynamics. The later also requires precise information of the vehicle's movement. At each iteration, the traveled distance since the last run, and meanwhile the occurred changes of direction is determined. Thus, the traveled route is added step by step. These changes in the direction can be obtained from a gyroscope or rotation rate recorded sensor. The accuracy of this method depends primarily on the quality of the sensors, for example. The more precisely the distance traveled and current alignment can be determined, the more accurate the location of the position is. Since calculations in this case are always based on the results of the previous calculations, measurement errors are never corrected, but always go with into the next iteration. Therefore the error grows if it is let without correction.

There are other research projects to find the position from exploiting already existing networks. The authors of Lu et al. [62] used a WiFi network using the strength of the received signal as indicator for the position of the vehicles. Fusion techniques have also been used to couple the above mentioned techniques. Sensor fusion means to find the best way of obtaining information from the environment from different sensor sources. In Sasiadek and Wang [82] it was shown that information can be improved by means of sensor fusion of GPS data with the data measured from different vehicle sensors.

2.1.3 Communication Technologies

The more the electronics of the vehicle take over complex functions in the vehicle's tasks, the more complex the interconnectivity between the different electrical control units in vehicles. Many requirements for the communication systems between these units should be achieved such as costs, bandwidth and fault tolerance. Modern bus systems like the Local Interconnect Network (LIN), Controller Area Network (CAN), Media Oriented Systems Transport (MOST) and FlexRay permit transferring data between these units with relatively high data rates. In addition to the communication between the vehicle components, the intelligent vehicle should be able to share/get information from its surrounding environments. In the future the vehicle will be a member of our interconnected world, a world in which entertainment, communication, knowledge and personal contacts anywhere, anytime for any man will be available. In some new vehicles the exchange of information between the vehicle and mobile devices (MP3 player, smart phones, smart watches, car keys, etc.) is already possible through WiFi and Bluetooth adapters.

Communication nowadays is not only limited to the car geometry. Wireless communication between the vehicle with other members of the road is possible forming Vehicular Ad-hoc Networks (VANET). VANET is the technology that allows vehicles to form a network between each other in which each vehicle is a node in a multi-hop network. The terms car-to-X or Car2X stands for wireless communication between vehicles (Car-to-Car, C2C) and for data exchange with the related infrastructure (car-to-Infrastructure, C2I). In the literature this can also be found as Vehicles-to-X (V2X), which includes V2V and V2I. In this thesis we use the terms V2V, V2I and V2X. Having the possibility of V2V communication has the advantage is there is no need for extra infrastructure equipment. However, these kinds of communication need complex protocols.

The communication protocol which is used is the standard IEEE 802.11. Improvements on this protocol for vehicular wireless communication were made and was called the 802.11p standard. US-FCC (Federal Communication Commission) has approved this standard and the communication has a frequency range from 5.85 to 5.925 GHz. The EU Commission has approved the use of some parts of the frequency band at 5.9 GHz in Europe for intelligent transportation systems.

Vehicle Communication Applications

Research of vehicular networks has gotten a lot of the attention in the last years. This is due to the fact that these networks extended the abilities of the car to observe the environment. These communications can also provide an interesting possibility to increase the safety and efficiency of road transport by gathering information from spots out of the range of the vehicle's sensors. Therefore there exist various ranges of applications for vehicular communication systems. These support many advanced driver assistance systems with many safety as well as comfort and economical strategies.

These communications can provide an interesting possibility to increase safety by enabling cooperative collision avoidance. Each car equipped with V2X communication can receive relevant data either from the vehicle or from roadside units. These can be spread out again to other road users or even transmit self-generated information. This creates a chain of information that can also bridge the distance on the road. This can be information of restrictions or problems on the road, for example. Collision warnings and better assessment of accident risk can be thus achieved. At some intersections these kinds of communications can be essential for avoiding an accident.

Some information which is sent via communication is also helpful to increase the flow of the traffic on the road. Traffic monitoring systems can also be in these networks, giving the necessary suggestions to avoid congestions, making the infrastructure facilities a powerful system. Mixing the information from the in-vehicle sensors with road sensor data provides the vehicles with interesting potentials. The use of V2I communication has significant potential to increase road safety at an important part of the road, the intersection. More information about these potentials will be explained in more detail in section 2.3.

There are a number of research papers that give a survey of the vehicular communication systems and their applications. A good survey in the literature on V2V communication systems can be found in Sichitiu and Kihl [90]. The survey presents a number of applications of such networks. Another good reference for a survey of vehicular communication systems in general, concentrating on the enabling technologies, can be found in Papadimitratos et al. [75].

Some examples of projects are:

- VSCC (Vehicle Safety Communications Consortium)
- VII (Vehicle-Infrastructure Integration)
- VICS (Vehicle Information and Communication System)
- CVIS (Cooperative Vehicle-Infrastructure Systems)
- PReVENT

2.1.4 Advanced Driver Assistance Systems

It has been explained previously that a number of intelligent systems are embedded in the vehicles. Examples of such systems are laser sensors for obstacle detection, GPS as a positioning system, and wireless technology systems for sharing information and enabling cooperative driving. These systems have been a part of more intelligent systems which help the driver in different scenarios. The goal of such systems is to collect the data of the surrounding environment and provide the driver with up-to-date information to perform the driving task; taking into consideration the reduction of environmental impacts, the improvement of safety and the reduction of the amount of congestion. Such assistance systems, which can take partial or full control of the car, are called Advanced Driver Assistance Systems (ADAS). A good summary of these systems and the description of their technologies and application can be found in Bishop [18]. Some examples of such systems are: automated parking, collision avoidance and lane-keeping assistance. In addition, there is a number of Automated Vehicle Control System (AVCS) technologies which will be a step towards the main goal of collaborative driving.

An example of systems which allow full automation of the vehicle is seen in Adaptive Cruise Control (ACC). Cruise Control (CC) systems were invented in order to regulate the speed of the car to a certain value. By adding sensors to sense the presence of other vehicles (lidar, radar) in the way of the vehicle, the ACC system was introduced. The vehicle drives in the desired speed as in CC systems, but once another vehicle is detected the velocity must be adjusted to the vehicle speed and the distance to the front vehicle in order to avoid collision.

Highways are an important part of the road. The idea of reducing congestion on highways by introducing automated highway concepts has emerged in the middle of the

1970s. With the development in many areas of vehicle sensor technologies this concept has been further investigated and Automated Highway Systems (AHS) proposed as a solution to cope with traffic problems. The concept of AHS has also defined a new relationship between vehicles and the highway infrastructure. The concept was to assign part of the roadway and let equipped vehicles operate under completely automatic control, providing safer and more convenient travel. To do this the existing highway infrastructure is also changed in such a way that the driving vehicle knows its position. These vehicles are then driving in platoon formation.

2.2 Platooning Systems

By eliminating the human reaction time, the necessary safety distance between cars can be much smaller, and thus achieve a higher traffic density. A Platoon is a string of electronically connected vehicles driving together on the roads keeping in close spacing. The platoon is then an advance system of the ACC, where the communication between the vehicles has been enabled and the vehicles can exchange sensor data between them, allowing the vehicles to drive autonomously. There is always one vehicle driven manually, called the leader, whereas the other vehicles are the controlled autonomously and called followers. Driving in platoon formation needs a driver even for the follower vehicles which drive autonomously. For legal reasons in most of these experiments the driver must be observing the road and the drive, but this apparently reduces the need of the full attention of the driver.

Platoon systems were investigated and were a part of the AHS research. They make changes on the road in the sense that the vehicle could find the location and keep in the lane. This could be done either by embedded sense magnets in the road or by a vision system on the vehicle which recognizes the painted marker stripes on the road. The idea is to form automated highways where the vehicles travel with small-space distances controlled through computer systems.

The California PATH project (Partners for Advanced Transit and Highways) addressed many projects related to transportation safety research since 1988 and it was one of the first leaders in the area of platoon systems, where they demonstrated automated vehicles driving in platoon formation. The PATH partners initiated a number of important projects in this area. During this project different control structures were proposed and tested by simulation. Still, few tests were performed on highways with real vehicles. In 1994 they demonstrated four automated vehicles driving in platoon on the San Diego highway in California. Shortly after three years they made a demonstration with a platoon of eight cars on the same highway. Japanese researchers launched a similar project. The Advanced Cruise-Assist Highway Systems Research Association (AHSRA) is also considered an important project in which Japanese developers tested different platooning scenarios. These were followed by some European projects like PROMOTE-CHAUFFEUR (1996-2004) and KONVOI (2006-2009). Some real tests were reported in the projects by [69], [33], [27] and [78]. In [64], a detailed description of the hardware implementation on real heavy-duty trucks is given. Two heavily loaded trucks are tested

(a) KONVOI: Testing an automated platoon of trucks on German highways (2009) [78]

(b) PATH: Testing an eight car automated platoon on an Automated Highway System(1997)

Figure 2.1: Real experiments with platoon driving

for constant inter-vehicle spacing. Redundant sensors are used to provide an accurate and reliable distance measurement via wireless inter-vehicle communication. In [27] the trucks were equipped with GPS modules. A control structure was tested on a single truck with built-in sensors including a fuel rate sensor, a tachometer for engine speed and a manifold pressure sensor (see [63]). A brake pressure sensor for each wheel and two wheel speedometer sensors - one for low speed and one for high speed - were used too. In this test a reference speed trajectory is used as the virtual front vehicle. In [48], the hardware and the software of an autonomous smart car were proposed. A camera and ultrasonic sensors were used for lane navigation. One latest project in this area is the European project SARTRE which will consider issues in order to allow platoons to operate on public roadways. In this project engineers are trying to design concepts of how vehicles can join or leave the platoon [2].

2.2.1 Platoon Architecture

Research on automatic platooning started a long time ago (see the works Sheikholeslam and Desoer [87] and Swaroop and Hedrick [95]). Mainly the problem of platooning was developed using the architecture introduced in Lygeros et al. [65]. Efficient control of vehicles driving in platoon was introduced by Shladover et al. [88]. Another coordination model using a decentralized coordination model has been proposed by Howell et al. [46]. The control of cooperating systems opens up new and promising approach opportunities in the automotive and robotics fields. Cooperative driving of a platoon of vehicles involves exchanging information via a communication network, which is important. Depending on the proposed control law, this information could be the distance between the vehicles, the deviation between their velocities or accelerations, or just their values. It has been proven in the literature that networked interconnection is needed to ensure string stability, which requires the attenuation of internal deviations as they propagate through the platoon. Swaroop and Hedrick [94] showed that there is no need for control data to be transmitted from the leader to other platoon members to ensure stability. In case of no communication failure the data measured from vehicles sensors are enough to be exchanged as explained by Liu et al. [59].

Multiple possibilities exist for the design of the platoon's communication structure. These structures will be presented shortly:

- Full communication:

  In this scenario all vehicles communicate with each other and each vehicle broadcasts its information to all participants of the platoon.

- Information from the preceding vehicle:

  Each vehicle accesses information only from the vehicle ahead. In this structure knowing this information improves the reaction rate and can reduce the error deviation of the system.

- Information from both the preceding and the lead vehicle:

  Since the leader of the platoon is the source of the input to the platoon system, knowing the status of the leader is combined with the preceding vehicle. Therefore the information of changes in the system is transmitted quickly and any disturbances the platoon can be quickly compensated.

- Information from all the preceding vehicles:

  This is similar to the structure above, but here information from all the preceding vehicles, including the leader, are known.

2.2.2 Efficiency and Safety

The transport system is suffering from traffic flow and safety on the roads in recent years, the road density has increased rapidly, and the infrastructure of highways in most major countries is congested due to the increasing number of vehicles and the increasing of freight transportation. Crashes on the road are the number two reason for causing deaths worldwide. The National Highway Traffic Safety Systems (NHTSA) provided statistics that 80-90% of all accidents are caused by driver error. Furthermore, the growth of transport activity affects the environment directly; owing to a significant share of the CO_2 emission in the environment, which consequently leads to the global warming phenomena. Therefore an optimum utilization of motorway capacities should be sought. The automatic driving in platoon formation, maintaining short but at the same time safe distances between different vehicles, could be a potential solution for these problems.

Efficiency

Using the platoon system can increase the road capacity, since platoons drive in small inter-vehicle distances. Consequently, this also has an effect on energy consumption reduction, especially for trucks, due to the reduction in the aerodynamic drag. Shladover

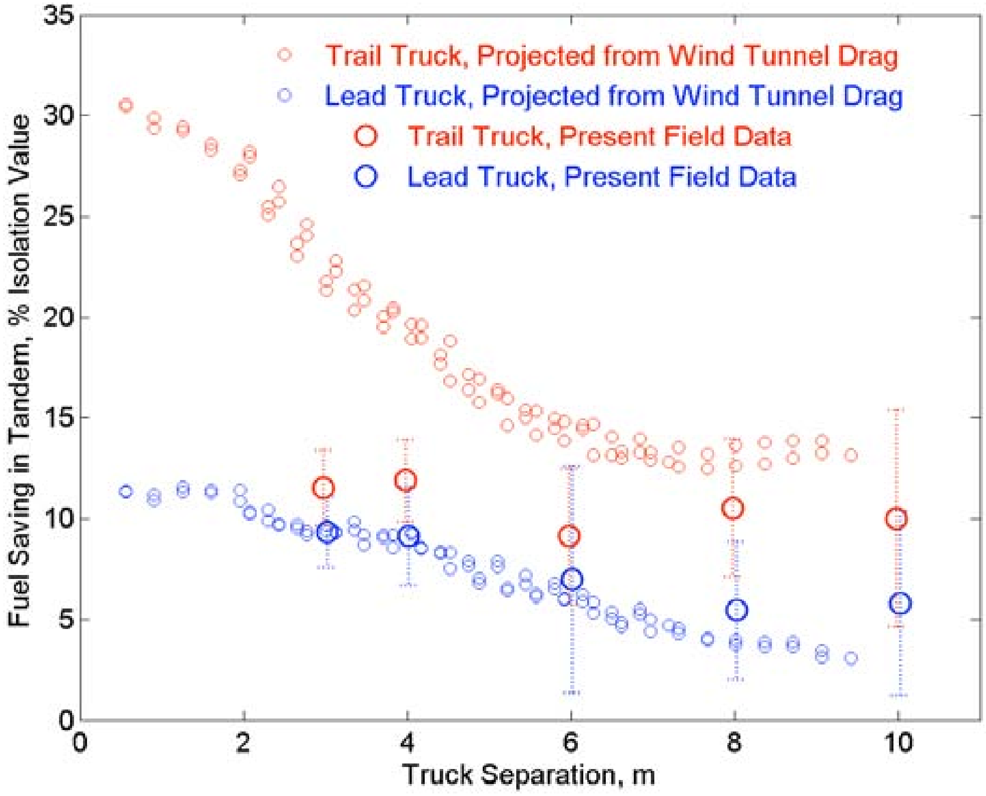


Figure 2.2: Fuel savings for two trucks driving: Comparing the results from field measurements and the wind tunnel tests [22]

with one of his earliest works [89] estimated that the capacity of the highway could be doubled by using the longitudinal control of platoon. Some results from the European project CHAUFFEUR reports that circa 20% fuel savings when driving in this formation is due to the reduction in the aerodynamic drag, not only for the follower vehicles but also for the leader.

In their review to the application of ITS for energy savings, Tsugawa and Kato [98] mentioned simulations, which is used in the Energy ITS project (Initiated in Japan in 2008) that investigated the effect of platoon driving on the aerodynamic drag. The results shows that when three trucks drive at $80km/h$ with the gap of $4m$, the coefficient of drag (CD) values decrease by about 50% on the middle, and more than 20% on the lead and the end. Consequently, results showed about a 15% reduction in fuel consumption.

In another work of Tsugawa et al. [99], the reference showed that by driving on a highway with a platoon of three fully automated trucks driving at $80km/h$ with the gap of $10m$, fuel can be saved by about 14%. They also showed that CO_2 reduction along expressways will be 2.1% and that this reduction can be increased by making the distance between vehicles smaller. The reference [22] shows the result of the fuel consumption of two trucks driving after each other. Different experiments were performed changing the inter-vehicle space between the trucks. Figure 2.2 shows the fuel saving results.

In addition, unnecessary braking due the wrong assessment of the driver could be avoided.

Safety

Traffic congestion increases the fatalities and injuries on the road. Therefore avoiding such congestion by driving in short distances, such as in platoon formation, can avoid such effects. The reaction time of the machine in the platooning system is much faster than that of humans. Thus, safety on the roads could also be increased by eliminating the human factor causing the accidents and the unnecessary braking. However, safety in automatic platoon driving should be investigated more in order to take into consideration all possible sources of risks. Communication between platoon vehicles increase the performance of the system, but the errors that could happen to the communication network could have a very dangerous effect on people's lives.

2.3 Intelligent Intersections

There has been a lot of effort, made from governments and from automobile manufacturers, in order to reduce the accident rate in road traffic. Although the statistics show a noticeable reduction of the amount of accidents, intersections continue to be classified as an accident-related hot spot. Almost 50% of accidents involving personal injury occur at intersections. Therefore, there is an urgent need to introduce systems and methods to assist the driver in emergency maneuvers.

There are different reasons for accidents at intersections. One typical reason is that the driver's sight was blocked by buildings or trees. Some collisions might occur because the driver's awareness of the intersection was too late to avoid collision, because he was not paying enough attention, or because he was distracted. In addition, misjudgments from the driver's side such as underestimating his speed or the distance to the stop sign have frequently caused accidents at intersections. Thus, improvement of the intersection safety can be divided into the ones concerning the road and the others concerning the driver. Some improvement on the road could be accomplished by giving the driver enough time to see the intersection, thereby giving him the possibility to estimate possible dangerous scenarios better.

Most of the accidents are not the fault of the road infrastructure itself, but rather that of human drivers. The latest developments in vehicle sensor technologies, appropriate facilities and the transport infrastructure coupled with communication offer a great potential to improve the driving at intersections. Although on-board sensors for intelligent vehicles could provide the driver with important information, a complete coverage of the intersection is not possible. By introducing wireless communication via Vehicle-to-Infrastructure (V2I), the driver and the intelligent vehicle could have important information about the driver's blind areas where potentially dangerous situations could occur. Vehicle-to-Vehicle (V2V) communication could in some cases provide the same function.

This technology has profound potential to increase road safety by informing the driver through direct communication between the vehicle and traffic lights about potential accident risks, and about construction site with changes of intersection characteristics. It also estimates and warns the driver of red-light violations and risky turning maneuvers and to inform of approaching vehicles with blue lights that want to cross the intersection quickly.

Communication with intersections with traffic lights could add another important potential role. Intelligent traffic light control and traffic optimization could also be achieved to support smooth driving. By knowing whether and at what speed the vehicle could pass the next traffic light with its green phase, the speed of the driver could be recalculated in order to reach the intersection with an optimum speed and minimal changes of acceleration and braking, thus making the drive quiet, safe and fuel efficient.

This chapter describes the intersection as a very important area on the road. It summarizes the latest efforts done in order to make crossing the intersection safer and more efficient.

2.3.1 Characteristics of Intelligent Intersections

Several technical solutions and a number of intersection assistant systems were developed recently. Based on the functionality of these systems three types of intersection crossing assistants are considered (see the work of Benmimoun et al. [17]):

- Informational Assistance:

  The system passes only the raw collected information to the driver. The driver must independently assess the situation and respond.

- Assistance Warning:

  The system collects information about the environment and assesses the traffic situation. The driver given a warning is only in the event of a critical situation.

- Intervening Assistance:

  The system evaluates the driving situation and the vehicle brakes automatically in the event of an impending collision.

In order to get a full picture of the intersection, intersection observers with a central unit and a communication module can be installed. This can also be connected to roadside sensors or to camera vision systems. For roads with many road users, this global knowledge of intersection situations can lead to an efficient solution. Using such intersection management units provides another advantage in which every road user needs to communicate with just one instance only. Compared with pairwise communication between all vehicles, this is a major simplification. The disadvantage of such system is that an additional unit must be installed at each intersection.

Control Management at Intelligent Intersections

Based on the method of how the number of necessary information is collected such as the position, velocity and acceleration, one can differentiate between several methods:

- Infrastructure measuring:

  Here the intersections are provided with developed sensors such as vision systems and/or road sensors which could measure the speed and the acceleration of the arriving vehicles. These can deliver necessary information to all vehicles approaching the intersection. Such measuring can have a significant advantage when not all vehicles on the road are intelligent and can communicate. The accuracy of these systems can however be low, depending on the quality of the sensors.

- In-vehicle measuring:

  Using the in-vehicle sensors to get the real-time position and speed. This has the advantage that there is no need to install detectors at the intersection area. But this system would be effective only if all vehicles can share their data through wireless communication.

- Vehicle-Infrastructure measuring:

  In these scenarios the data from both above methods are combined in order to reach the best performance.

In order to avoid collision and to regulate the movement at intelligent intersections, a differentiation can be made between two control approaches:

1. Centralized control:

   An intelligent intersection supervisor which is characterized by a central unit at the intersection collects information on the situation at the intersection and on incoming vehicles, and with this knowledge determines the behavior and how the vehicles enter the intersection. The V2I communication is then used to warn the driver and provide recommendations of the best behavior.

2. Decentralized control:

   In decentralized management no additional information from the infrastructure is available. Since not all intersections can be provided with such an intersection supervisor, the vehicles should manage to operate only with the help of the V2V communication. The control objective is achieved by adjusting the behavior of each vehicle individually.

2.3.2 Safety at Intersections

In order to avoid a collision on the roads, a number of collision avoidance systems have been developed in order to warn the driver or act in the case of a collision risk. This includes avoiding collisions for static and moving objects. Some of these systems can show the distance ahead of the car when the distance is so small. Using communication technologies open a new method for enhancing the traffic safety. Some developed systems based on a Dedicated Short Range Communication (DSRC) protocol can be used to initiate a warning message when emergency events occur [19].

Because of the high risk of collisions, collision avoidance at intersections is important. Traditional collision avoidance systems are not so effective at intersections, though, since their design and functionality are restricted for one vehicle. Therefore some kind of cooperative work between the different traffic partners should be achieved.

Increasing the safety at intersections has gotten a lot of interest and a lot of projects were initiated for this purpose. The key idea of these projects is to use infrastructure sensors along with in-vehicle sensors and the wireless communication technology to improve the safety factor at the intersection. Important projects in this area are: CICAS (USA) [1] and DSSS (Japan). The European project INTERSAFE, which was completed in January 2008, investigates dangerous scenarios at the intersection by enabling the development of ADAS to take into account crossing the intersection [35]. One of the goals in this project is the development of an intersection driver warning system based on bi-directional communication with traffic lights. The positions and the possible trajectories of the vehicle were done by on-board vehicle sensors and road sensors. Some promising results are shown in the reference [34]. The project INTERSAFE-2 builds on these results and develops the concept further towards more practical use. It concentrates on cooperative communication between the vehicles themselves as well as between vehicles and the infrastructure (traffic lights, sensors). Here it is also studied how the vehicles can act in case of danger by slowing down automatically.

As explained in Subsection 2.3.1, to avoid collisions mainly two types of systems which provide collision avoidance assistance in the intersection can be used: centralized and decentralized control. Since a lot of new vehicles could be provided more easily and faster with the necessarily protocols and technology than systems which entirely depend on infrastructure systems, decentralized control could be an advantage here. However, in central control a better view about the whole intersection can be provided. The combination of these systems is possible and can provide a better performance. Thus the vehicles exchange their speeds and locations to the intersection and then the intersection is able to use this information in addition to the data extracted from the infrastructure environment for better collision estimations. In addition, direct communication between the vehicles is possible.

2.3.3 Traffic Light Optimization

Traffic lights plays an important role in the transportation network and can be an effective way to prevent collisions by determining the right of way at the intersection. Still though, when this instrument is not activated optimally, it can cause a lot of unnecessary delays and time spent on the roads, permanent stops and a lot of anger. This consequently leads to a significant effect on the economy and on the environment. Therefore, reducing wasted energy and emissions coming the engines due to frequent stops at intersections became an important field of research in the last decades.

Synchronization and good planning of traffic signal controls can have many advantages. A wise traffic signal timing management is therefore beneficial. The goal of an optimized traffic signal is to reduce traffic delay, fuel consumption, fuel emission and intersection crashes. In case of roads with successive intersections along a road, the focus here is to shorten the travel time of the vehicles and to achieve that the vehicles reach the green waves in the right time. The paper in [74] discussed different signal controlling methods with the goal to reduce the delay time and to minimize the emission of CO_2. Other examples of such studies could be found in the work of Nishiuchi and Yoshii [73] where the authors compared different signal control policies.

Allowing the vehicle to communicate with the traffic light has the potential of saving fuel and reducing traffic delay. This can be done in two main topics. One option is that the traffic light changes its sequence in order to manage more system throughput such as turn to green right now when no other cars are in the road. The other option is to deliver some messages to the vehicle itself to permit, forbid or regulate passing the intersection. Information about the current traffic light status, the periods of the green, yellow and red phases, are also provided. By analyzing these kinds of information the speed to reach the traffic light without stopping at its red phase can be calculated or a stop is made if necessary. The driver should have this information early enough to slow down gradually. This increases safety as well as comfort since the driver has an idea of how approaching the intersection would be.

Wireless communication is not the only way to obtain information from the traffic light to the vehicle. The paper by Wada et al. [102] has introduced a unique way to do that. They used the LEDs on the traffic lights themselves. Using a high-speed camera, the turning on and off od individual LEDs can be recorded. They could achieve a transmission speed of 2.78kbps by using 192 LEDs. The disadvantage of this system is that communication can take place only on one side. The traffic light system has been given no opportunity to include any information from the vehicles.

A number of questions still has to be answered and further investigated. When is the right time for the driver to get information from traffic lights? What is the right interval of time in which the information can be sent repeatedly? The work of Tielert et al. [96] explored the effects of the communication radius and the appropriate choice of the gear on the fuel consumption. It was emphasized that efficient gear selection could save up to 20% of the fuel consumption. The work in Mandava et al. [67] showed that with intelligent planning of speed changes fuel savings up to 12-14% is possible. Further research to investigate the psychological factors and how the driver could deal with the new system has been done. These test how the driver can follow the instruction given to him from the system.

For automated driver assistance systems, knowing the traffic light information could be a significant advantage. Speed adaptation for a cruise control system was discussed in the work of Asadi and Vahidi [12]. There, the authors tried to achieve an optimum velocity and trajectories for the vehicle and how to plan desired velocity around the red lights, and achieving a desired velocity used the information. Since sharp accelerations/decelerations must be avoided because they directly affect the fuel consumption, Sassi et al. [83] developed an algorithm to provide dynamic speed advice to the driver to increase the probability of passing through a green light without stopping. The authors Saust et al. [84] introduced an architecture of a cooperative system adjusting the signal control and the vehicles' driving strategy. Automated lane changes were discussed to reduce stops at traffic lights.

3 An Experimental Testbed for Cooperative Vehicle Platoon

Today, an increasing number of vehicles are provided with many sensors and with over 100 microcontrollers. This number indicates clearly that vehicles do not have purely mechanical functions, but are strongly supported by electrical parts and their software. In both many universities and the industry there are many research programs which investigate the use of software in modern smart vehicles. Software can be used, for example, in order to improve safety, energy efficiency and to support advanced driver assistance, infotainment and communication systems. The development of autonomous vehicles and testing the advance logic controls of a vehicle's behavior has gained a significant share in these research programs.

On the one hand, hardware testing helps detect hidden problems, which cannot be detected during simulations, such as noises and time delays caused by hardware components. On the other hand, equipment of real trucks with the adequate hardware and recurrent tests on highways or special stretches are costly and often legally problematic. Using scaled vehicles for testing safety and traffic systems could provide a solution for this problem.

The chapter begins with some research projects using scaled vehicles. Afterwards, a scaled vehicle platoon platform which was developed for the testing of the autonomous platoon behavior will be introduced. This chapter also shows some of the different control methods which were used in this platform.

External Contributions

Subsection 3.3.1 presents methods for controlling the platform vehicles. This section presents two controllers regulating the distance between the vehicles in order to stabilize the longitudinal driving of the vehicles of the platform. These controllers were designed by Martin Chavez and by Jan Maschuw. The algorithm for compensation of nonlinearities in the scaled vehicle was developed by Martin Chavez. The related publications ([DCGBM+10] and [MDAK11]) explain the design of these control methods in more detail. The results of testing these controllers will be presented in Subsection 3.3.2. My contribution to these experiments was the design of a software tool for evaluating and presenting the results and helping to tune the controller's parameters while conducting the experiments.

3.1 Using Scaled Vehicles in ITS: State of the Art

In the phases of developing autonomous vehicles there is an increasing demand for conducting many experiments in order to assure the functionality of the autonomous systems, which opens opportunities for minimizing costs and risks. Making an evaluation of such systems by driving on roads is not easy, since there is a lack of facilities on which such experiments could take place. Performing driving tests on the roads also requires consents and approval from governments because the testing is potentially dangerous. Although using driving simulations tools can significantly contribute in the development process, these cannot replace real experimental tests. Developing scale-sized vehicles as test environments could be helpful in the development methodology of engineering problems, particularly in the development of future autonomous vehicle systems.

Scaled-sized environments were the topic of study of different engineering applications. In marine applications and the aerospace industry, scaled models were already used successfully in order to prove some characteristics of the full-scale system. Up until now, wind tunnel testing is still used in order to investigate the dynamics of experimental planes. Therefore using scaled ground vehicles in the process of developing new technologies gives new opportunities for more efficient experimental procedures. It allows the evaluation of the developed systems under different conditions. Some advantages of making tests on scaled models can be summarized as follows:

- Reducing the budgets in the testing process. This is mainly because of the high cost of equipping a full-scale autonomous vehicle.
- No risk factors and no restriction of government approval of experiment. The experiments are usually performed in the laboratory and do not endanger anybody. In addition, this makes conducting experiments always possible and there is no waiting or delays.
- Fast changes of simulated scenarios: The parameters of the controller or even the task could quickly be changed online without worrying about the other road members, which is the case in full-scale road testing.

In the literature there are a number of publications which include the current use of scaled experimental vehicles in the vehicle development process. The University of Urbana-Champaign and Penn State University have been two of the pioneers in using scaled vehicles for testing different vehicle dynamics control systems. For their research purposes they elaborate scaled vehicles in an 1:10th scale with an electric single-wheel drive and made many tests on the Illinois Roadway Simulator (IRS)[21]. The road simulator is specially manufactured test equipment consisting of treadmills and conveyors.

The focus of that research was not only on doing experiments to test the vehicle dynamics control system, they were also interested in finding methods to design scale model vehicles where the dynamic parameters of the scale model can somehow be taken out into the full-scale vehicle. Examples of these works can be found in the works of Brennan and Alleyne [20] in which the concept of scaling and similarity theories using dimensional analysis methods were applied. A dimensionless model of vehicle dynamics was introduced in order to design a robust controller which later can directly be implemented on a full-size vehicle. However, other factors should be considered and are very important: namely, real-world constraints such as latency, hazardous events, and errors caused by processors.

Examples of other publications where they concentrated on describing the dynamics of scaled vehicles in order to find methods for controlling the movement of these kinds of vehicles can be seen in the works of Hoblet et al. [45] and Travis et al. [97]. Examples of laboratories which developed scaled vehicles to serve as as a testbed for the cost-effective development of vehicle dynamics control systems can be found in the works of Verma et al. [101] and Huang et al. [47]).

A number of researchers have focused their work on conducting experiments on how to implement some of the intelligent vehicle systems on scaled vehicles. Longoria et al. [60] is one example use of scaled vehicles in the design process of autonomous vehicle prototyping. In this work the authors have selected a 1:5th scale radio-controlled (RC) cars for testing anti-lock braking systems (ABS). The authors in Katzourakis et al. [52] also showed how they equipped a 1:5th scale RC car with an electronic brake system for the implementation of an electronic stability control system focusing on a more economical solution. An approach to utilize a modified 1:10th scale RC car to investigate the influence of rollover propensity was done by Travis et al. [97]. In order to test number of collision avoidance strategies, in his work Ferrara [31] used scaled vehicles for the investigation. He used a 1:10 RC car to analyze and test an approach of an automatic pre-crash collision avoidance system.

By equipping their scaled vehicles with communication modules, some other researchers extended the test application area. For testing some applications of using V2X communication the authors Hecker et al. [40] have developed a scaled testbed vehicle to test these applications quickly and efficiently. Scaled vehicles were also applied in order to investigate the cooperative functions between many vehicles. Sakaguchi et al. [81] used wheeled mobile robots which use inter-vehicle communication. The work has particularly focused on data transmission protocol for the merging control of vehicles during lane changing on highways.

3.2 Cooperative Vehicle Platform Description

As described in the previous section, scaled vehicles have found their way in many tests and applications. In order to investigate the behavior of the platoon under different control methods and different real-world constraints, we developed a test platform consisting of four 1:14-scaled trucks driving in platoon formation. The platoon consists of three identical driverless automated trucks following a manually driven one, as can be seen in figure 3.1.

Model trucks from Tamia with a scale of 1:14 were chosen. Each vehicle is equipped with several sensors, actuators and controllers to interact with the environment in order

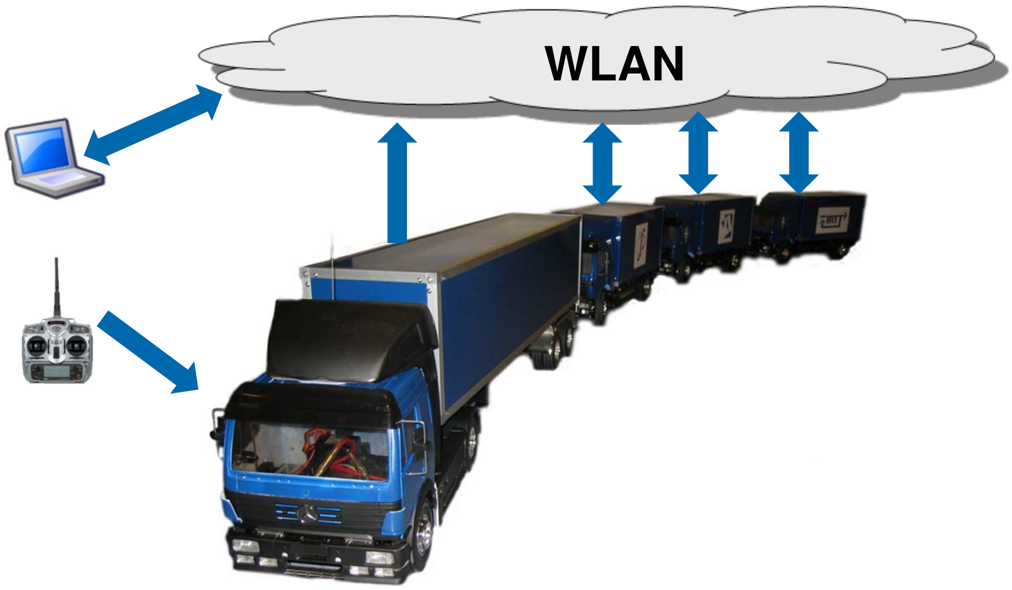


Figure 3.1: Platoon architecture

to achieve proper autonomous driving. One goal of this platform is to observe the effects of communication on the controller of the platoon vehicles and study their influence on safety. This section describes the hardware and the software architecture of the platoon platform including the communication network.

3.2.1 Hardware Architecture

Each vehicle is equipped with sensors to measure vehicle-specific variables such as inter-vehicle spacing, the angular deviation between the given vehicle and the one ahead, the velocity and acceleration. In addition, a wireless radio communication system is integrated, enabling each vehicle to share its real-time data with other vehicles in the platoon. Two microcontroller systems in each vehicle process the information and control the movement of the vehicle. Figure 3.2 displays the architecture of the hardware implementation and how the different elements are connected to each other. The position of the different electrical elements and modules in the model truck can be seen in Figure 3.3.

The rest of this subsection provides more description of the different hardware elements, implemented in each vehicle.

Microcontrollers of the Platform

Each vehicle is equipped with two microcontroller systems:

- Sensor data platform
- Controller platform

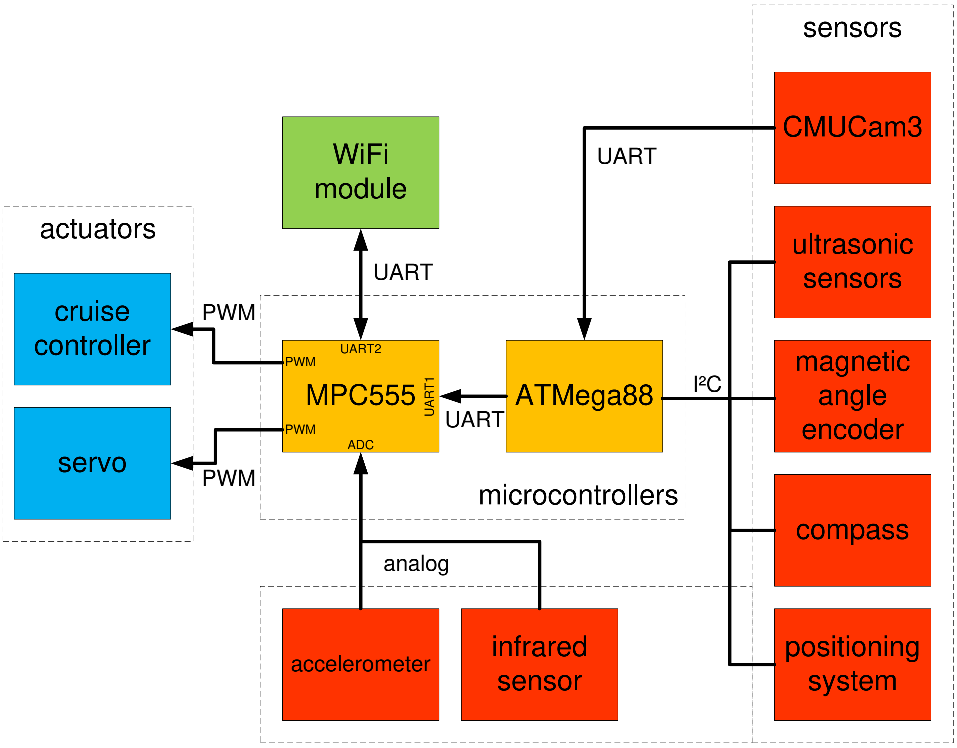


Figure 3.2: Hardware architecture of each truck

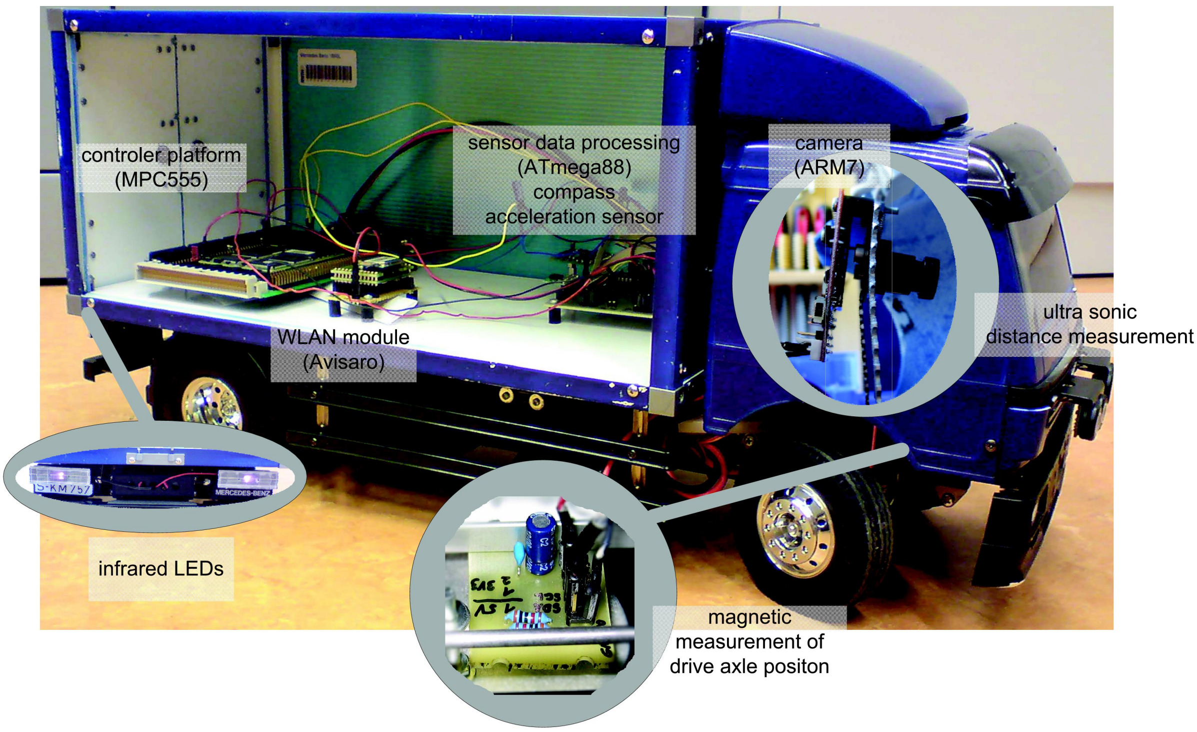


Figure 3.3: The position of hardware modules on each truck

The controller platform a phyCORE®-MPC555 Rapid Development Kit, is a high-performance system on module with a 32-bit PowerPC MPC555 microcontroller with a floating point unit that is designed to support complex applications, especially in the automotive field. The sensor data platform is a self-designed printed circuit board with an ATmega88 microcntroller from ATmega.

The sensor data platform is responsible for capturing the necessary information from different sensor modules and transmitting them to the controller platform, which gives the necessary values in order to steer the trucks automatically. Both the connection between the sensors and the sensor data platform and the connection between the controller platform and the sensor data platform were realized by a serial connection protocols, the Inter-Integrated Circuit Protocol (I^2C) and Universal Asynchronous Receiver/Transmitter ($UART$), respectively. The collected data from the sensor data platform is sent regularly (every $50ms$). The controller platform generates the Pulse-Width Modulation(PWM) signals required by the internal vehicle controllers. These signals are responsible for driving the motor block of the trucks (the cruise controller and the servo).

Sensing Information

The model trucks are equipped with various types of sensors to capture the necessary information for the controller.

- Distance measuring:

  Different ways for computing the spacing to the vehicle ahead were applied. In front of each vehicle we placed one infrared-based distance measuring sensor (GP2Y0A02YK) and two ultrasound-based sensors (SRF10). The reason for using many sensors with different measuring-based methods is the possibility of data fusion analysis for distance acquisition. Ultrasound sensors have actually been favorable for this application where no wide-range measurements are required. However, the infrared-based distance measuring sensors have been also used in some of the platooning experiments for testing the indoor positioning system, which will be introduced later in this thesis in Section 5.2.

- Speed measurement:

  This measurement is carried out by a contactless magnetic angle encoder (AS5046). A small magnet is attached to the drive axle of a truck. By doing that the magnetic angle encoder can now measure the angle of this magnet while it rotates with the drive axle. This gives the angular speed ω. The speed v of the truck can now be computed with $v = \omega r$ where r is the radius of the wheels of the truck.

- Acceleration measurement:

  An accelerometer (MMA7260QT) placed on the vehicle allows it to determine the lateral and longitudinal components of the acceleration.

- Direction measurement:

  A compass module (CMPS03) was installed to help determine the relative angle of the vehicle ahead. This is useful to support lateral control of the platoon.

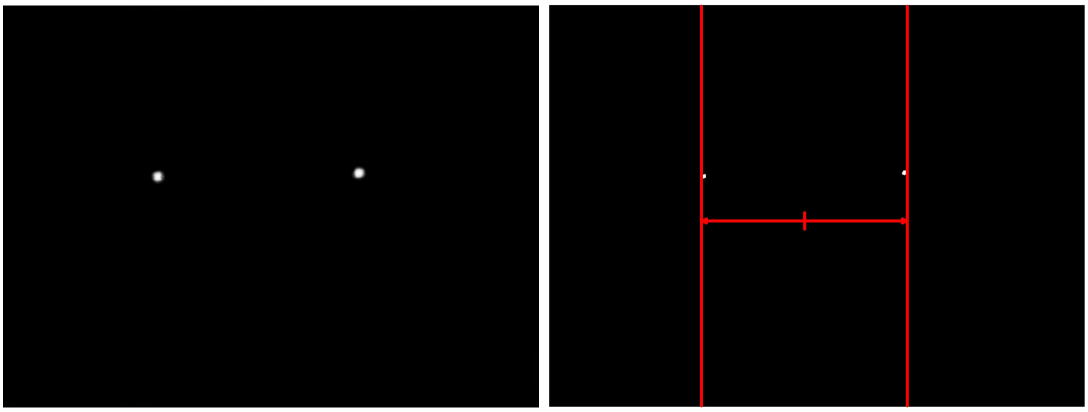
Image Processing

All of the trucks which follow the leader, need information about changes in direction of the truck ahead in order to be able of drive in the platoon. To test the lateral control of the platoon, we made the tracking using a camera module placed in front of each truck. The camera module consists of:

Figure 3.4: IR light in blue channel

- A camera with resolution (352x288) RGB color
- A main processor, Philips LPC2106(60 MHz), and
- A basic image manipulation library including functions such as image clipping, threshold and convolution functions.

In order to process the image of the whole truck, complex image processing techniques should be used, which has been shown to be time as well as memory consuming for the camera processor. Taking into consideration that the delayed information could have a serious affect on the control of the platoon, another approach has been used. One method consists of tracking infrared points. To realize this, two infrared LEDs were



(a) View with an IR filter (b) Estimated middle of the truck

Figure 3.5: Camera view with an IR filter with the estimated middle of the truck: Example 1

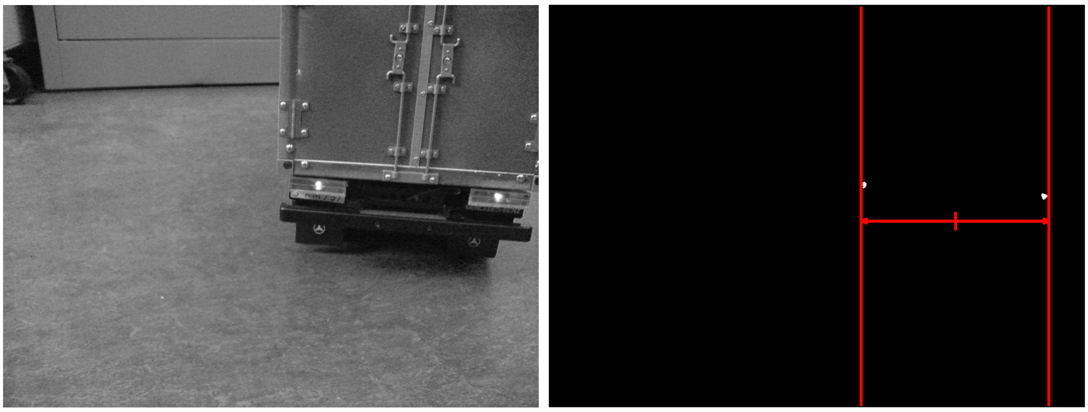
mounted on the rear side of each vehicle and the image processing now concentrates on detecting them. Figure 3.4 shows an image seen by the blue channel of a camera.

For simplifying the process, an infrared filter was put on the cameras. This filter consists of the black, fully exposed part of a negative film. Figure 3.5(a) shows the image after applying such a filter to the camera. A brightness threshold was applied to avoid confusion with other low intensity infrared lights by taking the brightest points in the image, causing the greyscale image from before to become really a black image with two bright white points. Assuming now only the IR LEDs cause a pixel set in this bitmap, it is very easy to determine the position of the truck in front (meaning the angle to the truck). By calculating the average of the smallest x-value and the highest x-value of a set pixel the middle of the truck is estimated like shown in figure 3.5(b).

The original camera image has a resolution of 352 x 244 pixels. To speed up the processing, the image is down sampled by the camera logic by a factor 2 in x-axis and factor 4 in y-axis. Also some lines of the picture in the top and bottom of the image are ignored for processing. The resulting image for processing has a size of 176 x 25 pixels. Using this improvement, the angle of the truck in the front can be calculated each 50 ms. Figure 3.6 shows another example, where the truck in the front moved to the right side.

To estimate the relative angular position $\Delta\theta$ between two vehicles, the distance δ_m between the center of the image and the middle of the two brightest pixels was computed. Depending on the value of δ_m three possible scenarios arise in the controller algorithm to follow the car ahead:

- $\delta_m = 0$, $\Delta\theta_m = 0$ there is no need to move the truck in the lateral direction.
- $\delta_m > 0$, $\Delta\theta_m > 0$ truck should move to the right to correct the direction.
- $\delta_m < 0$, $\Delta\theta_m < 0$ truck should move to the left to correct the direction.



(a) IR light in blue channel (b) Estimated middle of the truck

Figure 3.6: Camera view with an IR filter with the estimated middle of the truck: Example 2

3.2.2 Software Implementation

One of the goals of this platform is to help control engineers to develop different control algorithms. Many developers in the field of control systems use the environment of MATLAB/Simulink for modeling and simulating their mathematical algorithms. As a result, the software for the control of platform was also developed in MATLAB/Simulink, thus giving the developers the opportunity to deal with a familiar programming environment. Therefore many Simulink blocks have been developed for the platooning software which provide an interface to the platoon hardware.

The Target Support Package FM5 tool (no longer available alone and now part of the Embedded Coder tool) is included in the platform software. This tool provides device driver blocks which give access to the on-chip resources. Some drivers of this tool are:

- Pulse-width Modulation (PWM) generation
- Analog-to-digital converter
- Digital input and output
- Frequency and Pulse-width measurement
- Transmitting or receiving Controller Area Network (CAN) messages via the MPC5xx TouCAN modules
- Serial transmitting and receiving

We used some blocks of this tool and programmed extra blocks to interface the platoon hardware. Figure 3.7 shows the typical feedback control loop, with the different software blocks that needed to be implemented. In this structure, the controller in each vehicle gets the information from its sensor data and from the sensor data from other vehicles (via the WiFi). Based on the control algorithm, the controller gives the appropriate signals to the vehicle actuators to maintain a safe distance to the vehicle ahead.

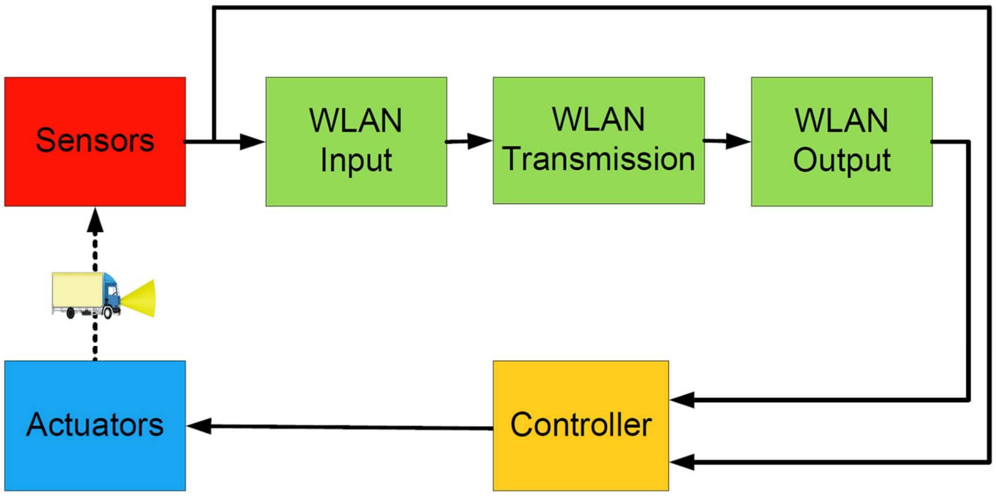


Figure 3.7: Control loop

To achieve this structure we implemented a number of necessary blocks. For example, the sensor block interprets the raw sensor data. The network structure of the platoon is managed by communication blocks which organize the sending and receiving of data between the WiFi modules. Another example is the selector block which collects all received data and separates it according to the correspondence to a certain truck. The controller blocks were mainly implemented by the institute "Institut für Regelungstechnik". These blocks have been discretized in order to be programmable in the MPC555 platform. This was done by using the discretization tool of Simulink.

Because of the hybrid nature of the system, an event-triggered switch between different controllers were also taken into account and was integrated. The software were structured in order to enable the switch between two different controllers in the same experiment phase, which helps testing different controller modes without significant effort.

Code generation is done automatically, guaranteeing a fast and small code; an important factor in embedded applications. The Target Support Package transfers the code generated from the Real-Time Workshop Embedded Coder in Simulink onto the Freescale MPC555 microcontroller by compiling, downloading, running and debugging the generated code on the hardware target. Figure 3.8 shows the process steps from the Simulink model to the microcontroller code.

Monitoring and Remote Controlling via PC

To monitor the course of the experiment and to collect information from different vehicles, a Graphical User Interface (GUI), programmed in MATLAB, was developed. This GUI collects the data from all trucks and visualizes them for the user without additional

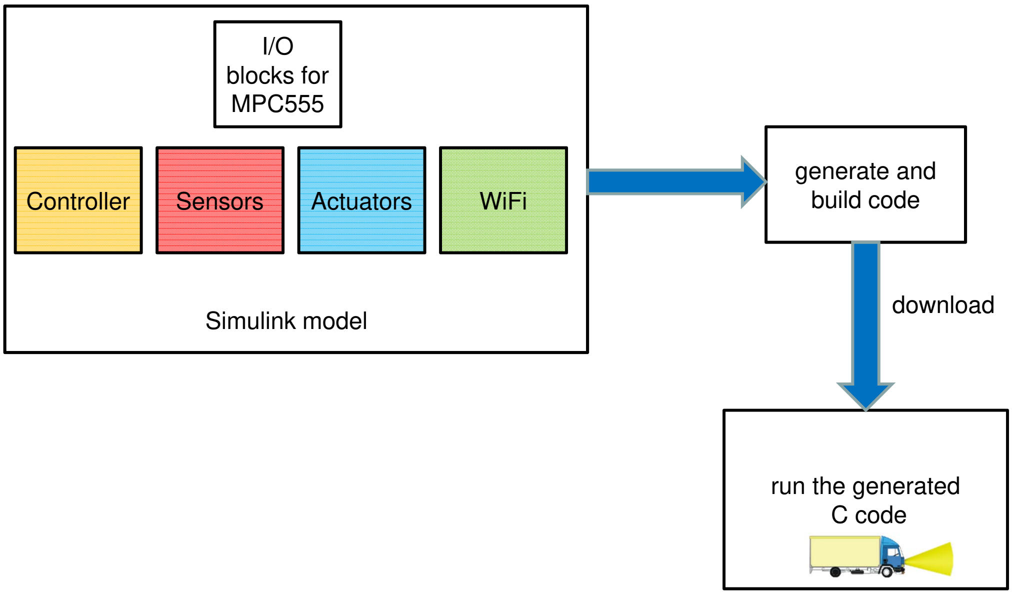


Figure 3.8: Automatic code generation

software. In order to use this GUI, the PC was considered as a node in the network. Moreover, for later data processing, results can be saved as MAT files.

Another important task of this GUI to provide the opportunity to change the parameters of the controller online without stopping the experiment for the purpose of reprogramming the hardware. This can be done by sending the changes from the host PC via WiFi. The user directly observes the effects of this modification and can adjust the control parameters appropriately.

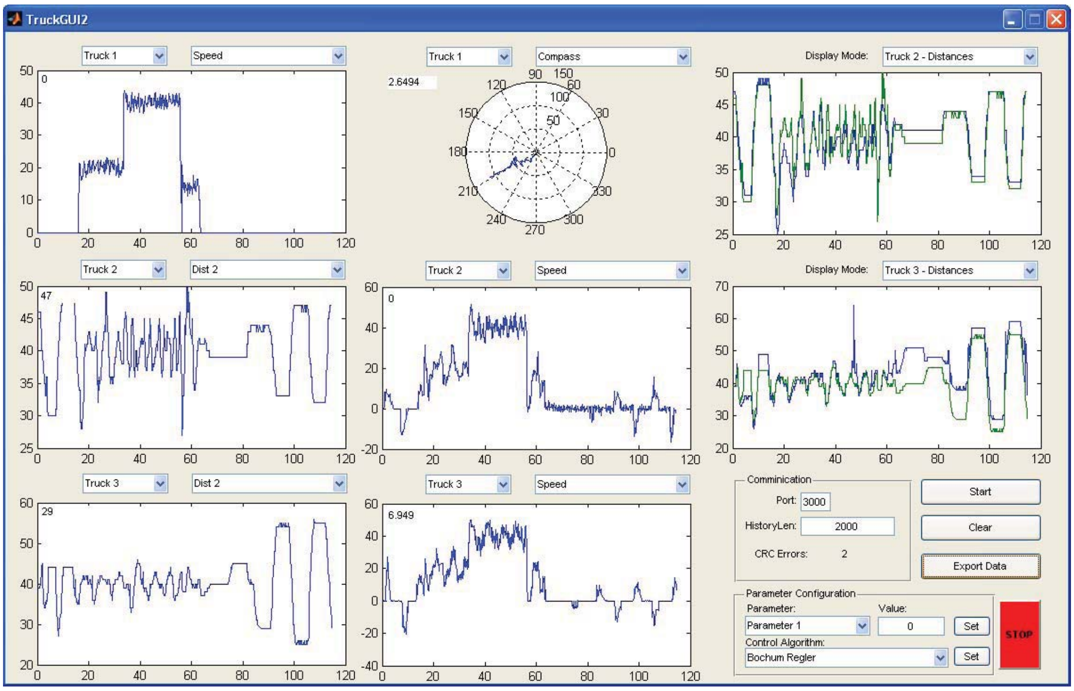
For experimental purposes, we modified the GUI to be able to send an advisory speed to the platoon leader. The leader uses this speed for its longitudinal driving.

Since one major interest of this study is to survey the effects of communication losses and communication delays on the performance of the controlled platoon, the developed GUI is able to send information to the vehicles directly with emulated communication problems. This results in different statuses of the communication network. Possible communication statuses and scenarios are:

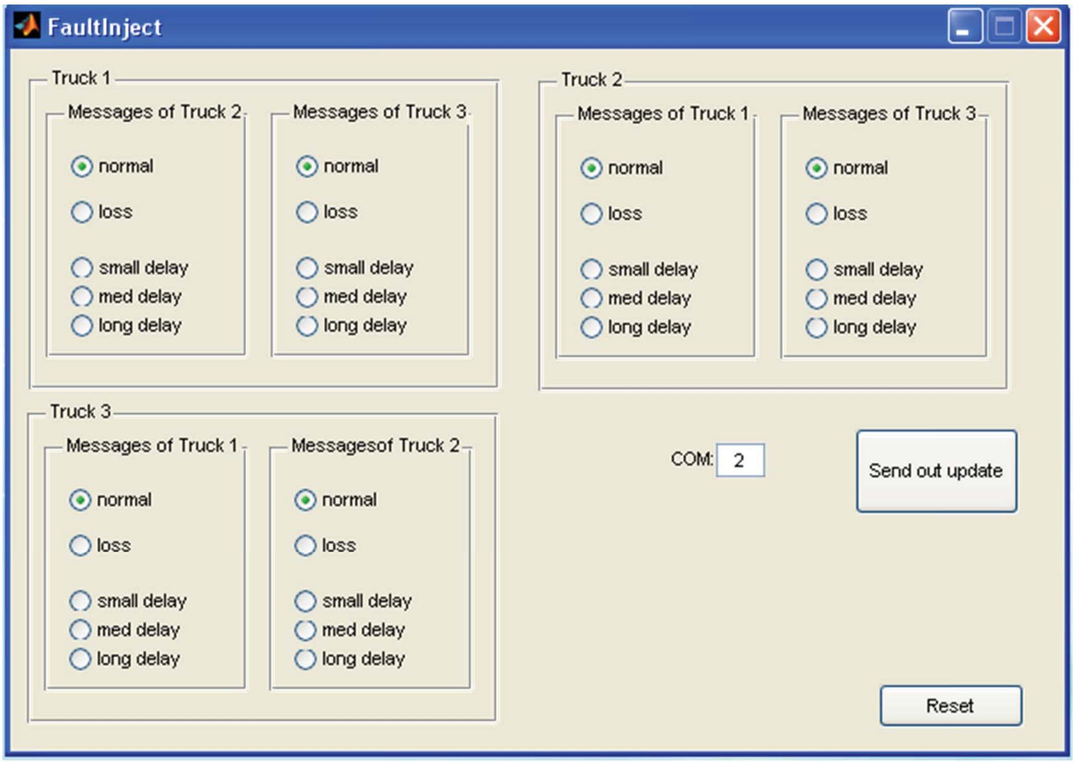
- Normal connection
- Total connection loss
- Connection loss between two designated vehicles
- Communication delays between vehicles

3.2.3 Network Structure

All trucks in the platform are equipped with a WiFi module, an Avisaro WLAN Module 2.0; forming a communication network for exchanging the required information. This



(a) Platoon mode



(b) Fault injection mode

Figure 3.9: Interactions through different GUIs

module supports two operation modes, the streaming mode and the packet mode. In contrast to the packet mode, in streaming mode communication time is reduced by sending all data immediately without waiting for the acknowledgment of the other modules. Since communication speed is an important factor for controlling the vehicles properly, the streaming mode is chosen for our platform.

However, using the streaming mode has a major drawback since the packet flow in this mode is controlled by the WiFi module. This means that the source of the received data cannot be determined directly. Therefore, special encoding of data is required in order to be able to identify the sender of each byte in the network.

Figure 3.10 shows how each byte is encoded before it is sent. Each byte is divided into two nibbles(each four bits). To each nibble of data other four bits are added. The first bit in these added bits is set as the start bit while the following three bits encode the identity of the communication device. The three ID identifying bits can therefore enable the coding of eight active devices in the communication network.

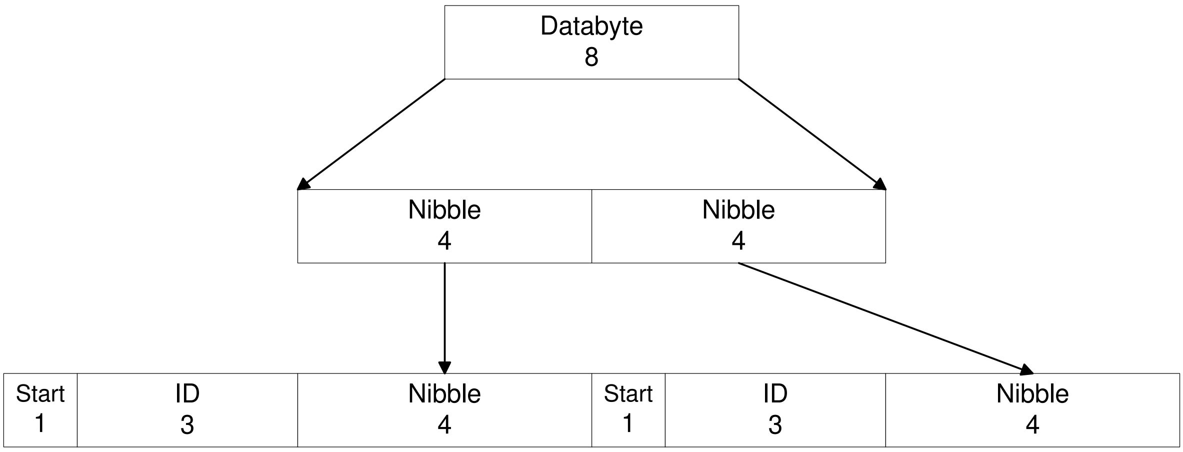


Figure 3.10: Encoding of each byte in the network

The main controlling microcontroller, the MPC555, sends via a serial interface all sensor data (seven sensors values, each 16-bit) to the WiFi module. In addition, three extra 16-bit-data are transmitted and can be used to send further information to the platform from each vehicle. Consequently, a message of 20 bytes has to be transmitted to the other modules. Since we opt for using the streaming mode we included one CRC (cyclic redundancy check) byte with this message. This byte is added to the message in order to detect whether the data has been corrupted during the transmission. The message sent via the WiFi module is shown in Figure 3.11. This module is configured to broadcast the message every 100 ms.

Angle front	Dist 1	Dist 2	Speed	Compass	Acc long	Acc cross	Data 1	Data 2	Data 3	CRC
16	16	16	16	16	16	16	16	16	16	8

Figure 3.11: The content of the message sent via a WiFi module

3.3 Control of Cooperative Scaled Vehicles Driving in Platoon

The platform is designed to test different control algorithms. The goal of these control methods is to guarantee small and constant gaps between the platoon trucks without affecting the stability of the total system. In a cooperation of the "Institut für Regelungstechnik" (IRT), some control methods were designed and tested to be well suited for our scaled vehicles. The methods presented is this subsection results from the cooperation with the IRT institute. A more detailed description of the used control methods and their characteristics can be found in our earlier cooperation publications ([DCGBM+10] and [MDAK11]).

3.3.1 Methods of Platoon Control

The control of the platoon leader is different from that of the other platoon members. The leader of the platoon can be controlled manually with a joystick, which sends radio frequency signals. The receiver on the truck interprets these signals as electrical control signals in order to determine the values for the actuators of the truck. The longitudinal dynamics of the leader can also be controlled by a sequence of preprogrammed maneuvers sent by the GUI as explained before. As for the followers, an inner control loop was used to slow the dynamics down to emulate real vehicles. Thus, the leader behaves as a first order system with a bigger time constant.

In the following different control algorithms, implemented for controlling the lateral and the longitudinal dynamics of the platoon followers, are explained.

Lateral Control

As already explained in Subsection 3.2.1, the information provided by the camera module mounted on each follower is used to control the lateral dynamics. This was achieved by using a proportional controller to keep the truck in the same track as the vehicle ahead. In order to avoid fast changes in the wheel's orientation a prefilter was applied. Choosing the controller parameter was done manually since it is not possible to measure the angle of the wheels. Since the response of each scaled vehicle model is different to the same inputs, the choice of the controller parameters was done separately for each vehicle.

Longitudinal Control

There are many researchers that are interested in designing control algorithms to control the longitudinal movement of a platoon of vehicles. Here two control strategies, which are implemented in our platform, are described. For both methods we chose the same notations. Figure 3.12 shows the platoon, consisting of one leader and i followers, and the following notations:

- x_i to describe the longitudinal position of the followers, where x_l is the position of the leader.

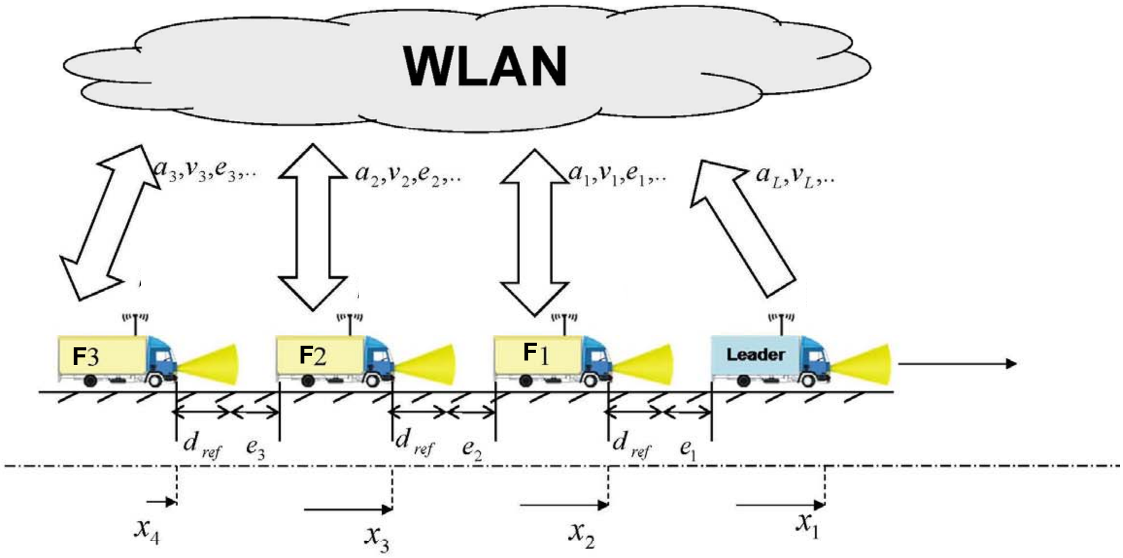


Figure 3.12: The structure of the platoon with notations

- The spacing between the vehicle i and its successive vehicle $i+1$ can be given by $d_i = x_i - x_{i+1} - l$, where l is the vehicle length.
- In order to control the spacing between the vehicles, the spacing error e_i is defined as the relative motion of each vehicle to one another and can be calculated as the difference between the actual distance to the predecessor and a (fixed) reference distance d_{ref}: $e_i(t) = d_i(t) - d_{ref}$.
- The follower vehicles provide the measurements of the distance to the vehicle ahead d_i and their velocities v_i in addition to the vehicle acceleration a_i.

For the longitudinal control of the vehicles of the platoon, some strategies were combined in order to achieve the desired performance. For both control methods, that we will introduce, it was necessary to compensate for the nonlinearities, results from the nature of the scaled model. In the following we explain this procedure.

Compensation of Nonlinearities

The scale vehicle models are nonlinear. This includes, for example, a dead-zone in the longitudinal dynamics. Moreover, the behavior of the vehicle depends considerably on the type of battery being used and its charge level. To compensate these nonlinearities it was necessary to implement an inner control loop consisting of a PID controller with an anti-windup system. In order to adjust the parameters of the controller, a linearized model of the vehicle was employed. This model was identified at a given operating point with a full battery. The parameters of the controller were tuned manually until the desired

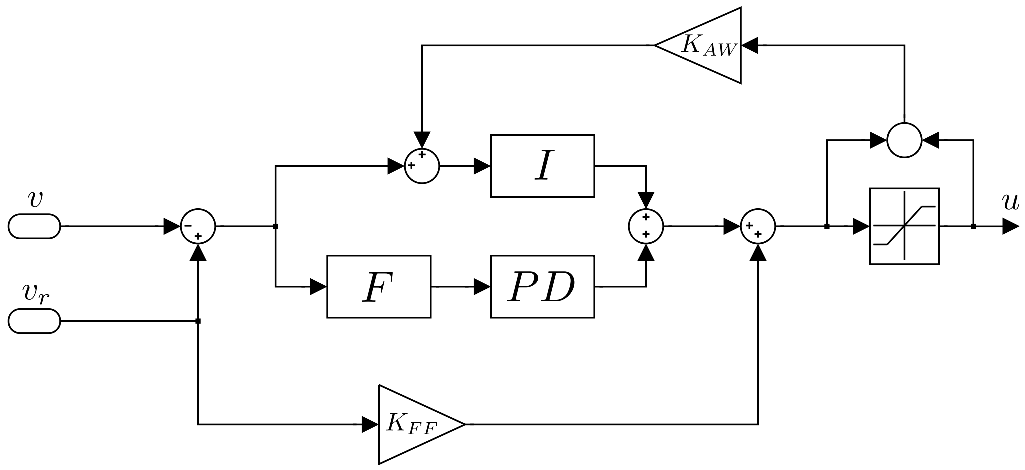


Figure 3.13: Inner loop of the control structure [DCGBM+10]

performance was achieved. This step was actually done by using the corresponding GUI (see Section 3.2.2) for transmitting new values for the parameters over the network.

This underlying controller tracks a reference speed v_r and gives the value of u_i as a control variable to the truck model. The input to the motor-actuator (output of the controller) has been normalized to the interval $u \in [-1, 1]$ where the value -1 means full power in opposite direction. Another purpose for this controller is to slow down the vehicles' dynamics to obtain time constants that are similar to the ones found in real trucks.

The structure of the inner loop controller is illustrated in Figure 3.13. It can be seen that it includes an integrator as well as a PD part preceded by a filter. This filter was necessary because of the noised signals coming from the encoders. The static factor K_{FF} plays the role of a feedforward controller, and its value was designed on the basis of the identified model. The feedback part of the controller accounts for correcting the errors caused by uncertainties in the model (for example, if the battery is not fully charged).

The anti-windup system was designed following the lines of [14] and it is shown on the right side of Figure 3.13, where the saturation block appears. This part of the system is responsible for detecting if the maximal output has been reached. Thus, these blocks were designed to prevent the integrator from continuing to accumulate the error signal.

This inner loop controller allows the approximation of the velocity dynamics by a linear first-order lag element and the approximation of the behavior of the system can be described by the transfer function:

$$G_v(s) = \frac{1}{\tau s + 1} = \frac{V(s)}{V_r(s)},$$

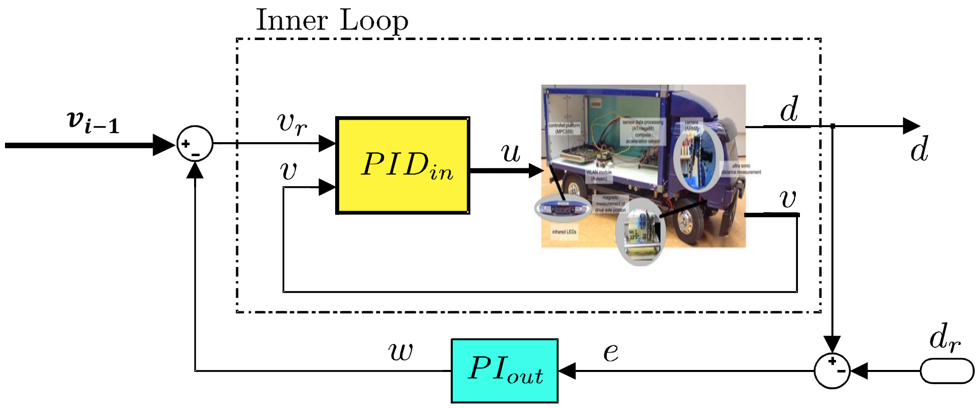


Figure 3.14: Cascade control structure [DCGBM+10]

Applying these to the platoon vehicles, the obtained platoon model can be described as follows:

$$\dot{e}_i = v_{i-1} - v_i, \tag{3.1}$$
$$\dot{v}_i = -1/\tau \cdot v_i + 1/\tau \cdot v_r. \tag{3.2}$$

where v_r is the controller output to be designed.

Two Control Strategies

In addition to the underlying PID controller explained before, another controller should be added with the aim of regulating the distance between the vehicles. Here the development of the controller for the electric model cars will be described. The related publications ([DCGBM+10] and [MDAK11]) used almost the same notations and explain the design of the controller in more detail.

1. **Using a PID Controller in the Cascade Control Structure:**

   This method needs only the information from the vehicle ahead and the design of this controller depends on this structure. By applying Equation 3.1 in 3.2 the expressions

$$\dot{v}_{i-1}\,\tau + v_{i-1} = v_r + \dot{e}_i + \ddot{e}_i\,\tau\,, \tag{3.3}$$

   can be obtained. By making the assumption that the operating point is consistent with driving with a constant velocity, the term $\dot{v}_{i-1}$ can be ignored. Another assumption was made, namely that the necessary information is provided over the network, which allows to divide the controller into a feed-forward part and a feedback part. The feed forward part, which is already explained in the subsection before, aims to compensate the term v_{i-1}.

By notating w as the feedback part of the control law, we can formulate the controller output as the following: $v_r = v_{i-1} + w$. If the difference between the transmitted and the real speed of the vehicle ahead can be approximated by a constant α, the following expression can be written:

$$\alpha = \dot{e}_i + \ddot{e}_i \tau + w. \tag{3.4}$$

As a controller, a PI controller is used making the feedback part of the control law written as follows:

$$w = k_i \int e_i \, dt + k_p \, e_i \tag{3.5}$$

By doing that the Laplace-transforming of the error signal, taking into account a zero initial condition, can be formulated as:

$$E(s) = \frac{\alpha}{\tau \, s^3 + s^2 + k_p \, s + k_i} \tag{3.6}$$

This equation indicates that the system can have a zero steady-state error under two conditions: For a constant α, and for an appropriate choice of the parameters. The implemented controller for the experiments is also equipped with an anti-windup system, similar to the one mentioned in the previous subsection.

Figure 3.14 shows the resulting control structure. The velocity v_{i-1}, which is the velocity from the vehicle ahead, is sent through the network via wireless communication. This figure also shows the inner loop control and is represented in the figure as the block PID_{in}. The block PI_{out} represents the feedback part of the controller which was designed in this subsection.

2. **Using a State Feedback Structure:**

   This method also uses the underlying PID controller described before in order to control the vehicle speed. The value of the D part, however, was set to zero. The time constant in the transfer function in (3.1) has been set to $\tau = 0,5s$. The analysis of the step response was done in order to identify of the transfer function. The response to the applied PI controller has shown a good tracking of the reference velocity.

   The stabilization of the vehicles in the platoon formation can be achieved by applying full static-state feedback control, by assuming that all states are available, either measured or transmitted through the network. Therefore this method uses not only the data from the vehicle ahead but also benefits from the communication network, so each vehicle uses the data from all the vehicles ahead. The problem can be then be summarized as finding a stabilizing feedback matrix where the controller also achieve the requirements of tracking and disturbance rejection in terms of small spacing errors and avoidance of collisions at a reasonable control effort. For controlling the distances between the platoon's vehicles, in this method we used the state feedback structure for the platoon controller introduced in [MDAK11].

Taking the derivative of the Equations (3.1)-(3.2) the dynamics of the platoon can be given as:

$$\begin{aligned} \ddot{e}_i &= a_{i-1} - a_i, && (3.7) \\ \dot{a}_i &= -1/T_i \cdot a_i + 1/T_i \cdot \dot{v}_{r_i}. && (3.8) \end{aligned}$$

From both of these equations the dynamics of the whole platoon can be summarized in state-space form with the state vector $x = [\cdots e_i, \dot{e}_i, a_i \cdots]^T$. The leading vehicle's acceleration a_L enters the dynamics as a disturbance and thus leads to the following linear state-space description:

$$\dot{x} = Ax + B_1 a_L + B_2 u. \tag{3.9}$$

The design of a controller, employed for stabilizing the platoon and to achieve an avoidance of collisions between the platoon vehicles, can be considered as a problem of linear optimal control including a weighting of spacing errors and control signals. In order to consider the performance restrictions, due to the limits of the system variables, a suitable tuning of the H_∞ norm of the corresponding transfer functions(spacing errors and accelerations with the disturbance input of the leading vehicle's acceleration) by adding them to the objective function. The resulting optimization problem was already presented in [68] and was also used for the control layout in this work.

This optimization is applied to a state feedback structure leading to the following control law:

$$u = K_1 \cdot x + K_2 \cdot a_L. \tag{3.10}$$

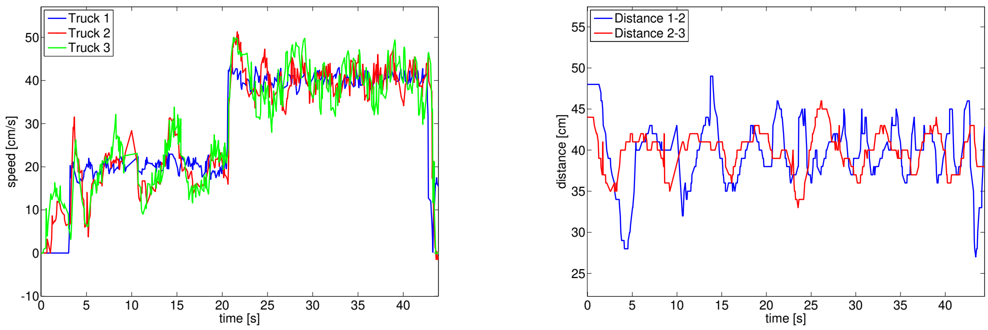
Taking the case when there is no disturbance in the information, this leads that $K_2 = 0$, and the closed loop system is given by:

$$\dot{x} = (A + B_2 K_1)x + B_1 a_L. \tag{3.11}$$

Here the optimization problem is solved for K_1 to achieve the best disturbance behavior with respect to the corresponding objectives.

3.3.2 Experimental Results

In order to show the effectiveness of the two mentioned control algorithms many experiments were carried out for both lateral and longitudinal control. Next the experiments and the corresponding results will be reported.



(a) Speed vs time: Stair-step signal (b) Distance vs time: Step signal

Figure 3.15: Stair-step driving

Results from the First Method Using the Cascade Control Structure

In these experiments, the platoon consisted of three trucks: a leader and two followers. The controller should maintain a constant distance of 40 *cm* between the vehicles. The controller was tested with different leader behaviors, which will be explained in the following:

1. Stair-step driving: The speed of the leader was changed stepwise between different values (0 cm/s, 20 cm/s and 40 cm/s). Fig. 3.15(a) shows the velocity of the platoon vehicles under the gradually changes of the leader's velocity and Fig. 3.15(b) shows the corresponding distances between the trucks.

   The communication between the platoon's vehicles has an important effect on system performance. The WiFi connection between the vehicles informs the followers about the change of the speed of the leader before the ultrasound sensors detect it. Therefore, a rapid change of the leader's speed can be detected quickly by the followers and the reaction to significant changes of speed can take place faster. A scenario where the other vehicles start to move before the leader does can be seen in Figure 3.15(a) at the beginning of the experiment. Thus, not choosing the controller parameter probably can lead to an inconvenient situation if other vehicles move, just by knowing that the leader should move next.

   Figure 3.15(b) shows a deviation up to ± 12 *cm* to the desired space between trucks, which is 40 *cm*. Taking the scale factor of the trucks into account, this variation can be considered as acceptable.

2. Gradual change of leader's velocity: This experiment is different from the one above, since the leader's velocity is changing periodically and gradually taking a sinusoidal shape starting from 0 cm/s and ending at 40 cm/s. Figure 3.16 depicts the velocity of the platoon vehicles. This figure shows how the platoon's followers can track the leader accurately.

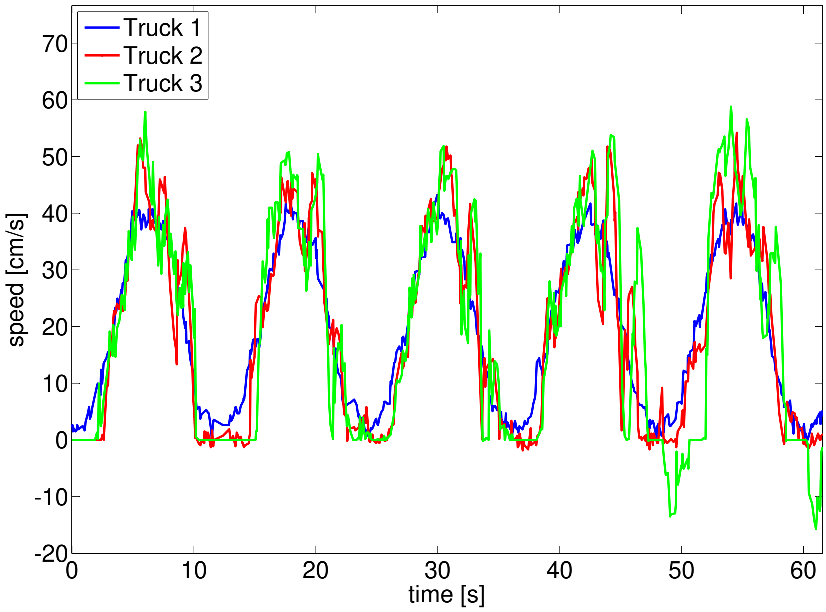
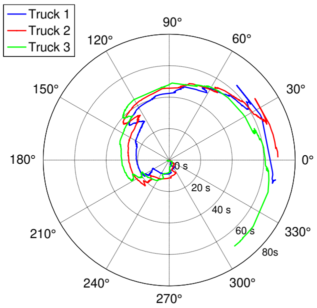


Figure 3.16: Speed vs time: Sinusidal change of the steering signal

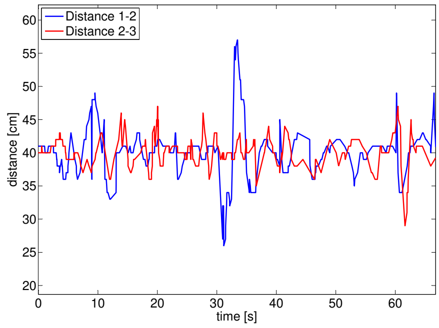
3. Circular driving: In this experiment the platoon's leader was driven in a circle trajectory. The goal of this experiment was to test the lateral dynamics of the controller. Figure 3.17(a) shows the direction of each truck represented by a compass plot, and Fig. 3.17(b) shows the corresponding distances between the trucks.

   The controller in this experiment uses information delivered from one of the ultrasound sensors; the one mounted on the left side of the front of the truck. As a consequence, a short-time disruption of the visibility to the truck ahead could occur. In time interval $[33, 34]s$ an error in the measurement due to this disruption of the visibility occurred. However, this error has not affected the circular path tracking of the platoon, which was successfully completed. Therefore, both of the ultrasound sensors were then used in order to avoid these kinds of error.

One advantage of this platform is that it enables fast, online testing of different types of controllers. Another experiment was made in order to test different modes of the controller, described in this subsection, by changing the proportional part value of this controller. Figure 3.18 shows the distances between the leader and the first follower in three different proportional part values. The figure shows that changing the P part of the controller leads to different reaction times in the movement of the vehicles. Similarly, other controllers or their parameters can also be tuned and compared in order to reach a better performance.



(a) Direction



(b) Distance vs time: Step signal

Figure 3.17: Circular driving

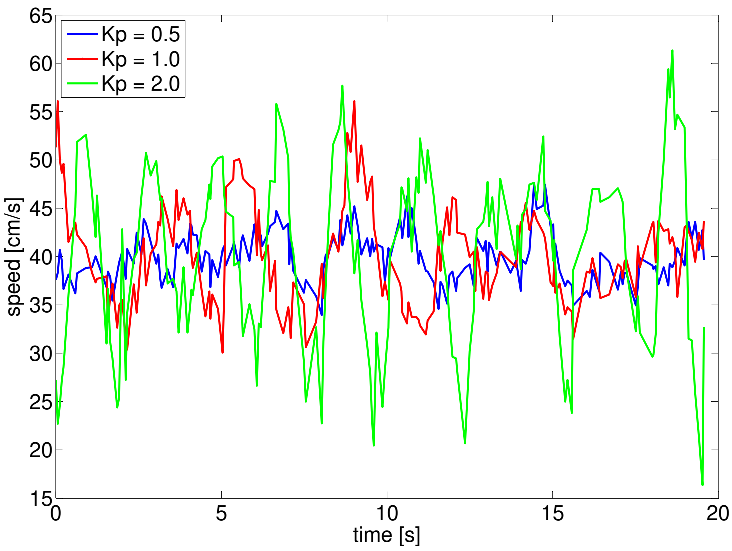
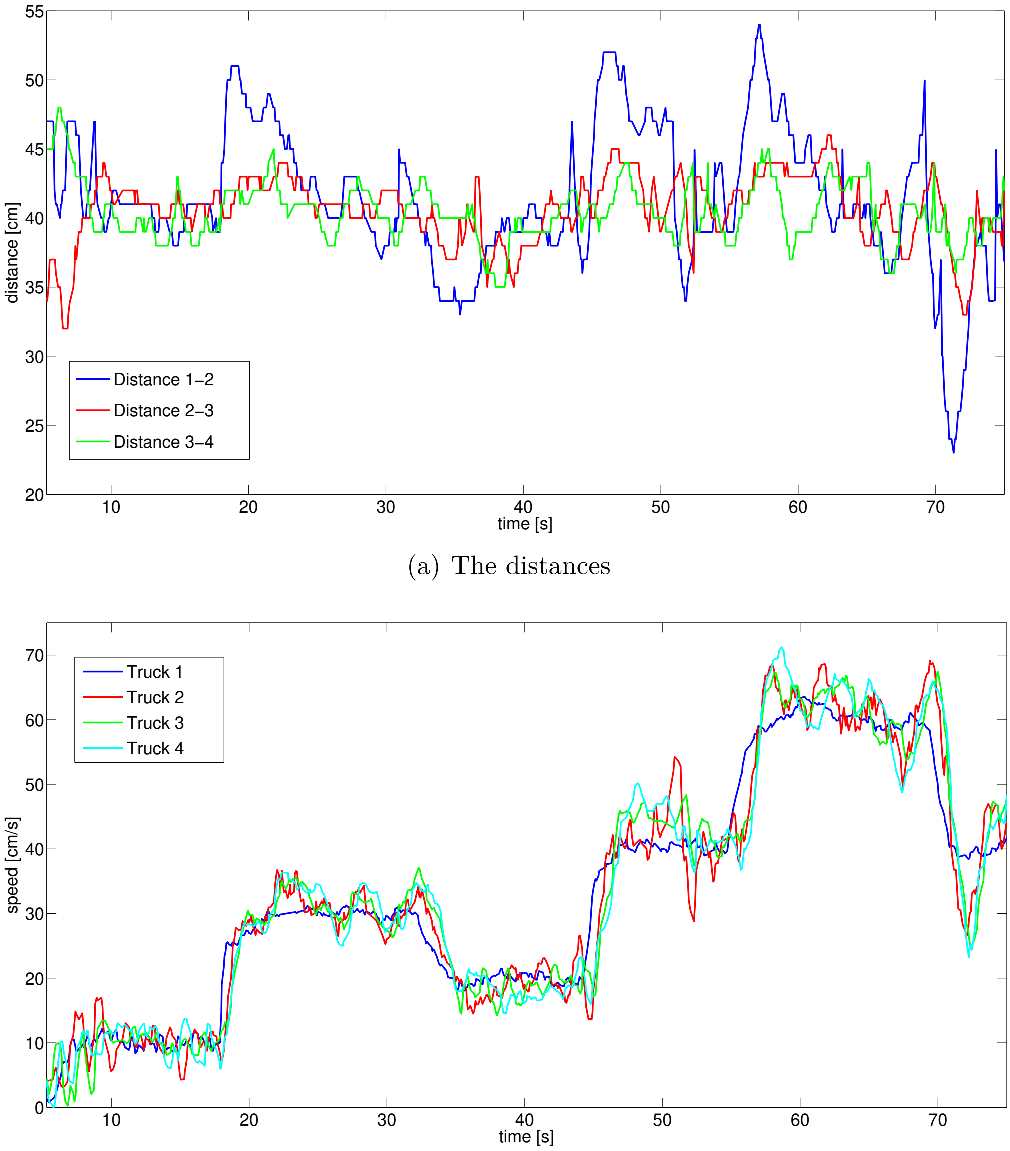


Figure 3.18: Speed VS. time: Different controller modi

Results from the Second Method Using the State Feedback Structure

In this experiment the platoon consists of four vehicles. The leader is a scaled tractor-trailer while the followers are normal scaled trucks. The proposed control approach with the state feedback structure was tested in this experiment. A stair-step driving of the leader was applied within the range of 10 cm/s to 60 cm/s. Fig. 3.19(b) shows the velocity of the platoon vehicles and Fig. 3.15(b) shows the corresponding distances between the trucks. It is shown that all velocity changes within this range were tracked by the model scaled vehicles. We also noted that maximum spacing errors are limited to 15 cm and decline from the first to the last follower.

Further experiments using this controller will be shown later in this thesis, in Chapter 5, since the same controller was used for testing how a platoon can cross an intersection.



(a) The distances

(b) The speed

Figure 3.19: Results of the control of a four-vehicle platoon

4 Formal Verification Methods for Platoon Safety

Some embedded systems can be seen as an interaction between different areas, such as computing, control and communication. In the design of some of these systems, safety must be taken into consideration. These systems are called safety-critical, where an error in the system could cause a dangerous situation and might put the lives of people at risk. Therefore it is important to find methods to guarantee that the device or the system is functioning properly in different scenarios. For controlled dynamic systems, simulations are the typical method for detecting errors in the design process. Here numerical techniques using many plots of different values of the controlled parameters are applied in order to analyze the system's behavior. However, even when a very large number of simulations is used, the assurance of safety cannot be guaranteed.

To proof that the system is correct and consistent with the specifications, formal methods are used. System verification has been applied in order to determine the safety of a system (predefined as a formal specification) by covering all the possible scenarios. Using formal methods can reduce the time needed for a large number of simulations, which consequently reduces the costs of the development process. Safety verification in the field of transportation systems is an important field for a number of studies because of the high safety-critical nature. These studies can gain more importance in many intelligent vehicle applications, where the vehicle can sense its surroundings and performs automated or semi-automated driving scenarios; raising the safety factor in such systems.

In this chapter the safety of a platoon of vehicles was investigated. Taking the failure of the communication in the platoon into account, a safety analysis and methods for modeling and analyzing the system are applied. Some control strategies used for platooning control by means of reachability analysis of dynamic and hybrid systems were analyzed.

External Contributions

Section 4.3 presents methods for the modeling and verification of the platoon behavior. The related publications ([BMMH+11],[BMDK12]) explain the verification results of these systems in more detail. The calculation of the reachable sets for both states, continuous and hybrid systems, was conducted by Ibtissem Ben Makhlouf. The modeling of the platoon in both states is a result of cooperation work with her. The method presented in Subsection 4.2.2 for calculating the reachable states was developed in cooperation with Paul Hänsch.

4.1 Safety Verification: Background

With the increase of the complexity of a control system, it is very important, especially for safety-critical systems, that such systems are proven to be immune to possible faults and that the system's performance will still be acceptable under some kind of system failures. For systems with uncertain initial states and/or inputs, an infinite number of simulations is necessary to cover all the possible scenarios. In such cases simulation techniques could point out some errors in the system design, but in general they are not sufficient to assure safety.

Using model-based design techniques in the design process of such systems has many advantages and could be the key to assure safety [44]. In these methods the description of the systems is formed by mathematical semantics and then the model is analyzed and the requirements are checked. Thus, using formal methods, safety verification can provide a proof if the safety of the system will still be intact even in presence of all possible combinations of failures [25]. This is done by testing if unsafe states can be reached under all the circumstances.

Many techniques are used to verify software and to proof the functionality of the behavior of a number of complex systems. Methods such as theorem proving and model checking have been used for verifying the properties and the safety of some safety-critical systems. Formal methods have been used for a number of industrial applications and the search for advanced methods and tools has been growing.

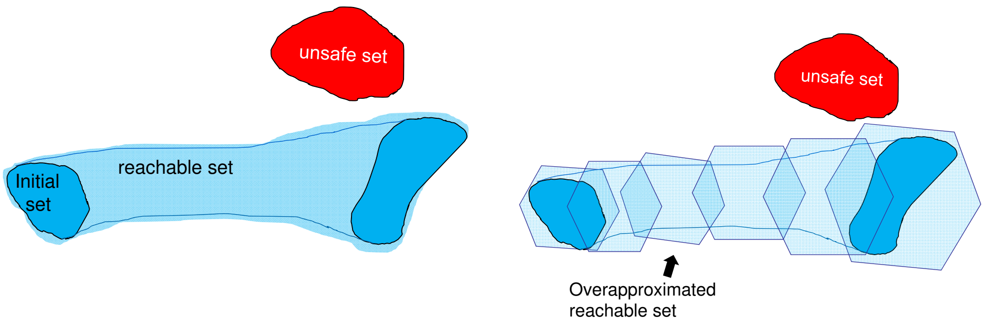
The goal of the verification techniques for continuous and hybrid systems (a combination of discrete and continuous systems) is to assure that the system trajectory does not intersect with predefined unsafe sets or so-called bad states. An example of such an unsafe state would be the collision between vehicles or objects in general. In this work the formal analysis technique used to proof the safety is the reachability analysis.

4.1.1 Reachability Analysis

Reachability analysis is an important tool for checking the correctness and safety of embedded controllers. This method calculates all the states that the system could reach starting from a given state. If it is guaranteed that the predetermined unsafe sets do not intersect with these calculated sets, the safety can be approved. Thus, this method is shown to be useful for detecting some design failures for controlled embedded systems. In this thesis the platoon behavior can be described and modeled as a continuous and/or hybrid system.

The major drawback of reachability analysis is the problem computations' complexity for a high dimension of the continuous state space. This increase in computation effort is exponential with respect to the increase of state dimensions, which limits the applicability of this method. Therefore many researchers are seeking the optimal solution for this problem. Two main approaches have been investigated:

1. Model order reduction methods:



(a) Reachable sets without approximation (b) Reachable sets with approximation

Figure 4.1: Reachable sets

Some researchers construct reduced-order models in order to make the number of the continuous states acceptable for the most verification tools. This reduction is kind of an approximation which must take the error of model reduction between the original model and the reduced one into consideration. An example of this technique can be found in the work of Han and Krogh [38].

2. A proper choice of the representation sets:

   A proper chose of the representation sets could allow the verification for systems with high dimension states. Exact computing of reachable states of a dynamic system is generally not possible ([43, 58, 77]) and this can be obtained only for very special classes of linear systems, see e.g. the work of Lafferriere et al. [57]. However, approximation techniques can be applied and calculate the over- or underapproximations of the reachable states. Depending on the problem, good over- or underapproximations are often sufficient. So in general it is usual and enough to show that an overapproximation of the reachable states does not intersect with the critical region. Therefore, a proper computation of overapproximations of reachable sets is sought.

Set Representations

An appropriate choice of data structures to represent the approximations of reachable states is necessary since it could strongly affect the verification process. In the literature there are many suggested data structures to represent the sets, such as boxes ([80]), polytopes ([39]), polyhedra ([32]), ellipsoids ([55]), zonotopes ([8, 36]) and support functions ([37]). Each one of these representations has disadvantages and advantages in the sense of computation time and the simplicity in the representation. Boxes, for example, are very simple data structures and easy to handle, but they introduce larger approximation errors than other representations.

For the computation of the reachable sets, a number of operations should be performed. Given two sets S_1 and S_2, definitions of basic set operations are given.

Definition 1 ($\oplus$-Operator is Minkowski sum)**.** The Minkowski sum of these two sets is denoted by $\oplus$-Operator and defined by $S_1 \oplus S_2 := \{s_1 + s_2 \mid s_1 \in S_1, s_2 \in S_2\}$

Definition 2 ($\mathcal{C}H$-Operator is Convex hull sum)**.** The convex hull of sets is denoted by CH-Operator and defined by the hull of its arguments.

Definition 3 (Linear Transformation $\mathcal{L}$)**.** Linear transformation $\mathcal{L}$ with a set $L.S1$: $A.S_1 = A.s_1|s_1 \in S_1$

4.1.2 Hybrid Systems

Hybrid systems are systems which have a combination of discrete and continuous behaviors. An example of such systems can be found in some sophisticated control systems where such systems have on one hand a continuous behavior due to the system's dynamics, and a discrete behavior due to the computing on the other [11]. In order to have a formalism of hybrid systems the model should have interactions between dynamics systems and discrete events. The continuous dynamic systems are usually defined by their differential equations, where the discrete events that occur can be defined as automata. The formalisms of hybrid automaton can therefore give a specification for hybrid systems.

Definition 4 (Hybrid Automaton)**.** In the work of Alur et al. [10] the concept of hybrid automaton was introduced. Starting from the definition of the state-space model on the continuous side, and the paradigms of the automata on the other side, the definition of the hybrid automaton has emerged. The hybrid automaton can be seen as finite state automaton with discrete nodes, where in each of these nodes continuous variables change according to the characteristics of the system.

In this thesis we use the following definition and the notation, which is tuple with the components $HA =< Q, \mathcal{X}, \mathcal{U}, F, Inv, G, R, q_init, X_{init} >$:

- Finite set of n discrete states or locations $Q = \{q_0, q_1, .., q_n\}$.
- The continuous system variables $\mathcal{X}$ which can represent the state-space variables.
- The continuous input variables $\mathcal{U}$.
- The flow function $F : Q \times \mathcal{X} \times \mathcal{U}$ which is a vector field of the evaluation of the continues variables.
- Inv is the invariance condition that defines the region of each mode and has the constraint of the dynamic variables. The continuous state x should satisfy $x \in Inv(q_l)$ for $q_l \in Q$.
- G is the description of the guard condition. The system will still be in the same discrete location q_1 until the continuous state x reaches the guard set. Afterwards the system should be switched to an another discrete location q_2 based on this guard condition $G(q_1, q_2)$.

- R is a reset mapping. This is a function which changes the values of the continuous state x vector when the system changes its discrete location.

- An initial graph $q_{init} \in Q$.

- An initial continuous states $\mathcal{X}_{init} \subseteq \mathcal{X}$.

There are a lot of examples of systems which can be modeled using the formalism of hybrid systems. Some of them have been been introduced in van der Schaft and Schumacher [100]. Here we mention a few famous benchmarks:

- Controlling the temperature of the room:

  Here the room temperature represents the dynamics of the system. The status of the heater can be seen as discrete dynamics.

- Bouncing ball:

  When a ball bounces on the ground it loses some of its strength to jump up. These characteristics can be captured by one location and one dynamic variable (distance from ground).

- Power converter:

  This describes the behavior of an electrical circuit with a diode, an inductor, a capacitor and a resistor. This circuit is regulated by an ideal switch. The switches (switch on, switch off) represents the two locations. The magnetic flux and the electrical charge are the two dynamic variables for this system.

For analyzing and testing hybrid systems there have are a number of used tools. Numerical tools from MathWorks such as Simulink and Stateflow has been used to find errors and test the behavior via simulation. For checking and verifying the algorithm, a number of tools used formal methods. Tools such as Checkmate[24], d/dt [13], HyTech [42] and PHaver [32] were developed using the 'model checking' technique. A tool such as HSolver [79] uses the constraint solver as another formal technique .

For verifying hybrid systems the reachability analysis is a method which combines the numerical approaches with the formal methods. Many researchers are in search of computationally efficient schemes for safety verification for both linear and nonlinear systems. For reachable analysis in hybrid systems there are many researchers whose goal is to compute the reachable sets efficiently. However, the works of Ben Makhlouf and Kowalewski [15], Ben Makhlouf et al. [16], and other works show that these methods are limited by the dimension of the continuous state space of the system. Systems with six to seven continuous state variables are difficult to calculate.

4.1.3 Safety Verification in ITS

Many embedded systems used in transportations are complex control systems and by nature safety-critical. An error in the system could lead to a dangerous situation. For intelligent transport systems, verification techniques can be more important and necessary, since taking the control of the vehicle, through very complex procedures, should be safe to a very high state.

Simulations were used to predict different behaviors of different traffic members. Monte Carlo simulation-based techniques have been widely used as a method for estimating the risk probability of collisions in the road traffic. In this technique multiple simulations are performed, where initial sets of positions and velocities of a number of vehicles in the road is defined and the system is let to evolve under these assumptions. Based on observations, these simulations then show, the probabilities of the accident that can occur, which helps in estimating the risk.

Other researchers used verifications methods to prove safety. In the work of Althoff et al. [9], the authors present an algorithm for verifying evasive maneuvers of road vehicles, where the safety of an autonomous vehicles in coordinated maneuvers is investigated. The same authors shows in [7] a method which computes the probability of reaching a safe state after a maneuver for robots. The work of Stursberg et al. [93] deals with the analysis of the cruise control system and proposes an approach to reduce the computational effort of the safety analysis of two cars driving on one lane, using the counterexample method to achieve that.

4.2 Reachability Analysis for Continuous Systems

Reachability analysis is one major approach for the safety verification of dynamic systems in order to assure that the system requirements are guaranteed. This method computes the reachable states, taking all possible initial states and inputs into account. Beginning with an infinite set of initial states and under uncertain inputs and disturbances, all the states that can be reached are computed. Reachability analysis is thus used to check if the dynamic system reaches hazardous states, or not, under all these conditions. This examination of the state space can check for specific applications as well as for general properties. In this section we are interested in computing the reachable sets of linear time-invariant systems.

4.2.1 Computation of the Reachable Sets

We will start with computing the reachable states of linear autonomous systems without an effect of the input as the system described in Equation 4.1. Then we will investigate the effect of the inputs by calculating the reachable sets for the system described in Equation 4.2.

$$\dot{x}(t) = Ax(t) \tag{4.1}$$

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{4.2}$$

where $x(t)$ is a vector of continuous system variables at time t, $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are constant matrices, $u(t) \in \mathbb{R}^m$ is the input to the system at time t bounded by $u(t) \in [\underline{u}_1, \overline{u}_1] \times \ldots \times [\underline{u}_m, \overline{u}_m] =: \mathcal{U}$, for all $t \geq 0$.

In the following we will make some further definitions that will be used in this thesis.

Definition 5 (Trajectory, Reachability)**.** A *trajectory* of system (4.2) is a time-valued continuous function $x : \mathbb{R}_0^+ \to \mathbb{R}^n$ which satisfies the following condition: There exists an admissible input $u : \mathbb{R}_0^+ \to \mathbb{R}^m$ such that for all $t \geq 0$ (except for those where u is not continuous) the function x satisfies Equation (4.2). A state $\xi \in \mathbb{R}^n$ is *reachable* from $\xi_0 \in \mathbb{R}^n$ at time τ if there exists a trajectory x of the system under consideration such that $x(0) = \xi_0$ and $x(\tau) = \xi$.

Definition 6 ($\mathcal{R}$-Operator)**.** By $\mathcal{R}(\mathcal{X}, [t_0, t_1])$ we denote the set of all states $x(t)$ reachable by system (4.2) at a time $t \in [t_0, t_1]$ with the initial condition $x(0) \in \mathcal{X}$ under some admissible input $u \in \mathfrak{U}$. If $\mathcal{X} = \{x_0\}$ is a singleton we write $\mathcal{R}(x_0, [t_0, t_1])$ and for $t_0 = t_1$ we write $\mathcal{R}(\mathcal{X}, t_0)$.

We can now state the *reachability problem* as computing overapproximation of the set $\mathcal{R}(\mathcal{I}, [0, T])$ starting from set of potential *initial states* $\mathcal{I} \subseteq \mathbb{R}^n$ and a time horizon T.

Computing Reachable States of Linear Autonomous Systems

Given a linear time-invariant autonomous system as the system defined in 4.1, the analytical solution for this differential equations admits

$$x(t) = e^{At}x_0,$$

where $x_0 = x(0)$ is the desired initial condition. It follows that the set of states that are reachable from a given set $\mathcal{X}$ at time τ is

$$\begin{aligned} \mathcal{R}(\mathcal{X}, \tau) &= \bigcup_{x_0 \in \mathcal{X}} e^{A\tau} \cdot x_0 \\ &= e^{A\tau} \cdot \mathcal{X}, \end{aligned} \tag{4.3}$$

which is simply a linear transformation of $\mathcal{X}$ and the matrix $e^{A\tau}$ can be computed by standard numerical tools with a high degree of accuracy. For the underlying algorithms please refers to the work of Moler and Loan [71].

Similarly, we have

$$\mathcal{R}(\mathcal{X}, [0, \tau]) = \bigcup_{t \in [0,\tau]} e^{At} \cdot \mathcal{X}.$$

As proposed in the work of Girard [36], this set can be overapproximated by

$$\begin{aligned} \mathcal{R}(\mathcal{X},[0,\tau]) &\subseteq CH(\mathcal{I}, e^{A\tau}\mathcal{I}) \oplus \{x \mid \|x\|_v \le \alpha\}, \\ \alpha &= (e^{\|A\|_m\tau} - 1 - \|A\|_m\tau) \cdot \sup_{x\in\mathcal{X}} \|x\|_v, \end{aligned} \tag{4.4}$$

The matrix norm $\|\cdot\|_m$ and the vector norm $\|\cdot\|_v$ should to each other apply two conditions:

1. submultiplicative: $\|AB\|_m \le \|A\|_m\|B\|_m$ for all matrices $A, B \in \mathbb{R}^{n\times n}$
2. consistent: $\|Ax\|_v \le \|A\|_m\|x\|_v$ for all $A \in \mathbb{R}^{n\times n}$

Equation (4.3) and inequality (4.4) complete the general computation scheme given in Equations (4.14) and (4.15). The remaining details depend on the data structures used in the implementation of this method.

Computing Reachable States of Linear Systems with Inputs

Adding the input to influence the system we have the continuous system with input defined in 4.2. In this work we assume the input $u : \mathbb{R}_0^+ \to \mathbb{R}^m$ to be stepwise constant with respect to a predefined time step r, i.e., $u : (ir, ir+r) \to \mathbb{R}^m$ is constant for all $i \in \mathbb{N}$. We denote the class of all such bounded and stepwise constant (in short *admissible*) inputs by $\mathfrak{U}$. Therefore the problem is to achieve the reachability analysis of linear systems with stepwise constant inputs.

In the standard approach to the reachability problem of linear systems it is known (see e.g. [76]) that the expression

$$x(t) = e^{At}x_0 + \int_0^t e^{A(t-\tau)}Bu(\tau)\mathrm{d}\tau$$

is the solution of the initial value problem $x(0) = x_0$ associated with system (4.2). From this, one can show in a straightforward manner that

$$\mathcal{R}(\mathcal{I},[0,r]) \subseteq \bigcup_{t\in[0,r]} e^{At}\mathcal{I} \oplus \bigcup_{t\in[0,r]} \bigcup_{u\in\mathfrak{U}} \int_0^t e^{A(t-\tau)}Bu(\tau)\mathrm{d}\tau$$

holds for system (4.2) where $\mathfrak{U}$ is the class of allowed input functions. An overapproximation of this set and hence of the reachable states in the first time interval $[0, r]$ can be found in [37], for example.

To cover the successing time intervals $[r, 2r]$, $[2r, 3r]$,... an iterative scheme is needed. It can be derived from another straightforward consequence, namely

$$\mathcal{R}(\mathcal{I}, r) = e^{Ar} \cdot \mathcal{I} \oplus \bigcup_{u\in\mathfrak{U}} \int_0^r e^{A(r-\tau)}u(\tau)\mathrm{d}\tau. \tag{4.5}$$

Time-invariant systems satisfy the following Equation (4.6) and the subsequent (4.7) follows immediately from (4.5):

$$\begin{aligned}\mathcal{R}(\mathcal{I},[ir,ir+r]) &= \mathcal{R}(\mathcal{R}(\mathcal{I},[ir-r,ir]),r) &(4.6)\\ &= e^{Ar}\cdot\mathcal{R}(\mathcal{I},[ir-r,ir])\oplus\bigcup_{u\in\mathfrak{U}}\int_0^r e^{A(r-\tau)}u(\tau)\mathrm{d}\tau. &(4.7)\end{aligned}$$

In the case of stepwise constant input, the above formula can be simplified to

$$\mathcal{R}(\mathcal{I},[ir,ir+r]) = e^{Ar}\cdot\mathcal{R}(\mathcal{I},[ir-r,ir])+A^{-1}(e^{Ar}-I)B\mathcal{U}. \tag{4.8}$$

which, finally, is the desired iterative formula.

The standard approach is therefore to plug the overapproximation of $\mathcal{R}(\mathcal{I},[0,r])$ into Equation (4.8), giving an overapproximation of $\mathcal{R}(\mathcal{I},[r,2r])$ which again can be plugged into Equation (4.8), etc. until the desired time horizon T is reached.

Computing the Reachable Sets Using Zonotopes

The computation scheme presented in the previous section can be implemented with different data structures. Zonotopes can represent complex geometric figures and are still one of the most popular data structures in reachability analysis. In this work, we use zonotopes to over-approximate the reachable set. Here, we will overapproximate each of the sets of reachable states $\mathcal{R}(\mathcal{I},[ir,ir+r])$ with a zonotope. Our choice is motivated by their closure properties under linear transformation and the Minkowski sum.

Definition 7 (Zonotope)**.** A *zonotope* Z of order k/n is a tuple $Z=(c,g_1,\ldots,g_k)$ with $c,g_1,\ldots,g_k\in\mathbb{R}^n$ and $k\geq 0$. The characteristic set of Z is

$$\{c+\Sigma_{i=1}^k\alpha_i\cdot g_i \mid -1\leq\alpha_i\leq 1 \text{ for all } i=1,\ldots,k\}\subseteq\mathbb{R}^n.$$

The parameter c is called the *center* and $g_1,\ldots,g_k$ are the *generators* of the zonotope. The term *zonotope* can refer to this tuple and also to its characteristic set. To simplify the notation, we will sometimes write a zonotope as a matrix, where the first column represents the center and the successive columns represent the generators of the zonotope.

Fig. 4.2 shows an example of zonotopes in 2-dimensional space together with their matrix representation. From left to right:

- $[0 \quad 0)^T]$ has no generators and represents the singleton $(0 \quad 0)^T$.
- $[0 \quad 3;\ 0 \quad 0]$ represents a straight line segment.
- $[0 \quad 3 \quad 0;\ 0 \quad 0 \quad 2]$ expands over the two dimensions.
- $[0 \quad 3 \quad 0 \quad 2;\ 0 \quad 0 \quad 2 \quad 1]$ represents the last figure to the right.

In the following we show the basic set operations while computing the reachable sets with zonotopes:

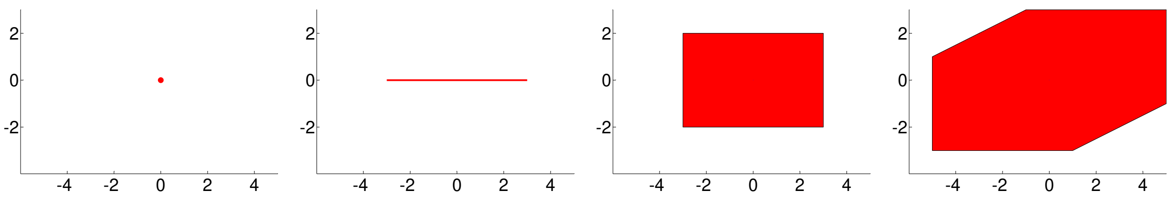


Figure 4.2: Some examples for zonotopes in two-dimensional space

- The Minkowski sum of two zonotopes $Z_1 = (c_1, g_1^1, \ldots, g_1^k)$ and $Z_2 = (c_2, g_2^1, \ldots, g_2^k)$ is simply $Z_1 + Z_2 = (c_1 + c_2, g_1^1, \ldots, g_1^k, g_2^1, \ldots, g_2^k)$.

- Unfortunately, the convex hull of two zonotopes does not need to be a zonotope. Since each zonotope is symmetric about its center, the convex hull of the line segment and the singleton, for example, is therefore not a zonotope. The convex hull of the sets Z and $e^{rA}Z$ may be approximated, according to the algorithm proposed by Girard [36], as follows:

$$\begin{aligned} CH(Z, e^{Ar}Z) \quad \subseteq \quad (& \frac{c + e^{Ar}c}{2}, \frac{g_1 + e^{Ar}g_1}{2}, \ldots, \frac{g_k + e^{Ar}g_k}{2}, \\ & \frac{c - e^{Ar}c}{2}, \frac{g_1 - e^{Ar}g_1}{2}, \ldots, \frac{g_k - e^{Ar}g_k}{2}). \end{aligned}$$

  Note that the resulting zonotope contains then $2k+1$ generators.

- To compute the reachable set we need first of all to evaluate the image of a zonotope Z by the linear mapping e^{rA}:

$$e^{rA}Z = \left(e^{rA}c, < e^{rA}g_1, \ldots, e^{rA}g_k >\right).$$

It is important to note that the two last operations increase the order of the resulting zonotope rapidly. In order to control the complexity, a maximum allowed order is fixed. Methods for this generator reduction could be found as a part of our previous work in [BMMH+11] and in the work in Girard [36]. After reviewing the different zonotope operations we are ready to compute the overaproximation of the reachable set as stated in the iterative formula in 4.8.

Using Zonotopes as Reachable Sets for Hybrid Systems

As explained, zonotopes have been shown as a good representation for the reachable sets. They are closed under the Minkowski sum. Furthermore, methods proposed in [36] for the computation of an over-approximation of the convex hull of both zonotopes Z and $e^{rA}Z$ as well as the proposed order reduction of zonotopes make them more attractive for this practical study. However, this representation faces a problem while calculating the reachability sets for hybrid systems.

Zonotopes are not closed under intersections. This means that computing an approximation of the intersection of two zonotopes can be computationally expensive or inaccurate or both. This could be a problem in computing the reachable sets of hybrid systems when the calculation of the intersection between the reachable sets with guard sets, which can be represented as zonotope or polytope, is needed. In particular, the computational complexity and inaccuracy of approximated intersections increase with the representation size of the zonotope. Althoff et al. [8] propose an idea of combining zonotopes and polytopes as a representation when the reachable sets face an intersection set. The authors of that work have also proposed a number of methods to switch between both of the representations. Doing the intersection with polytopes is possible and seemly computationally cheap.

In this thesis, the verification of the resulting hybrid system requires no calculation of intersections of the reachable sets with guard sets. Therefore a further investigation of the set intersection problem is out of the scope of the thesis.

4.2.2 Alternative Approach

In this section we present a new approach to calculate the reachable states of linear systems with uncertain inputs under the assumption that the inputs are stepwise constant. This method has already been introduced in our publication with the reference [HDMK13]. The guiding question is whether the growth of the zonotope can be diminished. In the case of linear systems *with inputs*, the representation size of the zonotope (approximating the reachable states) increases with each step of the algorithm. Informally, the representation size of the zonotope is crucial for the accuracy and computation time of reachability analysis in hybrid systems. Our idea is to reshape the system matrix and to map the reachability problem of a system with inputs to a problem that involves only a system without inputs. This approach is not restricted to a special geometric data structure. It assumes stepwise constant inputs, which is a reasonable assumption in many scenarios: for example, if the input is determined by some digital controller.

The original system S with inputs is transformed into a system S' without inputs such that the reachability problem of S can be reformulated as a problem that involves only S' and thus the inputs need no longer be considered. Our idea is to map the reachability problem of a system with input given by Equation (4.2) to the reachability problem of an autonomous system,: i.e., a system without the input Bu, such as the system given in (4.1). Under the assumption that the inputs are stepwise constant, this can be done by shifting the input Bu into the state space. The price we pay for having a system without inputs is that the number of dimensions increases by the number of inputs. In the following we consider the following symbolic example of a n-dimensional system

$$\dot{x}(t) = Ax(t) + u(t), \quad u(t) \in [\underline{\mu}, \overline{\mu}] \times \{0\} \times \ldots \times \{0\}, \tag{4.9}$$

where only the first component u_1 of the input u is not zero. For system (4.9) we define the *lifted autonomous* system by

$$\dot{x}'(t) = \underbrace{\left(\begin{array}{c|c} & 1 \\ A & 0 \\ & \vdots \\ \hline 0\dots0 & 0 \end{array}\right)}_{A'} x'(t) \tag{4.10}$$

which has $n+1$ dimensions. The additional state variable x'_{n+1}, which is the last component of the state vector x', substitutes the input. As one can tell from the last row of A' the variable x'_{n+1} does not change over time. This corresponds to our assumption of stepwise constant input. Further, because of $A'_{1,n+1} = 1$ and $A'_{2,n+1} = \ldots = A'_{n+1,n+1} = 0$ the variable x'_{n+1} has an impact only on $\dot{x}'_1$ just as $u(t)$ has an impact only on $\dot{x}_1$. Finally, if we take for $x'_{n+1}(0)$ all values in $[\underline{\mu}, \overline{\mu}]$ as possible initial values into account, then the variables $x'_1, \ldots, x'_n$ can reach the same states as $x_1, \ldots, x_n$, as long as the input $u(t)$ is a constant value in $[\underline{\mu}, \overline{\mu}]$.

In order to formalize the above paragraph, we introduce some further notations.

Definition 8. ($[\mathcal{R}(\cdot,\cdot), \mathcal{R}'(\cdot,\cdot)]$)
$\mathcal{R}(\cdot,\cdot)$ denotes the reachable states of system (4.9), i.e., the original system.
$\mathcal{R}'(\cdot,\cdot)$ denotes reachable states of system (4.10), i.e., the lifted autonomous system without input.

Definition 9. ([The projection π_k])
The projection π_k is defined as the projection of a vector onto its first k components. for example, $\pi_2((4,5,6)^T) = (4,5)^T$.
We extend this projection to sets of vectors by $\pi_k(V) := \{\pi_k(v) \mid v \in V\}$. Then, for example, $\pi_2(\{(4,3,2,1)^T, (5,6,7,8)^T\}) = \{(4,3)^T, (5,6)^T\}$.

Now we can say that, under the assumption that the input u is constant in $[0, r]$ and restricted to values in $[\underline{\mu}, \overline{\mu}]$,

$$\mathcal{R}(\mathcal{I}, \tau) = \pi_n(\mathcal{R}'(\mathcal{I} \times [\underline{\mu}, \overline{\mu}], \tau))$$

holds for all $\tau \in [0, r]$ and hence also

$$\mathcal{R}(\mathcal{I}, [0, r]) = \pi_n(\mathcal{R}'(\mathcal{I} \times [\underline{\mu}, \overline{\mu}], [0, r])) \tag{4.11}$$

holds.

Time-invariant dynamic systems (as is the case for (4.9)) satisfy the following property

$$\mathcal{R}(\mathcal{I}, [a+\delta, b+\delta]) = \mathcal{R}(\mathcal{R}(\mathcal{I}, [a, b]), \delta).$$

Now, one can derive

$$\begin{aligned} \mathcal{R}(\mathcal{I}, [ir, ir+r]) &= \mathcal{R}(\mathcal{R}(\mathcal{I}, [ir-r, ir]), r) \\ &= \pi_n(\mathcal{R}'(\mathcal{R}(\mathcal{I}, [ir-r, ir]) \times [\underline{\mu}, \overline{\mu}], r)), \end{aligned} \tag{4.12}$$

where the first equation follows immediately from the above property, and the second equation is true under the assumption that, starting from $R(\mathcal{I}, [ir-r, ir])$, the input remains constant for r time units.

Equations (4.11) and (4.12) provide a way to reduce the reachability problem for the original system with inputs to the reachability problem of a system without inputs:

1. To compute the reachability $\mathcal{R}(\mathcal{I}, [0, r])$ Equation (4.11) can be used if there is a method to compute the reachable states $\mathcal{R}'(\cdot, [0, r])$ of the lifted autonomous system. The necessary technique to compute $\mathcal{R}'(\cdot, \cdot)$ has already been discussed in Subsection 4.2.1.

2. Equation (4.12) shows how to compute $\mathcal{R}(\mathcal{I}, [ir, ir+r])$ if we have $\mathcal{R}(\mathcal{I}, [ir-r, ir])$ and a method to compute $\mathcal{R}'(\cdot, r)$.

The Generalized Method

Here we want to solve the reachability problem for a general form of linear systems given by Equation (4.2) and by considering stepwise constant inputs. For this system we define its *lifted autonomous* system by the state vector $x' = (x'_1, \ldots, x'_{n+m})^T$ and the dynamic behavior

$$\dot{x}'(t) = \underbrace{\left(\begin{array}{ccc|c} & A & & B \\ \hline 0 & \cdots & & 0 \\ \vdots & & & \vdots \\ 0 & \cdots & & 0 \end{array} \right)}_{A' :=} x'(t). \tag{4.13}$$

The variables $x'_1, \ldots, x'_n$ correspond to the original state variables $x_1, \ldots, x_n$. Whereas the variables $x'_{n+1}, \ldots, x'_{n+m}$ are substitutes for the m different inputs.

As before, $\mathcal{R}(\cdot, \cdot)$ denotes the reachable states of the original system (4.2) and $\mathcal{R}'(\cdot, \cdot)$ those of the lifted system (4.13).

Equations (4.11) and (4.12) from the previous subsection can be generalized to

$$\mathcal{R}(\mathcal{I}, [0, r]) = \pi_n(\mathcal{R}'(\mathcal{I} \times \mathcal{U}, [0, r])) \tag{4.14}$$

$$\mathcal{R}(\mathcal{I}, [ir, ir+r]) = \pi_n(\mathcal{R}'(\mathcal{R}(\mathcal{I}, [ir-r, ir]) \times \mathcal{U}), r). \tag{4.15}$$

Understanding these equations involves the same arguments that have been given in the previous subsection and will not be repeated here. It remains to compute the reachable states of the lifted system, which will be done in the next subsection. Then, equation (4.14) gives $\mathcal{R}(\mathcal{I}, [0, r])$ which can be plugged into Equation (4.15) and gives $\mathcal{R}(\mathcal{I}, [r, 2r])$, which can be plugged again into the same equation and gives the reachable states for $[2r, 3r]$ and so on, until we reach the desired time horizon.

One common way to specify the initial region $\mathcal{I}$ and also the inputs $\mathcal{U}$ is to use an interval. Each interval $\mathcal{I} = [\underline{a}_1, \overline{a}_1] \times \ldots \times [\underline{a}_n, \overline{a}_n]$ is also a zonotope which can be written as

$$\mathcal{I} = \left(\begin{array}{c|ccc} \frac{\overline{a}_1 + \underline{a}_1}{2} & \frac{\overline{a}_1 - \underline{a}_1}{2} & & \\ \vdots & & \ddots & \\ \frac{\overline{a}_n + \underline{a}_n}{2} & & & \frac{\overline{a}_1 - \underline{a}_1}{2} \end{array} \right)$$

where except for the first column, the matrix is diagonal, empty entries are 0. Hence, the implementation starts by transforming $\mathcal{I}$ and $\mathcal{U}$ into a zonotope.

Given two zonotopes $Z_1 = (c_1, g_1^1, \ldots, g_1^k)$ and $Z_2 = (c_2, g_2^1, \ldots, g_2^k)$ the set $Z_1 \times Z_2$ can be represented by

$$Z_1 \times Z_2 = \left(\begin{array}{c|ccc|ccc} c_1 & g_1^1 & \ldots & g_1^k & & 0 & \\ \hline c_2 & & 0 & & g_2^1 & \ldots & g_2^k \end{array} \right).$$

We use this formula to compute $\mathcal{I} \times \mathcal{U}$ (see Equation (4.14)), after having transformed $\mathcal{I}$ and $\mathcal{U}$ into zonotopes if necessary.

Next, we have to compute $\mathcal{R}'(Z, [0, r])$ for a given zonotope $Z = (c, g_1, \ldots, g_k)$. According to Equation (4.4) we first need the convex hull of Z and $e^{Ar}Z$.

If Z is simply an interval[1] and r is sufficiently small, a closer zonotope overapproximation can be obtained by computing the interval hull[2] of Z and $e^{Ar}Z$:

$$IH(Z, e^{Ar}Z) = [c_1 - \sum_{i=1}^{k} |g_i^1|, c_1 + \sum_{i=1}^{k} |g_i^1|] \times \ldots \times [c_k - \sum_{i=1}^{k} |g_i^k|, c_k + \sum_{i=1}^{k} |g_i^k|]$$

and transforming it into a zonotope Z_H. The resulting zonotope Z_H has only n generators. Back to Equation (4.4), we have to compute the Minkowski sum of Z_H and $\{x \mid \|x\|_v \leq \alpha\}$. Therefore, we have to agree on a vector norm $\|\cdot\|_v$. If using zonotopes, it is handy to use the infinity norm $\|x\|_\infty := \max_i |x_i|$. In that case

$$\{x \mid \|x\|_\infty \leq \alpha\} \quad = \quad [-\alpha, \alpha] \times \ldots \times [-\alpha, \alpha],$$

is an interval (with α as in Equation (4.4)) and thus a zonotope.

Altogether, we have a zonotope overapproximation of $\mathcal{R}'(Z, [0, r])$.

The projection $\pi_n(Z)$ is obtained by applying π_n to the center and each generator of Z. The matrix representation of $\pi_n(Z)$ is equal to the first n rows of the matrix representation of Z. In general, the projection can produce generators that have all entries equal 0. These generators can be deleted to reduce the size of the zonotope.

The number of generators increases in each iteration by the number of inputs, which is due to the operation $\mathcal{R}(\mathcal{I}, [ir - r, r]) \times \mathcal{U}$. Note that the assumption of stepwise constant inputs gives a particular advantage: In Equation (4.7) the union over all admissible inputs u is usually a full-dimensional set, whereas its counterpart in the case of stepwise constant inputs, namely $A^{-1}(e^{Ar} - I)B\mathcal{U}$ (see Equation (4.8)) is of at most the dimension of $\mathcal{U}$. Hence, the number of generators added in each iteration step can be significantly smaller if we can assume stepwise constant inputs.

[1] If $\mathcal{I}$ and $\mathcal{U}$ are both intervals then so is $Z = \mathcal{I} \times \mathcal{U}$.

[2] i.e. a tight interval overapproximation

4.3 Verification of Platoon Behavior

In this section we show the verification of the platoon behavior using the reachability analysis, which is described in the last sections. If there are no communication faults between the members of the platoon, the system can be modeled as a continuous one. Occurrences of possible failures in the communication should, however, be considered in the model. The system then has a hybrid nature, due to the discrete events, and the control strategies should then be investigated by means of the reachability analysis of hybrid systems. The goal of such a verification is to guarantee both short and safe distances between the platoon vehicles. The linear system by which we want to investigate its behavior consists of a platoon of one leader and three following vehicles.

4.3.1 Continuous System Verification

Taking into account the notations which are already explained in 3.3.1, the dynamics of the whole platoon can be summarized in a state-space model. For the case of three following vehicles the system comprises nine states $x^T = (e_1, \dot{e_1}, a_1, e_2, \dot{e_2},\ a_2, e_3, \dot{e_3}, a_3) \in \mathbb{R}^{1\times 9}$. The model and the longitudinal controller of this platoon are derived from the controller proposed in [68]. The settings of the state feedback control are obtained from the optimization methods described in [68][BMMH$^+$11] which is out of the scope of this thesis. The leading vehicle's acceleration a_L enters the dynamics as a disturbance and thus leads to the fact that the controlled dynamics of the whole platoon follows the following linear state-space description:

$$\dot{x} = Ax + Ba_L. \tag{4.16}$$

where $A \in \mathbb{R}^{9\times 9}$ and $B \in \mathbb{R}^{9\times 1}$ are constant matrices, $a_L \in \mathbb{R}$ is the input to the system and bounded by $a_L \in [\underline{a}_L, \overline{a}_L]$. For this model the matrix $B = [0, 1, 0, 0, ...]^T$, where just the second element is equal to one and otherwise zeros, we get the following equation:

$$Ba_L = u, \quad u \in \{0\} \times [\underline{a}_L, \overline{a}_L] \times \{0\} \times \ldots \times \{0\} \tag{4.17}$$

By replacing this equation in 4.16 we get the final equation

$$\dot{x} = Ax + u, \quad u \in \{0\} \times [\underline{a}_L, \overline{a}_L] \times \{0\} \times \ldots \times \{0\}, \tag{4.18}$$

which is similar to the Equation 4.9.

For the safety analysis of this system, the method which is already explained in Section 4.2.1, is used. For the verification of such a system, different values could affect the accuracy of the verification process, such as the initial set I, the acceleration range of a_L, the time step r and the time horizon T. Therefore many experiments were made for testing the behavior of the platoon at these different conditions (for more details please see the work of [BMMH$^+$11]).

Our contribution here is to observe and proof the absence of collisions between the platoon trucks. In addition, the verification should also include the check of whether the spacing errors e_1, e_2, e_3 between the vehicles remain small and do not exceed the safety constraints. Thus, the different ranges of system variables, which guarantee the safety of the platoon under these conditions, should be determined. One critical case that faces the platoon is when the platoon's leader decelerates with different values. An example of such an extreme decelerations can be found in emergency braking. These maximum brake values depend on the specific vehicle, tire and road conditions. Here we assume maximum values during braking of $-9m/s^2$.

The results show that the gaps e_1, e_2 and e_3 attenuate down the platoon and do not exceed $16m$ if we analyze the system for $4s$ with leader acceleration varying between $-9m/s^2 < a_L < 0m/s^2$ for a time horizon of $T = 4s$ and beginning with an initial set closer to the origin. After $15s$ the maximum error amplitudes for e_1, e_2 and e_3 remain constant and the largest spacing error e_1, does not exceed $26m$. In this test we could assure the safety of the platoon in the case of abrupt braking if the reference distance was taken larger than $26m$, which indicates that the investigated controller preserves both the stability and safety criteria under uncertain inputs with short spacing errors.

4.3.2 Hybrid System Verification

Communication between distributed systems makes additional information available that could not be determined by a single subsystem. The use of this information can essentially improve the control of distributed systems such as an automated platoon. However, the occurrence of possible failures in the communication system has to be considered within the investigation and design of an automation concept. With communication breakdowns the platoon behavior is no longer just a dynamical system. These faults force the controller to act differently from the nominal state forming other continuous states. The nature of the system changes to a hybrid one, where these breakdowns trigger the discrete switches from one continuous state to another. Therefore we used hybrid systems in order to model our system and to catch the problems that can affect the performance in the case of a loss of communications.

For modeling the platoon with the communication faults, we start from Equation 4.18 which describes the dynamics of the controlled platoon with the assumption of full communication. In the case of communication loss the matrix A which describes the controller structure will be changed. The notation A_{i-j} is used to describe the new matrices in case the vehicle with index i gets no information via communication from vehicle with index j.

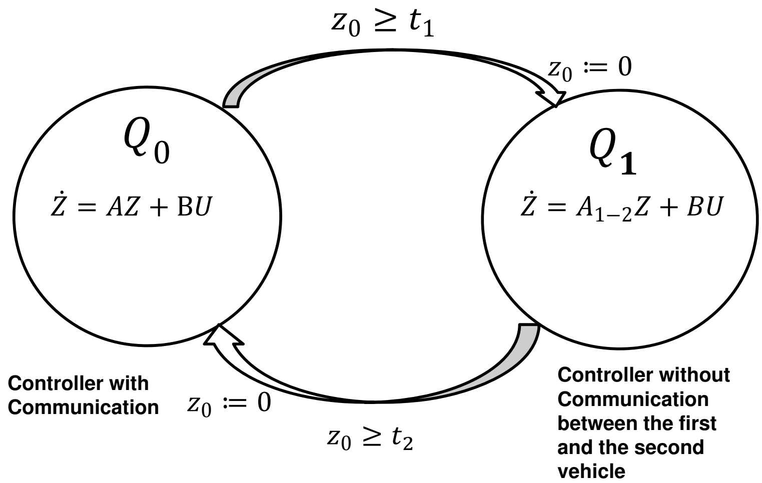
In order to regulate the switch between the different dynamic locations, we extended our state-space model to include a time clock. The idea of using the time clock for switching between different locations when an error happens was introduced in the works of Ben Makhlouf et al. [16]. The new state-space vector Z is defined as $[z_0, X]$ where z_0 is the clock variable which applies to the equation ($\dot{z_0} = 0$). The nominal state-space model is then described as $\dot{Z} = AZ + u$ and the other locations, where there are communication losses, are described in the equation $\dot{Z} = A_{i-j}Z + u$.

Examples of the resulting hybrid automaton can be seen in Figure 4.3. In Figure 4.3(a) the model describes the case when the communication between the first and the second vehicle is lost for time t_1 and then recovered after time t_2. We also consider the worst case scenarios when the communication between all platoon members is lost. This is denoted by the matrix A_{nc} and can be seen in Figure 4.3(b). Please note that in this model we do not need to deal with the intersection of the reachable sets with the guard condition, since we take all the resulting sets as initial set for the next location. The reset function in this model is always $R = f(Q) = 1$.

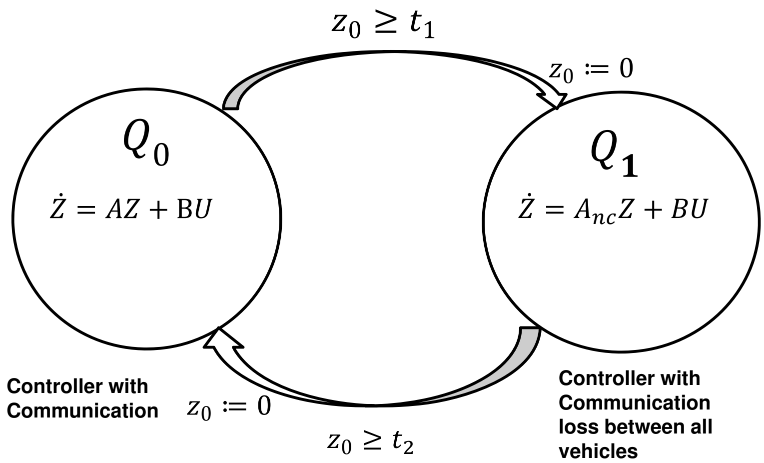
Some experiments were made with different values for the system input which is the acceleration of the leader. Some of these results were presented in our work in [BMDK12] where the verification method using the reachability analysis is also depicted. A summery of these experiments is given as follows:

1. Permanent loss of communication: In this scenario we tested the platoon behavior in the worst case if there is a permanent loss of communication between two vehicles or more. The time t_1 is chosen to be as big as no other switch is possible. With an acceleration of range $[-3, 1]m/s^2$ the maximum spacing errors were noted as $e_1 = 9m, e_2 = 9m, e_3 = 4m$. We also tested a bigger range of leader acceleration $[-9, 1]m/s2$ which represents some real acceleration ranges. The results show that the values of maximum spacing errors do not exceed $e_1 = 25m, e_2 = 25m, e_3 = 8m$. These values decrease as expected with the maximum deceleration value of the leader. From these values the minimum safe spacing between vehicles should be considered.

2. Time-limited communication loss: For switchings between full communication and total breakdown other experiments were made in order to test a periodic loss of the communication. In these experiments not only investigating the minimum safe space but also checking the stability of the controller is important. The time t_1 is equal to 20 seconds. In spite of the switching, we noticed that the controller was immune to communication errors and the same range of spacing errors remains for this set of experiment parameters.



(a) Hybrid Automaton: Example of loss in communication between two vehicles



(b) Hybrid Automaton: Example of full loss in communication between all vehicles

Figure 4.3: Modeling the platoon as a hybrid automaton

5 A Platoon Crossing an Intersection: Experimental Study

Intersection problems arise not only in urban areas but with the rapid growth of traffic volume, they also occur on rural highways. In Section 2.3 the importance of intelligent intersections as a method for achieving safe and efficient crossing of intersections was shown.

This chapter points out some methods and some experiments to answer the question: How can intersection driving be optimized in order to make a platoon of autonomous vehicles travel efficiently and safely? Vehicles should never collide in the intersection and the system should be safe even if there are communication problems in the intersection area. In order to investigate the behavior of the platoon while crossing an intersection we extended the platform described in Chapter 3 with an intersection management system. This system should prevent collisions in the modeled intersection area in different complex scenarios.

Since traffic lights are an important part of the traffic network we developed a traffic light system that allows safe and efficient crossing of the platoon through the intersection. The traffic light system is able to respond to the current traffic situation. In this chapter, the the extensions of the platform to make the platoon able of crossing an intersection is explained. The description of the system and its requirements will be determined and the results are shown. Some parts of this chapter were introduced in an earlier publication [DBMK12].

External Contributions

The modification of the positioning system to a one-dimensional positioning system in Subsection 5.2.3 and the hardware implementation of the collision detection in Section 5.4 was part of the bachelor thesis of Ali Tarzan [Tar12]. In Subsection 5.4 some of the definitions of the strategies and the hardware implementation of some test cases for a signalized intersection were part of the bachelor thesis of Lars Tholen [Tho12].

5.1 Platoon of Vehicles Crossing an Intersection

It is common, especially for manually driven trucks, to travel and arrive at intersections in platoon formations. In Subsections 2.3.2 and 2.3.3 a number of project initiatives and research groups which investigate and develop methods for intersection optimization and safety were introduced. In general these developed intersection assistant systems do not take the presence of such a platoon formation as a unity into account. The focus of such systems lies on managing the crossing of one vehicle, making the control methods and intersection management components fixed for one vehicle. Such a kind of systems can force an autonomous platoon to an unnecessarily stop or it even leads to the loss of the platoon formation. As a consequence the platoons are delayed or split and the intersection signalization becomes inefficient.

5.1.1 State of the Art

In Section 2.3, a number of projects and methods used to control the movement of a single vehicle while crossing the intersection were already discussed. Since in our work we are interested in how an autonomous platoon of vehicles crosses an intersection safely and efficiently, we introduce some other works with two different perspectives:

- How non-automated vehicles driving in a platoon cross an intersection optimally:

  The research in the field of intersection management has recently moved one step further to deal not only with one vehicle but with a number of vehicles approaching the intersection. These vehicles build a communication network between all participants in the intersection area (other vehicles and the infrastructure unit) to improve traffic control at intersections. For example, the work of Jiang et al. [49] proposed a solution on a rural corridor in Indiana. Therein a platoon-based adaptive algorithm was suggested for traffic signal timing. Platoon size, the average headway of vehicles within the platoon, the platoon speed, and the inter-arrival time between consecutive platoons are hence collected and used for control purposes.

  In order to achieve an optimum crossing of such non-automated platoons, many works in this field suggested that the vehicles lose their control over the vehicle once they enter the intersection area and they apply to the rules of the intelligent intersection. In the work of Neuendorf and Bruns [72], it was suggested the vehicles should form an automated platoon. The authors of this work proposed a platoon controller to control the vehicles through the intersection.

- How automated vehicles cross the intersection:

  Once autonomous vehicles become popular, interactions among multiple vehicles will be possible and the movement of these vehicles can be controlled while they cross the intersection. Dresner and Stone introduced in their works [28][29] an Autonomous Intersection Management (AIM) where they used a multiagent system approach for traffic intersection management. Here, the vehicle, which is an agent, reserves a time slot in the intersection in order to have a collision-free trajectory. This time slot is given by the agent manger which controls the agents. For testing the management of autonomous intersections, a number of works used robots as a prototype for such systems. Examples of these works can be found in the work of Kolodko and Vlacic [53] where they introduced a platform of testing the scenario of autonomous vehicles of mobile robots passing through an unsignalized intersection. To perform certain maneuvers an event-based control algorithm was used. The authors of Kolodko and Vlacic [53] used mini robots as a prototype and tested a distributed algorithm to avoid collision. For localization the authors used landmarks on the road. Each robot vehicle knows via communication the direction of the other robots and based on this information the robot can determine the collision-free path through the intersection.

5.1.2 Our Approach

As already explained before, systems do exist which manage the arriving of automated vehicles at an intersection so that vehicles are able to deal with any potential of collisions between them. However, these systems concentrate on each single vehicle and do not take the approach of a platoon of automated vehicles into account. The argument that the platoon could simply be treated as a number of individual autonomous vehicles is not applicable since the vehicles of the platoon should apply to the control rules of the total platoon controller. A management system developed for single vehicles to prevent collisions could not reduce or eliminate the risk of collisions of the automated platoon. In this case an efficient solutions should be chosen for many different scenarios and for specific cases.

Criteria, such as the total waiting time of all vehicles, for example, should be considered. If systems, meant for controlling single autonomous vehicles are used in the autonomous platoon case, this could lead to long waiting times for members of the platoon due to unnecessary stops or splits in the platoon. In addition, some problems could occur if splitting the platoon is necessary to be performed, and the scenario of merging the platoon back together could be difficult to achieve in systems which do not take this case into consideration. So while it makes sense from the perspective of the intersection management that each vehicle of the platoon is defined as an individual item, it is advantageous from the viewpoint of the platoon, if it is treated as a group.

The difference to the other works where robots are used for intersection management is that in this work the platoon is seen as an entity or a group of robots that want to manage the intersection. These robot vehicles approaching the intersection apply only to the control rules of a platoon where autonomous vehicles follow the leader. Therefore we provided the vehicles of the platoon with a platoon management system to adjust the behavior of the platoon to the intersection's conditions. Information about the intersection is gotten from an intersection management unit, installed at the intersection. However, the vehicles do not lose their control to this unit.

In our platform we thus divided the functionalities of a vehicle approaching the intersection into two main categories. The first one is responsible for dynamics and the actual movement of the vehicles in the intersection. This one is already explained in Section 3.3. The other system has the role of finding the right modus for the existing scenarios and to give the instruction to the platoon vehicles (for example: stop, pass, and even split).

In order to implement this functionality, the platoon should get extra information about the intersection. The necessary data is:

1. The platoon should know its position according to the intersection. Therefore an indoor positioning system, which is mounted on each vehicle, should be integrated in the platform.

2. The platoon should not just exchange the information between its members, but it also should exchange its information with other vehicles around the intersection area and with an intersection management unit. This communication allows the platoon to have more information about the other participants at the intersection and the status of the intersection.

The platoon should be able to cross two kinds of intersections: intersections with and without traffic lights. For intersections without traffic lights, the platoon should estimate the situation at the intersection with help of communication with intersection unit sensors in order to stop or even split the platoon if necessary. The question with intersections equipped with traffic lights can be simply formulated as finding the optimal speed (or the range of speed) for the leader in order to let all the vehicles in the platoon go through the green phase of the traffic light and without collision between its items; assuming that the leader vehicle of the platoon can be automatically controlled. This has many advantages such as increasing the throughput of the road and consequently has advantages for the environment and for safety.

The next sections in this chapter describe the implementation of an indoor positioning system, the implementation of an intersection management unit, and how they have been integrated in our platoon platform.

5.2 Description of the Positioning System

Knowing the position of the vehicles in our platform is helpful to perform more complex maneuvers than ordinary platoon driving. Information about the position of different vehicles inside the intersection area is very important for the intersection management. In addition, knowing the position of the vehicles allows us to use the platform for further scenarios such as a vehicle joining or leaving the platoon. In this section the hardware implementation of an indoor positioning system, which we integrate in our platform, will be introduced. This system is able to provide the position to an arbitrary number of objects in a test field independently. By using the sensor data of the vehicles the accuracy of the position measurements was increased. The evaluation of the system was done for stationary objects and moving vehicles. In addition, emergency platoon driving depending only on the vehicle's positions, not on the distance sensors, was tested.

5.2.1 Methods of Indoor Positioning

In Subsection 2.1.2, a number of methods used for computing the position of full-scale vehicles were reviewed. Due to the difference in the environments, other systems and technologies are used for finding the position and tracking the objects in the indoor environment. In this subsection some technologies used for finding the indoor positioning will be presented.

Indoor location systems have won more interest in many applications designed for mobile navigation systems over the past few years. Depending on the applied technology, a variety of developed positioning systems already exists. Computer vision systems have been also used for different tracking applications ([104],[54]). The work of Yilmaz et al. [105] gives a good survey of the different methods of how image processing and computer vision is used in object tracking. The need of a fast recognition of the searched object and the need of fast image processing time are mainly the difficulties that restrict the use of such systems.

Some of the research groups use the strength of the radio frequency signal as an indicator of the distance between the receiver and a number of transmitters with a known-position. The researchers who use this method have recently concentrated on finding the position in areas where typical wireless data network are installed. They use the benefit of existing data networks (for example IEEE 802.11 network) and use the WLAN location determination technique network. Examples of such systems can be found in the works of Youssef et al. [106] and Ladd et al. [56]). Although this method is cost efficient since no extra construction is required, it lacks accuracy (about 3 m).

In the work of Want et al. [103] a network of infrared receivers was spread around the indoor environment. A so-called an 'Aktive Badge' permanently sends a signal which is received by the network and consequently enables the determination of the object location. Using infrared signals , however, has a drawback of measuring the position in dead spots. Dead spots are some areas in the room where the signal is not reached. Another well used technique is to send two signals with two different natures. The fast signal activates a timer until the reception of the slower one. The time difference is then used to measure the distance. Using ultrasound sensors as a timing signal has shown promising results. A well known example of such systems is the Cricket location system [91].

Other research groups that use radio technologies are interested in using the latest technologies of Radio Frequency Identification (RFID) in object localization, such as the work of Chawla et al. [23]. In Jing and Yang [50] an algorithm based on SLAM (Simultaneous Localization and Mapping) was used to calculate the position of the mobile robot. The plane area of the floor has an arrangement of RFID tags and the bottom of this robot has an RFID reader which reads these tags in order to know its absolute position.

5.2.2 Positioning System Architecture

For some intelligent driving operations, such as intersection crossing for autonomous vehicles, positioning systems play an important role. This subsection introduces the indoor positioning system which we have developed for our platform. For the choice of a suitable positioning system and its related hardware for our scaled models, many factors such as cost, sufficient accuracy, and the adequate update rate of the computed position data have been taken into consideration. Furthermore, the position information should be provided to an arbitrary number of vehicles and the positioning system for our platform should allow the vehicles to calculate their position locally, not by a central processing unit.

In the last subsection a number of the technologies used in indoor positioning systems are reviewed. The usual procedure of computer vision systems is that a camera system tracks an object, which means that the requirement of decentral detection is not fulfilled. Even by putting a camera on each truck to detect fixed points on the testing track, factors such as the need of a line of sight and the need of expensive high-resolution cameras restricted the ability of using such systems in our platform. Infrared-based systems are excluded due to their short range and WLAN-localization based systems do not provide sufficient accuracy for platoon driving.

Therefore we chose to implement an ultrasound-based positioning system. Two approaches for realizing the positioning system exist:

- The object which needs to be localized sends out the signals. Some sensors installed on the infrastructure environment, such as on the walls of a room, receive these signals and use it to calculate the object's position. These systems thus calculate the position of an 'active' object. The receive sensors are called 'listeners' and they wait to receive a signal. The disadvantage of such a system is the degradation of the performance with the increasing number of tracked objects.

- In contrast to the above mentioned system, the 'passive' localization method is to make the tracked object a listener for other signals sent by infrastructure transmitters. The performance of the system is therefore independent of the tracked objects.

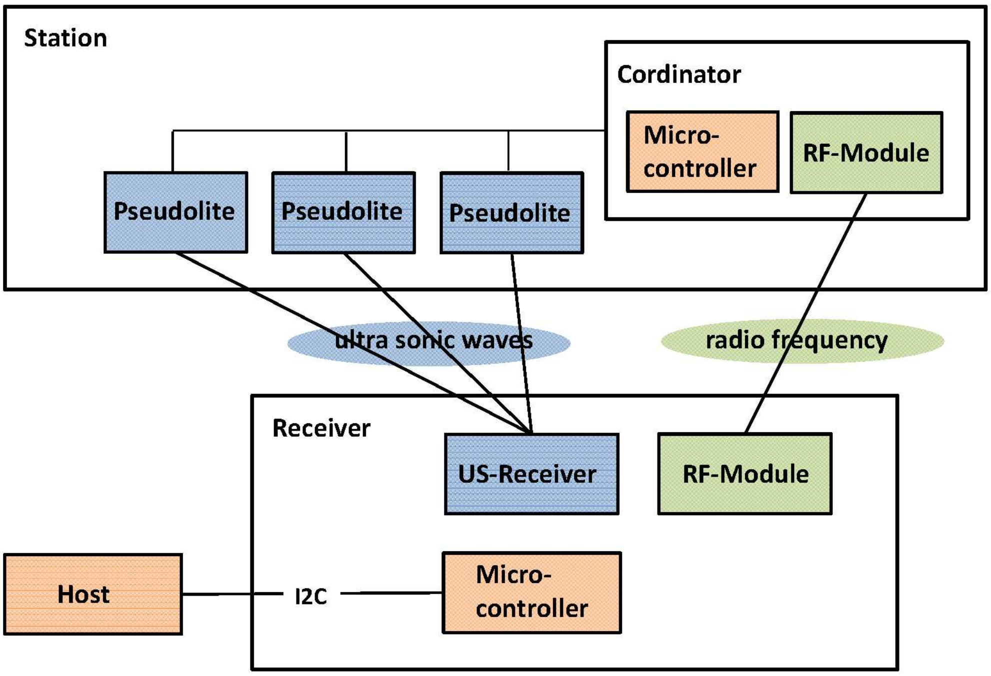
The goal of our positioning system is to determine the position of each truck locally and independently from each other. A passive localization system meets this requirement and keeps the costs at a very low level. Thereby a given number of pseudo-satellites (pseudolites) send out signals and an arbitrary number of receivers can calculate their positions from these signals without disturbing each other. In order to reduce the complexity of the position calculation it is sufficient for our purpose to get the position in a plane, by performing the trilateration algorithm with three pseudolites.

The hardware architecture is divided into the station and the receivers. The station sends out radio signals and ultrasonic waves periodically. The receiver, mounted on the top of each vehicle, gets these signals, calculates the position and provides it to the vehicle. The block diagram of the hardware architecture of the positioning system can be seen in Figure 5.1.

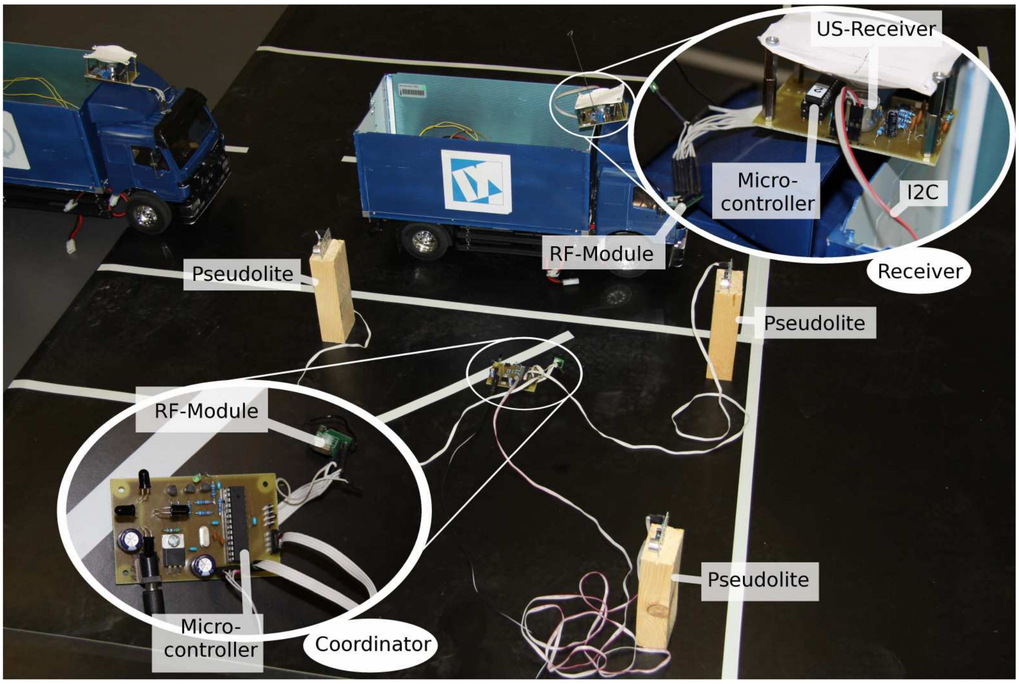
Station

The station consists of two elements

- Pseudolites: These are ultrasonic transmitters which are connected to the coordinator.



(a) The architecture



(b) The hardware

Figure 5.1: Hardware and architecture of the positioning system

- Coordinator: This consists of an ATMega88 microcontroller and a 868 MHz radio sender-module. The microcontroller controls and coordinates the transmission of signals from the station.

At the same time the station send two signals; An ultrasonic wave from one of the pseudolites and a radio signal. After a certain time interval the station transmits an ultrasonic from another pseudolite and a radio signal simultaneously. The station transmits these signals periodically. In this study, the time interval between two transmissions has been set to 150ms. This time could not be reduced to a shorter delay, since otherwise echoes would be received, which has a significant effect on the measuring accuracy. Taking this choice, the cycle time of the station transmission of signals from all the three pseudolites is 450 ms, making the update rate at the receiver about 2 Hz.

Receiver

On the receiver side a microcontroller ATMega168, a 868 MHz receiver-module and an ultrasonic receiver-module are involved to calculate the position. Once the radio signal is received, the microcontroller starts to measure the time untill the arrival of the ultrasound signal. This time difference measurement is used to calculate the distance to the pseudolite which has sent the signal. The time between the transmission and the reception of the signal is called the trip-time. In the indoor environment the trip-time of the radio waves, which trigger the time measuring, can be ignored. Thus, the trip-time from the station to the receiver t_{trip} is assumed to be the trip-time of the ultrasonic wave in the air. By this simplification the distance to the satellite D_{sat} can be calculated according to the following equation:

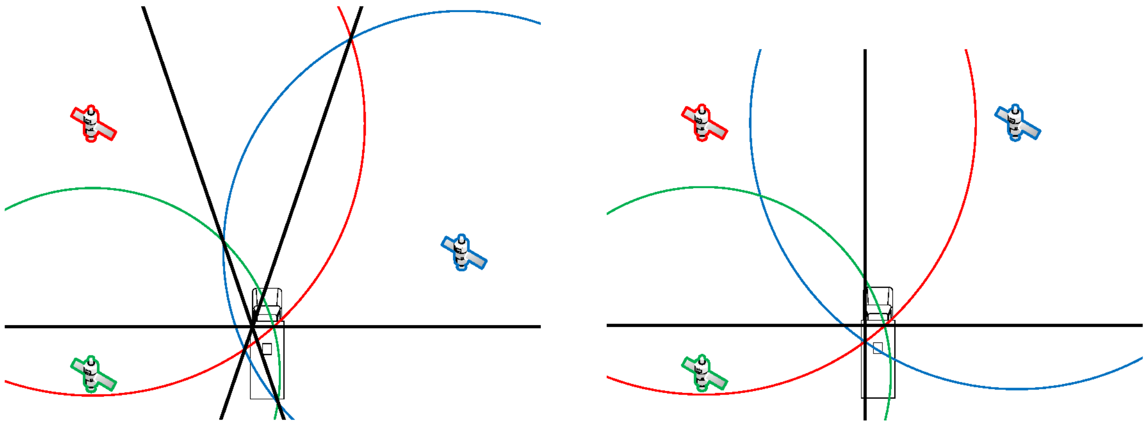
$$D_{sat}[cm] = \frac{c_{air}[cm/s]}{t_{trip}[s]}$$

where the speed of sound in air c_{air} is known. The time needed by all the station's receivers to send their signal once is called the cycle time of the station. After the end of the station's cycle time, the receiver's distance to each one of the pseudolites is used by the microcontroller in the receiver to calculate the position. This position is then sent to the hardware of the truck (the host) via the I^2C bus of the truck.

5.2.3 Algorithm

The position of the truck can be calculated by a trilateration algorithm. The distance to each of the three pseudolites is known by the receiver as explained before. By computing the intersection point of three circles around the pseudolites (each circle has a diameter equal to: $2 * D_{sat_i}$), the exact position of the truck in the plane can be determined. Since measuring the distance to the pseudolites is not always exact, it is almost impossible to get the same intersection point of each pair of the circles. Therefore, the position of the truck, as shown in Figure 5.2(a), is calculated as follows:

1. For each pair of circles, the two intersection points are calculated.



(a) Ordinary positioning computation algorithm.

(b) Fast positioning using only two intersection lines

Figure 5.2: Two algorithms for computing the position

2. Through each pair of points a straight line (intersection line) is determined.

3. The intersection points of all the three resulting intersection lines are computed.

4. The delivered position is then the mean value of those intersection points.

By an appropriate arranging of the pseudolites in the plane, in which they form a right angle with each other as shown in Figure 5.2(b), the position computation can be done more efficiently. In this case, two of the intersection lines are parallel to the axis of the coordinate system. This therefore simplifies the above algorithm by calculating the intersection point of these two lines.

Optimization Methods of the Trilateration Algorithm

Using the method of pseudolites, the calculation of the position of moving objects is more inaccurate than calculating the position of stationary ones. In order to improve the accuracy of the position determination while the truck is in movement we applied different optimization techniques:

1. As mentioned before, the time interval between two consecutive pseudolites is set to 50 ms. In order not to wait for the other signal of the same pseudolite after 150 ms, the receiver uses the new data send from the first pseudolite directly in the next calculation without waiting for the reception of signals from the other pseudolites. This does not increase the total cycle time as the information of one pseudolite will be used again. Our simulation (as explained in 5.2.3) showed a significant accuracy gain with this optimization.

2. Further filtering and optimization is done by the MPC555 host controller of the truck by using the in-vehicle sensors. We implemented a simple approach where

the driven distance, measured by the magnetic angle encoder, is used to check if the new position is inside a possible radius depending on the actual velocity. On the one hand, this avoids spikes in the position measurement caused by erroneous ultrasonic measurements, but on the other hand it reduces the update rate.

3. The trilateration algorithm assumes that within a station cycle the object should ideally remain unmoved, since the measurements of the distance to the pseudolites have to be done one after another untill the end of the station cycle. Due to the fact that there is a defined delay until the next position that calculation can be performed, it is therefore possible for a fast-moving receiver that data can be discarded if the trilateration algorithm cannot provide a plausible result.

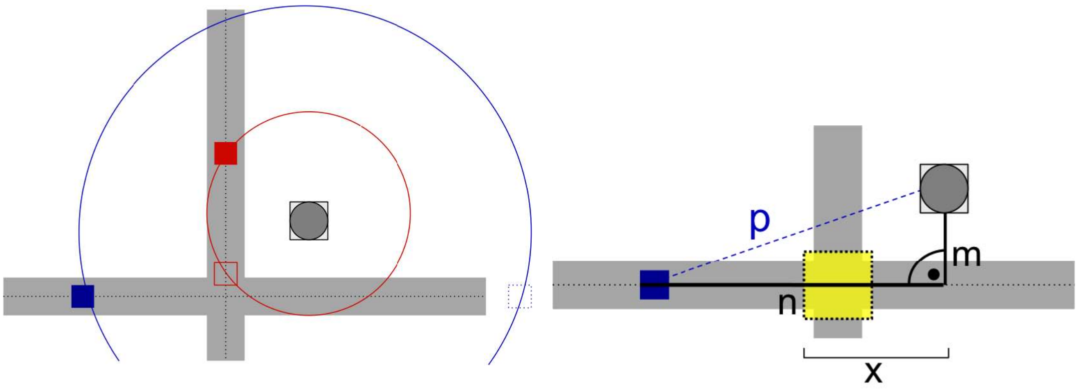
   Since the determination of position is very important for performing the autonomous driving it is essential to have a redundant sources of information, so the platoon is still able to calculate its position data. If an initial position is available, we use the speed information again in order to estimate how far the vehicle drove in the meantime. If the last position y is known at the time t, along with the new velocity V and the new time clock, the new position is calculated approximately as follows: $y_new = y + (t - t_{act}) * V$. In this approach it should be assumed that the vehicles are driving straight without any lateral movement. However, tests have shown that this uncertainty is negligible. To smooth out the position data between the times at which there are current positions, velocity sensors are used in order to estimate the position since the last receipt of data. The estimations are calculated regularly at intervals of δt, where δt is sufficiently small within that time; the speed can thus be regarded as constant.

The proposed positioning system provides the coordinates of the position in the X and Y direction. For the intersection scenario, it can be sufficient to find the relative position of the vehicle to the intersection area. The next subsection discusses the optimization of the positioning system in some cases where the position is needed in one dimension.

One-dimensional Positioning System

Assuming that the vehicle just has longitudinal dynamics it is not necessary to have the position coordinates in two dimensions, instead just the distance to the intersection. Therefore the already explained positioning system can be modified so that instead of three pseudolites, one pseudolite in each direction is used as shown in Figure 5.3. This modification of the positioning system was introduced in the bachelor thesis of Ali Tarzan [Tar12]. Using such a one-dimensional positioning system has the advantage that no trilateration is necessary to determine the position. This means that the accuracy can be increased and better results can be achieved because of the elimination of the trilateration algorithm errors. In addition, there is no need to wait the signals from all of the three satellites to uniquely determine the position. This enables updating the location data more frequently.

There are two options of where to place the pseudolite relative to the intersection. Figure 5.3(a) explains these two options and the considered scenario.



(a) Position of the satellite

(b) Calculating the position of the distance to the satellite

Figure 5.3: One-dimensional positioning system

- The pseudolite is placed between the intersection and the receiver (red rectangle): There are two possible solutions to calculate the position of the vehicle.

- The pseudolite is placed behind the corner farther from the approaching vehicle to the intersection (red rectangle): One of the solutions is excluded and the position is clearly determined. Therefore we chose that in each direction, one pseudolite is positioned in the corner after the intersection.

Figure 5.3(b) shows how the position of the vehicle to the stop line is calculated. The values X and m are available to the vehicle or provided by an intersection management unit as will be explained in Section 5.3. The receiver positioned on the vehicle calculates the distance p. Then the distance n can be calculated from the Pythagorean theorem as $n = \sqrt{p^2 - m^2}$. The desired distance is then $n - x$.

Simulation of the Algorithm

Before building and while implementing the positioning system, several simulations have been performed to estimate the accuracy of the system. The simulation allows for a given position and speed of the vehicle, defined pseudolite positions, and delay times to determine the position as computed by the algorithm. Furthermore, the accuracy of the ultrasonic distance measurements of the pseudolites can be adjusted to get realistic results.

The first optimization method, mentioned before, has been analyzed in the simulation. By using only data of the current pseudolite cycle the position error was around 50 cm for a simulated drive of 60 cm/s. For this simulation the distance measurement error was ignored in order to estimate the accuracy of the algorithm itself. The error could be reduced to 5 cm by using additional data from the first pseudolite of the next cycle. Figure 5.4 shows a simulated drive, starting at position (80, 500) driving along the y-axis

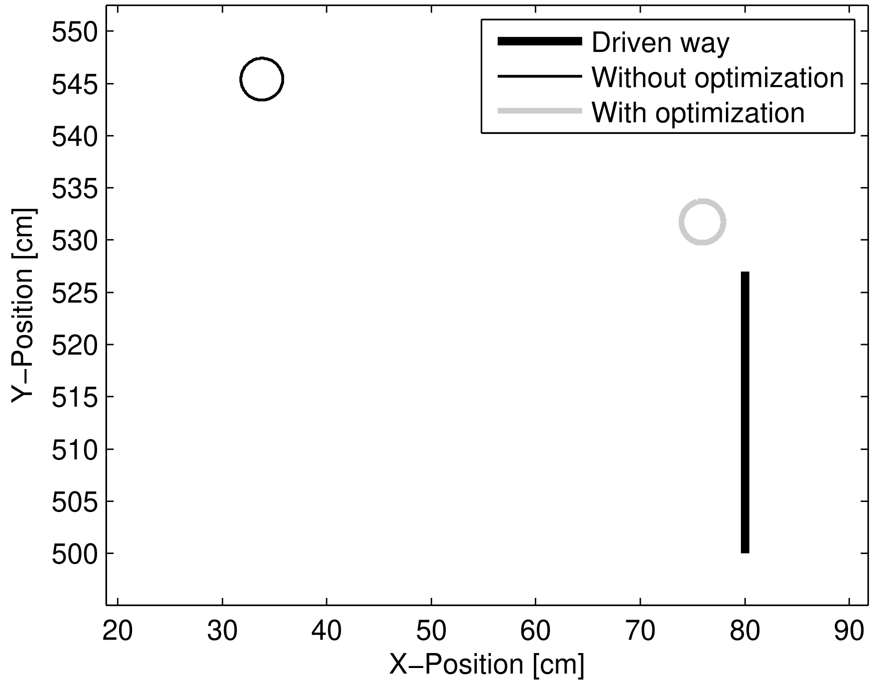


Figure 5.4: Position simulation with an assumed velocity of 60 cm/s

with 60 cm/s. The circles show the calculated position when the vehicle is positioned at (80, 527).

5.2.4 Evaluation and Results

By integrating the positioning system into the platoon platform, we were able to perform more complex scenarios such as crossing the intersection safely. In addition knowing the vehicle position allows even the platoon leader to drive autonomously. This is done by defining a number of points which the leader should reach.

In this subsection the positioning system is evaluated by integrating it into the platform. It also presents some of the experiment results. The positioning system has been tested in different scenarios. This includes the stationary and moving objects as well as testing a platoon controlled by positioning data. The position is represented in two-dimensional coordinates with the two values (x_{value}, y_{value}). In those experiments the pseudolites were placed at the coordinates (0,0), (0, 100) and (100,0). These values are measured in cm.

Stationary Evaluation

In the stationary test, the truck has been placed at a fixed position and the position is computed by the positioning system for a duration of about one minute. The position data are sent via the WiFi module along with the other sensor data to monitor and log the data with our GUI.

Position	Measured Avg.	Std. Deviation	Max. Deviation
(80, 150)	(79.8, 149.7)	1.17 cm	2.71 cm
(80, 210)	(77.1, 212.4)	1.43 cm	3.75 cm
(80, 230)	(75.8, 231.2)	1.88 cm	4.04 cm
(80, 510)	(77.4, 514.9)	4.03 cm	11.45 cm

Table 5.1: Stationary accuracy

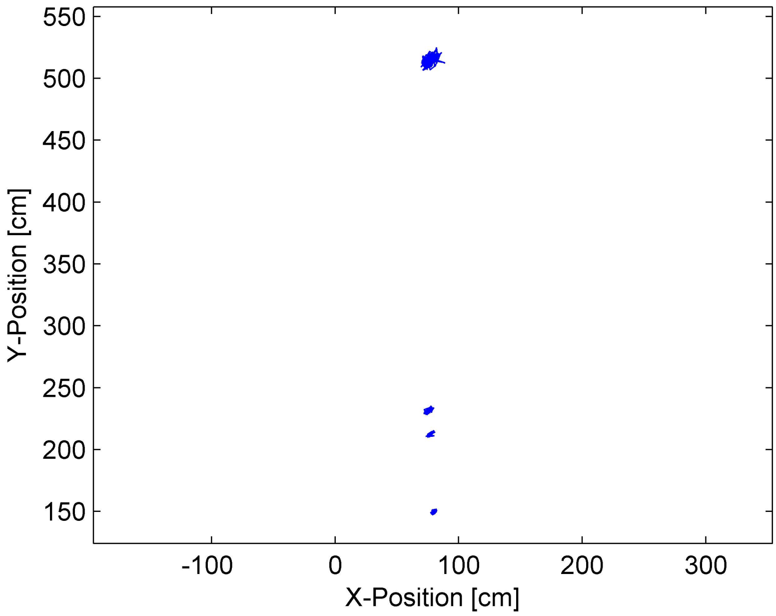
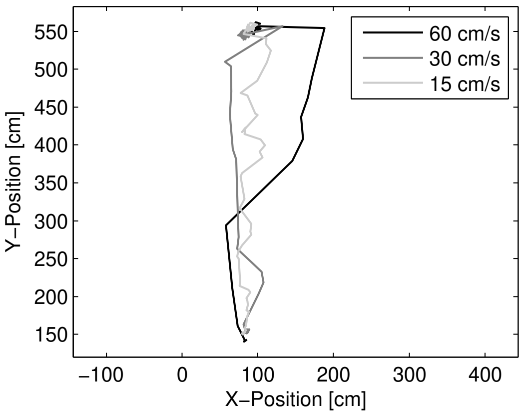


Figure 5.5: Stationary accuracy

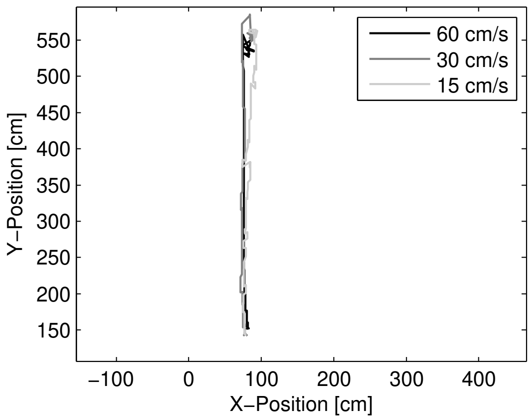
For this experiment a truck was located on different positions (80, 150), (80, 210), (80, 230) and (80, 510). Table 5.1 lists the averages of the calculated positions as well as the standard deviation and maximum deviations in relation to these averages during the experiment. The graphical representation of these measurements is depicted in Figure 5.5.

Tracking Evaluation

As explained before, the transmission delay time between the signals from two pseudolites has a huge impact on the accuracy while the truck is moving. Figure 5.6 shows the accuracy for a straight drive of four meters with different velocities. The vehicle started at position (80, 150) and moved along in the y-direction. The results without optimization show that the deviation from the real position increases with the velocity. These results, shown in Figure 5.6(a), have already been observed in our previous simulations. Figure 5.2.4 shows that the impact of the velocity to the accuracy is reduced a lot using the optimization methods mentioned in Section 5.2.3.



(a) Without optimization



(b) With optimization

Figure 5.6: Tracking accuracy

Platooning Evaluation

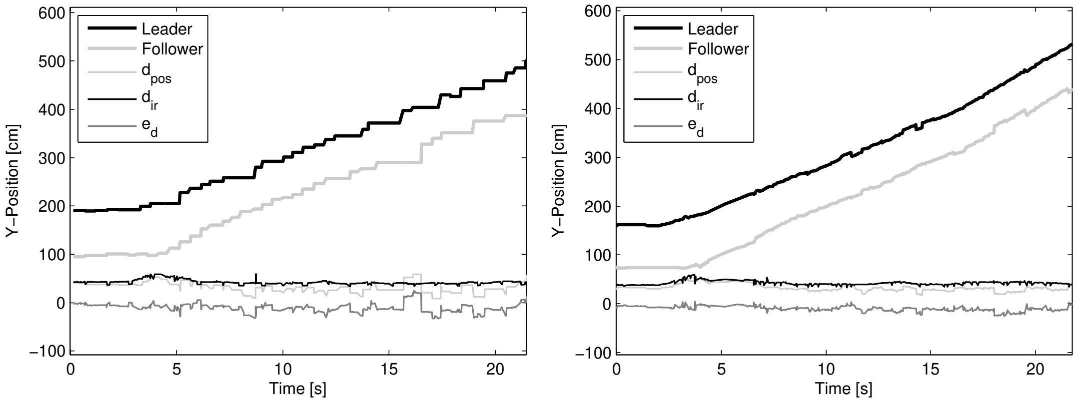
The goal of these experiments is not to replace the vehicle's sensors by the positioning system but to show that the information retrieved by the positioning system is seemly reliable and can be used for further autonomous driving tests. Furthermore, in case of total failure of the vehicle's local sensor network, the position values can be used temporarily until the vehicle enters a safe state, otherwise an emergency stop should be performed. If the control algorithm is able to use only data collected by the positioning system instead of using the local distance sensors to regulate the movement in the platoon without collision even for short time, the hardware of the platform is then considered to have some degree of redundancy. This is needed when driving autonomously.

In the following experiment we have tested the control of just two trucks: a leader and a follower. The controller on the follower should maintain a constant distance of 40 cm to the leader. As in the case of the tracking scenario, this test drive took place along the y-direction. The leader has a constant velocity of 20 cm/s. Based on how the distance between the vehicles is retrieved, we differentiate between two methods:

- d_{ir} is the distance measured by the infrared sensors.
- d_{pos} is the distance calculated by the difference of the two vehicle's positions.

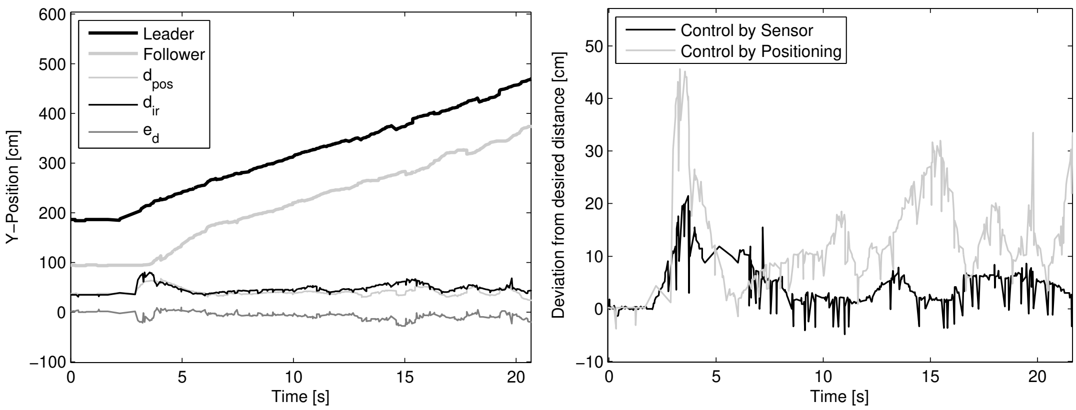
To calculate the distance by using the positioning system, the length of the trucks l has to be taken into account. With the difference of the truck positions Δ_{pos} the distance d_{pos} can be calculated as $d_{pos} = \Delta_{pos} - l$.

In two experiments we differentiated between using either d_{ir} or d_{pos} as a control variable. Figure 5.7 shows the recorded y-coordinates while performing a platoon drive, where the controller of the follower controls its speed according to the measured distance d_{ir}. In addition, the figure shows also the distances d_{pos} and d_{ir} as well as their difference $e_d = d_{pos} - d_{ir}$.



(a) Without optimization (b) Optimization with in-vehicle sensors

Figure 5.7: Platooning with two trucks and tracking of their y-position



(a) Using d_{pos} as a control variable (b) Deviation from desired distance

Figure 5.8: Platooning test using the positioning system

The same experiment has been performed twice to show the improvement using the in-vehicle sensors for the optimization of the algorithm as explained before. In Figure 5.7(b) we can see that e_d is smoother and smaller and thus the calculated distance d_{pos} shows less difference from the distance d_{ir} measured by the infrared sensor.

The results of the experiment for the case of d_{pos} as a control variable are given in Figure 5.8(a). By comparing the error in relation to the desired distance of both experiments (Figure 5.8(b)) we observe, as expected, a bigger deviation of the desired distance with using d_{pos} as a control variable. However, there are no drastic changes in the system's performance.

5.3 Intersection Management

In this section the methods and the hardware used in our design in order to enable the truck platoon in our platform to cross the intersection safely and efficiently are represented. An intersection management unit was added to our platform in order to allow the platoon to get more information about the status of the intersection. This section explains the modifications, done to the software of the platoon vehicles, in order to let the platoon change its modes to apply to intersection driving.

In this study we have made the following assumptions:

- To simplify the problem, we consider a single isolated intersection. The intersection consists of two straight intersecting roads, forming a right angle, where each road is treated as a single lane. This means that we consider a scenario where a platoon of vehicles drives along a single-lane road and approaches an intersection with two lanes crossing each other. A study of a complex network of roads and intersections are not in the scope of this thesis.
- In the intersection area, we have no influence on other vehicles which do not belong to the platoon. Even if the vehicles have the possibility of communication with the platoon, we assume that the right of way is always considered to these vehicles and not to the platoon.
- The intersection could either be regulated by a traffic light (signalized intersection) or not (unsignalized intersection).
- In the intersection scenario, the experiments concentrate on the longitudinal dynamics of the platoon movement, assuming that the lateral components have very little influence (since there is no other lane to change to).

5.3.1 Hardware of the Intersection Management Unit

An intersection management unit was implemented in order to allow the platoon to get more information about the status of the intersection. Although data processing is mostly done by the platoon controllers, the intersection unit controller could sometimes be used for giving advisory instructions, in case the platoon was unable to do that. Figure 5.9

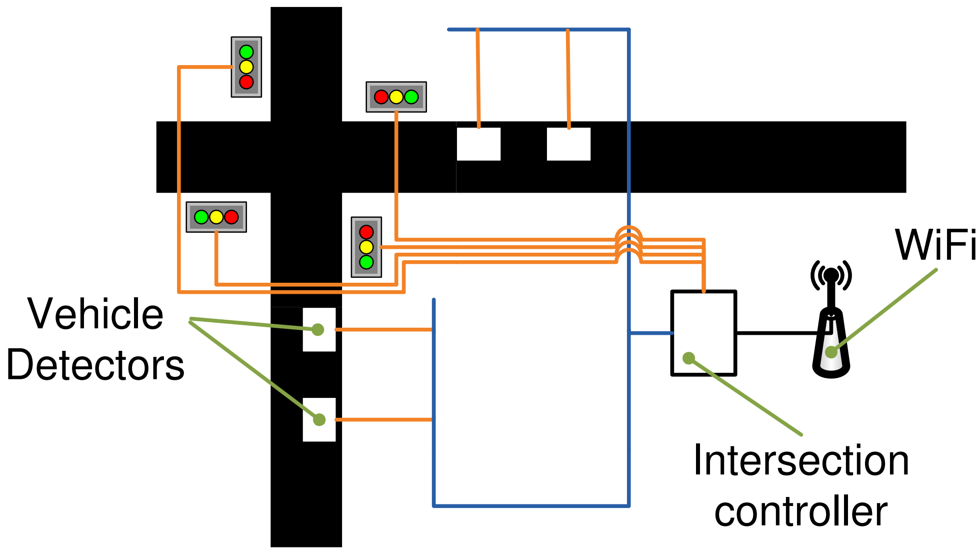


Figure 5.9: Hardware architecture of the intersection management

shows the hardware architecture of the intersection management unit. It consists of roadside sensors, traffic lights and a WiFi module. These elements are connected and controlled by an intersection controller. The controller is an ATmega664 and is part of a development board [92].

For testing signalized intersections, two scaled traffic lights were placed at the intersection. Each traffic light consists of three LEDs to represent the traffic light phases (red, orange and green) and is adjusted to not change to green light simultaneously with the other traffic light. In Figure 5.10 the hardware used for the traffic light is shown.

The road sensors are distance measuring sensors (GP2Y0A02YK) with a detecting distance of 10-80 *cm*. In order to monitor the speed and position of the vehicles in each lane, we have placed two of these sensors along each road at a known distance. Figure 5.11 shows how each pair of sensors was mounted together.

These sensors do not provide any tracking analysis, but rather the position at specific points of the road. The following algorithm is used to calculate the position and velocity of the vehicle, where $IR1$ and $IR2$ are the first and the second infrared measuring distance sensor:

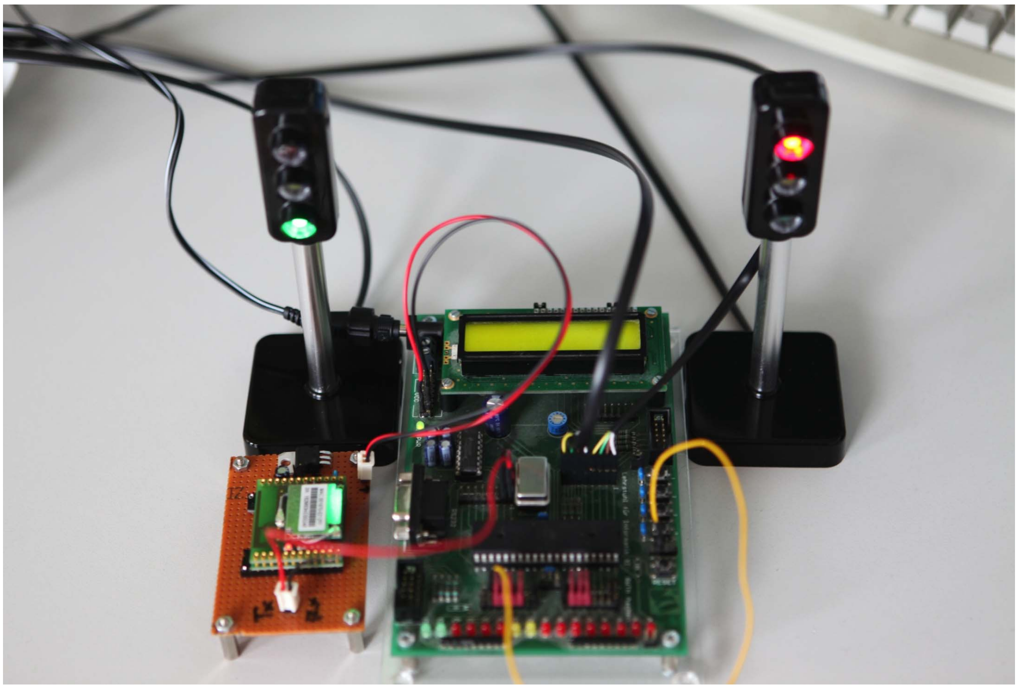


Figure 5.10: Hardware of the traffic light

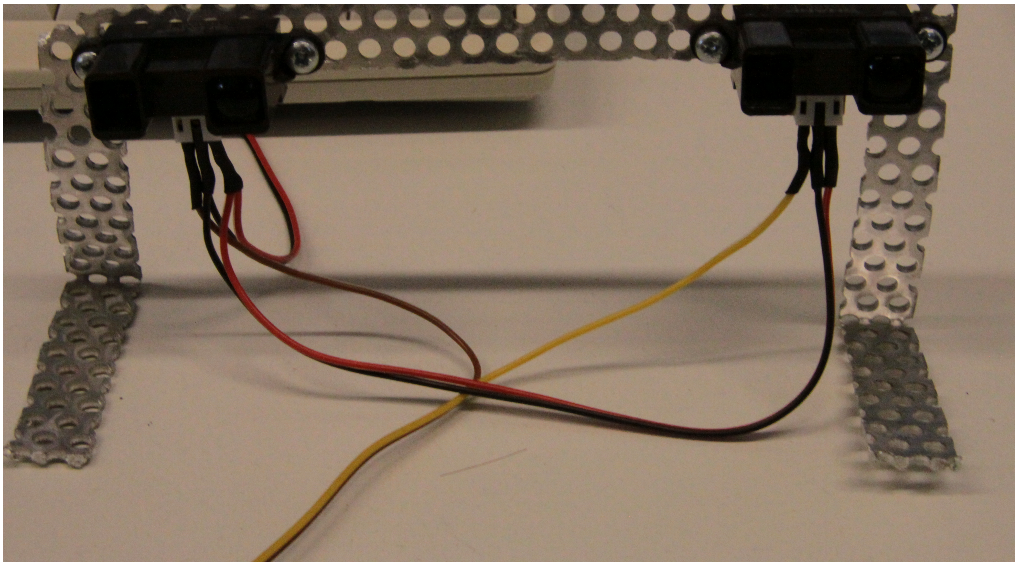


Figure 5.11: Road detectors of the intersection management unit

```
#define Position_IR1
#define Position_IR2
#define Distance_between_Sensors
#define Detect_Threshold

While(Vehicle_not_Detected)
{
if measureIR1<Detect_Threshold
 {
 [TimetoSensor2]=wait(measureIR2<Detect_Threshold);
 Vehicle_pos= Position_IR2;
 Vehicle_speed=Distance_between_Sensors/TimetoSensor2;
 }
}
```

5.3.2 Communication with the Platoon

The intersection management unit is provided with a Wifi module allowing it to share the data with the platoon vehicles. The communication module is an Avisaro wireless module (as the same used in the truck). As explained before the platoon network was designed in order to exchange the information between the members of the platoon. The communication scheme, represented in Figure 3.11, had to be modified to include the presence of a positioning system and an intersection management unit into the platform network.

By introducing the positioning system into our platform, each truck in the platoon should send its position to other members in the communication network. The position information in both x and y directions should be included in the information packet of the truck together with the other sensor data. Because the accelerometer values are not considered in the evaluation of the scenario of a platoon crossing the intersection, we replaced the accelerometer data with the position. Figure 5.12 shows the new content of the message sent via a WiFi module from each vehicle.

Angle front	Distance	Min. Dist.	Speed	Compass	Position X	Position Y	Data 1	Data 2	Data 3	CRC
16	16	16	16	16	16	16	16	16	16	8

Figure 5.12: Integrating positioning information into the information packet

The intersection management unit is also considered as a node in the network and should have the same communication structure as the one used by the platoon vehicles. The main data that is sent can be summarized as:

- Information about the traffic light, such as the actual phase, the time to next red phase, and the time to next green phase.
- Information about other vehicles (position and speed) and the potential of collisions with them.
- The suggested platoon scenarios, such as changing the platoon mode, which will be explained later, and changing the velocity for the platoon leader.

5.3.3 Intersection Controlling

Using the positioning system, described in Section 5.2, each vehicle in the platoon gets its position via a receiver mounted on top of it. We assume that the position and the size of the intersection are already known by the approaching vehicle. Thus, each vehicle can estimate the duration of stay within the intersection area by assessing the entering and leaving time.

In order to coordinate the movement of the platoon in the intersection we propose two variants depending on who is making the decision:

- By the intersection management unit:

  In this method the intersection management unit collects the platoon information and the information from the other approaching vehicles. Since it has the information of the road sensors and the traffic light (In case of a signalized intersection), the best behavior of the platoon can be chosen and send as an instruction to the platoon to change its operation mode.

- By the platoon leader:

  The platoon leader gets information from the intersection management unit and sometimes from the other approaching vehicles. Based on this information it chooses the best platoon operation.

In the following the possible platoon operations at the intersection are described.

Platoon Operations

On the highways the platoon usually has one operation mode. The vehicles follow the leader longitudinally while maintaining a safe and constant distance. When facing an intersection, the platoon must be ready to operate in other modes:

1. PASS: This mode is similar to the ordinary mode. Here all vehicles go through the intersection following the leader without changing any control parameters.

2. STOP: The platoon with all its vehicles should stop before entering the intersection area.

3. SEPARATE: Some vehicles, including the leader, pass through the intersection at wait after crossing it, whereas some platoon vehicles should stop before entering the intersection area in order to not risk a collision or cross the intersection through the red light. The first vehicle of the ones left behind takes the role of the leader and forms a second platoon.

4. JOIN: This case follows the split mode and means that the second platoon should try to join the original platoon again. When the distance between both platoons is sufficient to form one platoon again, the platoon goes back to the usual highway mode.

As mentioned in Section 3.3 the platoon leader in the platform is manually controlled (either with a joystick or through signals from the laptop monitor), while all other vehicles in the platform are following the leader and trying to maintain a constant distance to the front vehicle. In the intersection area it is necessary to adjust the behavior of the platoon to deal with the new circumstances. Therefore, in addition to the platoon operations defined above, the main software of the vehicles has been extended to determine different modes of each vehicle. The individual states of the platoon vehicles determine the total behavior of the platoon. The vehicles get their mode changes via WiFi either from the platoon leader or from the intersection management unit. Each truck in the platoon can have one the following modes:

1. Platooning: All platoon vehicles usually have this mode by default. The followers use the control algorithm and follow the leader, which is driven with a predefined speed, maintaining a safe distance to the vehicle ahead. If the platoon is in its default Pass mode, all its vehicles have the Platooning mode. This status is also set automatically after a successful Join platoon operation.

2. Split: A vehicle could have this mode after a change in a platoon operation to Separate. A vehicle with this mode is then the first vehicle left behind the intersection and will be the leader of split trucks. If a vehicle is in this mode, it should stop and wait until its mode changes.

3. Merge: By changing the platoon operation into JOIN, the leader of the split trucks with the Separate mode will change its status into this Merge status. This vehicle should then travel with a constant predefined speed. This movement is performed until the distance to the front vehicle corresponds to a threshold value. If this is the case, the status of the platoon operation should be automatically reset to Pass.

In order to the achieve the changes in the platoon operations two commands can be defined:

1. ForceSpeed(v): In this operation the vehicle moves forward with the given speed v. The vehicle does not apply to the control rules of the platoon to maintain a constant speed and the values retrieved by the controller are then ignored. If the distance in front of the vehicle becomes smaller than a critical value (15 cm), the vehicle performs an instant break in order to avoid a collision.

2. GoTo(x,v): The vehicle starts moving with the given velocity v untill the position x is reached and then stays there. This function applies the ForceSpeed(v) untill a short distance before the given position x and then a brake is performed.

Each vehicle in the platoon has two platooning tables plTableInt and plTable. plTableInt is an array of variables in which in this array the ID of the vehicles and their position in the initial mode of the platoon are included. The actual arrangement is included in plTable. In the ordinary platoon mode the table plTable is equal to the initial state plTableInt. Such platooning tables could be as follows [4 1 2 3 0 0 0 0]. It means that the current platoon consists of four trucks. The truck with ID 4 is the leader and the trucks with the IDs 1, 2 and 3 are the first, second and the third follower, respectively. A table of [1 2 3 0 0 0 0] means that the platoon consists of three trucks only, a leader with the ID 1 and other followers with the IDs 2 and 3. When changing to the platoon operation Separate, a change of the plTable follows so that in this table just the vehicles after the split are included.

Figure 5.13 shows the process of the vehicle and how it has been extended to manage the split and merge of the platoon.

Example

As an example for the platoon operations, the following initial table is given: plTableInt=[4 1 2 3 0 0 0 0]. If the vehicle with ID 2 must be split and a split command is initiated changing the platoon mode into the Separate mode. The vehicles with IDs 1 and 4 still keep the truck Platooning table plTable= [4 1 2 3 0 0 0 0]. Trucks with IDs 2 and 3, however, have the new Platooning table plTable= [2 3 0 0 0 0 0 0]. After the split, trucks 4, 1 and 3 have the status of platooning and only truck 2 has the status of split. So truck 2 is the leader of the left behind trucks. In the Join mode, truck 2 will have the Merge mode and drives with constant speed until the first platoon is approached, then the truck status changes into the Platooning mode. Thus the Platooning table of trucks 2 and 3 is set to the initial table plTableInt again, which means that all the trucks will again have the Platooning table [4 1 2 3 0 0 0 0] again.

5.4 Unsignalized Intersection

In this section, the process of a platoon of trucks approaching an unsignalized intersection is investigated. Some methods used to detect the collision and how to maneuver to avoid that are introduced. In the case of dangerous situations where a collision could happened in the platoon, the platoon must act to eliminate this risk. The intersection consists of two straight roads intersecting and forming a right angle, where each road is treated as a single lane. The hardware implementation of the collision detection and avoidance systems were done with the help of the bachelor work of Ali Tarzan [Tar12]. The platoon is moving toward the intersection on one road and another vehicle V_s moves through the intersection while following the other road. In a Cartesian coordinate system with two dimensions the platoon is driving along the x-axis, so that V_s is driving along the y-axis. In these test cases we consider only the longitudinal dynamics of the vehicle, so all vehicles only go straight. Figure 5.14 shows this scenario.

5.4.1 Test Cases for an Unsignalized Intersection

We consider a scenario where a truck platoon P with n vehicles approaches the intersection and a vehicle from the side V_s does not reduce its speed when approaching the intersection.

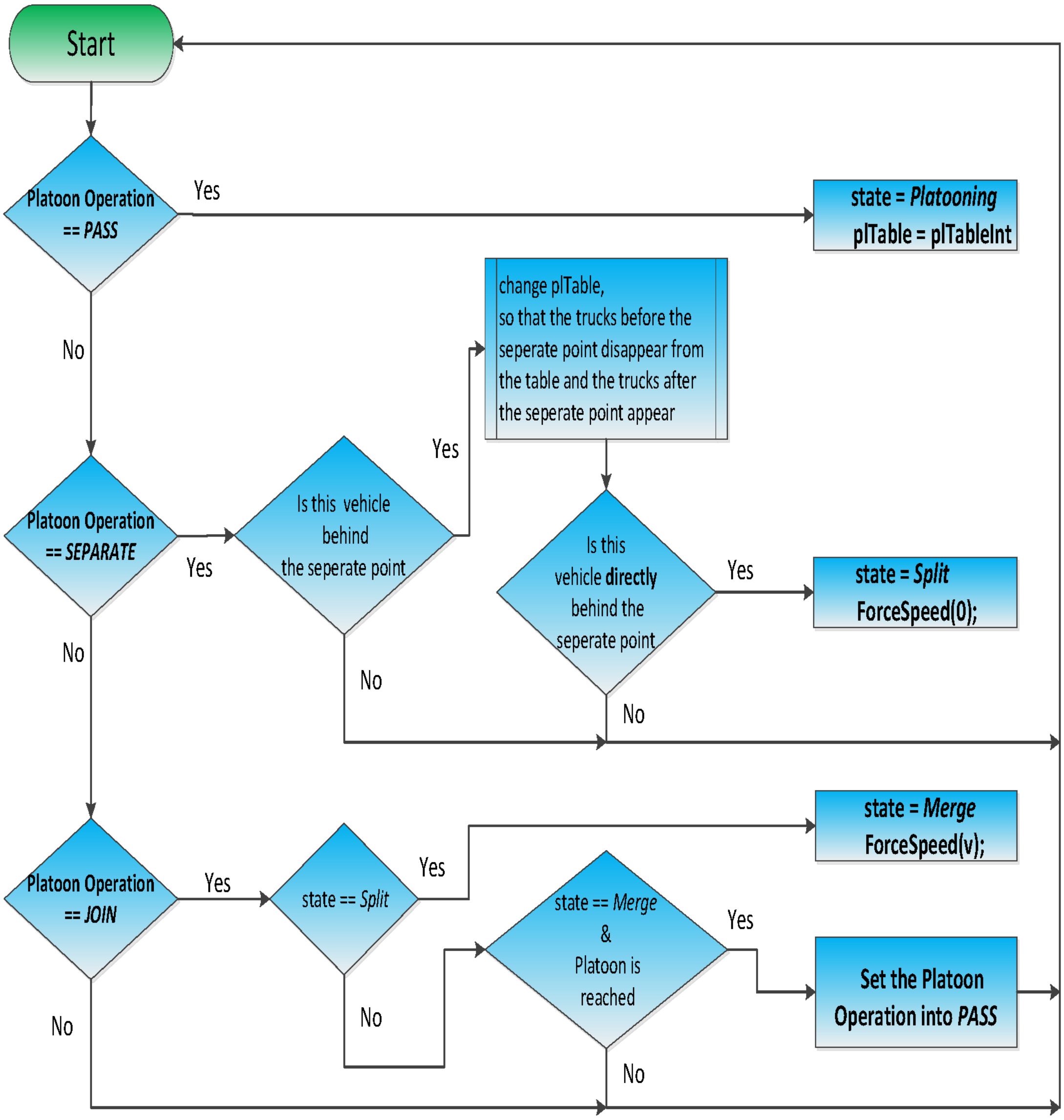


Figure 5.13: Description of Separate/Join operations

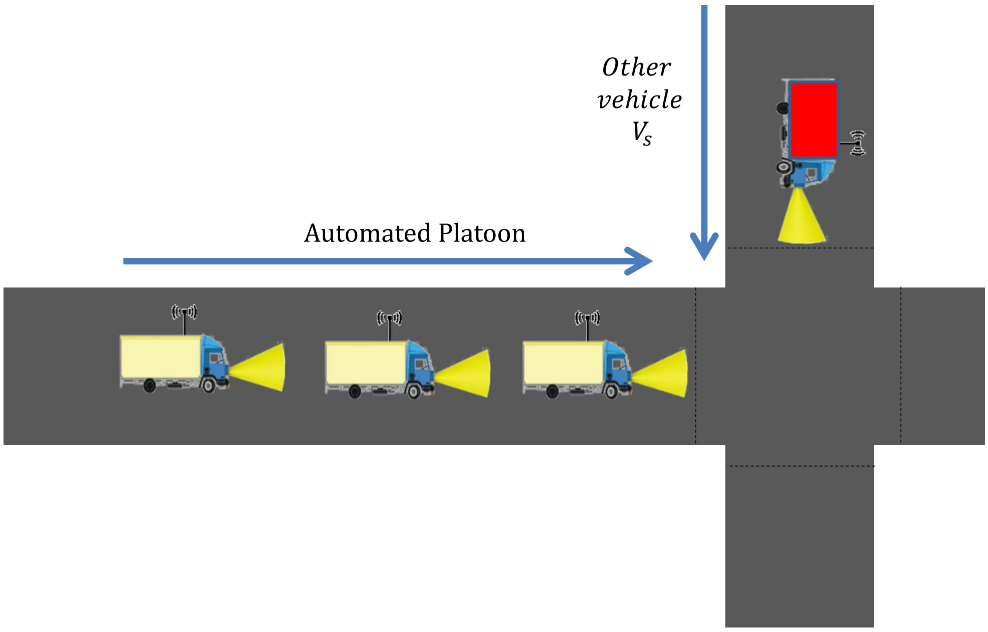


Figure 5.14: Intersection scenario

This scenario could lead to a collision with the platoon vehicles. In this work we assume that we have no influence on vehicle V_s. Hence the platoon has to act and avoid the collision. The platoon gets information from the intersection management unit about other vehicles approaching the intersection (in our scenario just information about V_s).

Depending on the characteristics of V_s (a communication system and/or positioning system are available or not), the intersection unit gets information about V_s in different ways. These possibilities are summarized in Table 5.2.

In absence of the intersection management unit due to some kind of errors, the decentralized control is used. The vehicle V_s and the platoon use the V2V communication to exchange information and based on this information the platoon leader decides about the platoon operation. This kind of control is effective only if only vehicle V_s is provided with both a communication system and a positioning system.

5.4.2 Collision Detection

The platoon uses the information (speed and position of V_s) sent either from the infrastructure management unit or from V_s or from both in order to estimate the position of V_s while crossing the intersection. The critical area is the middle of the intersection. In this area vehicles entering from different directions should be avoided. If we take the Figure 5.4 as an example for the movement of the platoon and vehicle V_s, the intersection area is therefore defined in this coordinate system, as the area $x < -ISL \wedge x < ISL$ and $y < -ISL \wedge y < ISL$, where ISL is the infrastructure stop line which can be varied to test different intersection areas.

Table 5.2: Measuring the characteristics of an approaching vehicle

Characteristics of Vs	Calculation of speed of Vs	Calculation of Position of vehicles Vs
Vs is equipped with both, communication system and positioning system	Vs internal sensors	indoor positioning system
Vs is only provided with a communication system	Vs internal sensors	infrastructure road sensors
Vs has neither communication nor positioning systems	infrastructure road sensors	infrastructure road sensors

At any time the following state should be avoided:

$$\forall i \in [1, n] : [(P_{x_i} > -ISL) \wedge (P_{x_i} < ISL)] \wedge [(V_{S_y} < ISL) \wedge (V_{S_y} > -ISL)],$$

where P_{y_i} is the position of the platoon vehicles and V_{S_x} is the position of V_S.

The above mentioned condition does not take the size of each vehicle into consideration. Therefore it is necessary to extend this condition. First, the predicted entry time and the leaving time for each vehicle into the intersection area is calculated. Knowing these times for each vehicles, the collision can occur only if two vehicles are in the intersection area at the same time.

$$(A.enterTime < B.leaveTime) \wedge (B.enterTime < A.leaveTime),$$

where A and B, respectively, denote the vehicles. This condition is tested in pairs for all vehicles with different directions. Vehicles in the same direction are not checked for collisions, since they are allowed to use the same exclusive area.

Applying this condition to our specific platoon scenario, the new condition is:

$$\forall i \in [1, n] : (P_i.leaveTime < V_S.enterTime) \wedge (V_S.enterTime < P_i.leaveTime)$$

5.4.3 Collision Avoidance

If a collision was detected, a change in the planned trajectory for one or more of the road users needs to be considered. In this work only the case of changing the platoon's behavior is considered, since we have no influence on the other vehicles. Therefore the platoon reacts and changes its mode.

It should be noted that the vehicle may not manage to stop before entering the intersection area even if there is a collision potential. This can happen if the vehicle cannot reduce its speed in order to stop based on the deceleration values that the vehicle can achieve. Using the current speed and the actual distance to the intersection, the estimated maximum stopping distance is compared to the distance to the intersection. If it is bigger, a stop before the intersection is not considered for this vehicle.

The estimation of the braking distance was initially performed experimentally based on different speed values. We then obtained a function which gives the maximum braking distance as a result for the current speed. For scaled models it was noticed that is comparison to real cars there were very short braking distances, even for high speeds. Therefore, the calculation of the braking distance was modified in order to take the values from the real vehicles into account. So, for each vehicle there exists information on whether the vehicle could cause a collision or stop before entering the intersection area and therefore avoid the collision.

When implementing the algorithm, we have taken into account the following factors: It is sometimes advantageous and even necessary for the platoon members to not separate from each other. This option was thus implemented. In addition, if an acceleration of the platoon through the change of the leader's speed is necessary to prevent a collision of some vehicles, this is done. However, it is important that the vehicles of the platoon should not be forced to accelerate suddenly or sharply, because this can lead to a collision inside the platoon itself.

Thus, the platoon can have the following behaviors related to collision management:

- If no collisions are detected:

  Maintain the speed of all vehicles and drive in PASS mode.

- At least one collision is detected and all vehicles can come to a stop before the intersection:

  Let all vehicles at the crossing, STOP mode until the other vehicle passes.

- At least one collision is detected and not all vehicles can come to a stop before the intersection:

  SEPARATE the platoon if possible by stopping the back part of the platoon and, if needed, accelerate the forward separated part of the platoon.

In the case of the third option it is still necessary to determine at which point the platoon should be separated. The search for a vehicle which has collision potential starts with the leader and proceeds up to the last vehicle in the platoon. The leader of the platoon should also be able to accelerate if needed. The speed which the leader should achieve depends on the time t that the other vehicle, with which the platoon could collide, needs to reach the defined intersection area. If the distance of the last vehicle in the in-front split platoon in order to leave the intersection area entirely is denoted with s then the necessary velocity is calculated by $\frac{s}{t}$.

Since it is important in some critical situations that the target speed is actually reached rapidly and stably, in some test scenarios we considered that all vehicles have to be accelerated directly to the target velocity, rather than determining the driving behavior of the vehicles in the platoon by following their control rules (which can reduce a slight delay). The calculation of the target speed, which is calculated by the platoon controller, is then ignored for a short time and the target speed determined by the intersection management is directly used as the input to the speed controller for each vehicle.

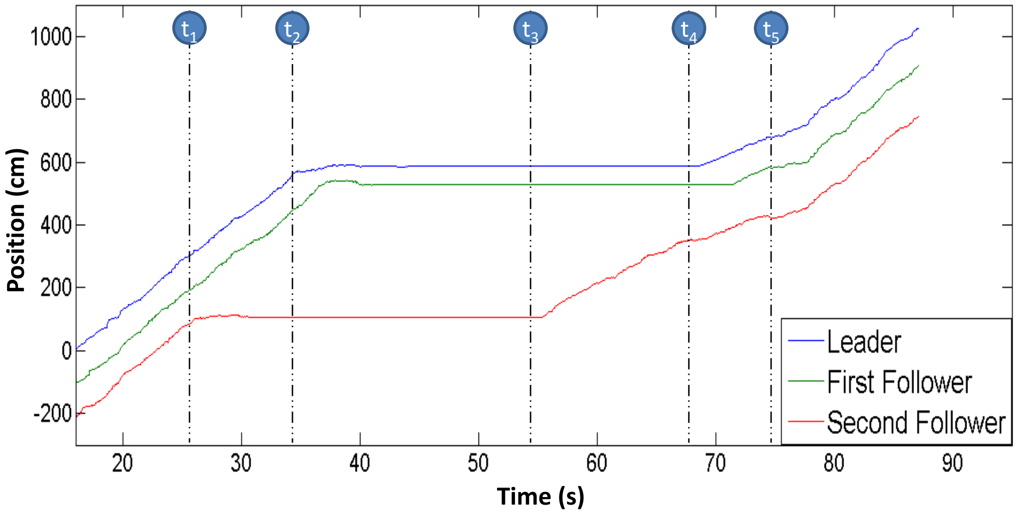
5.4.4 Experiments and Results

In this experiment, the platoon consists of three trucks: a leader and two followers. The platoon drives in the ordinary mode with 30 cm/s until it approaches an intersection. The distance between the platoon vehicles should be maintained at 40 cm. The length of the trucks l is 60 cm. The process of this experiment is depicted in Figure 5.15.

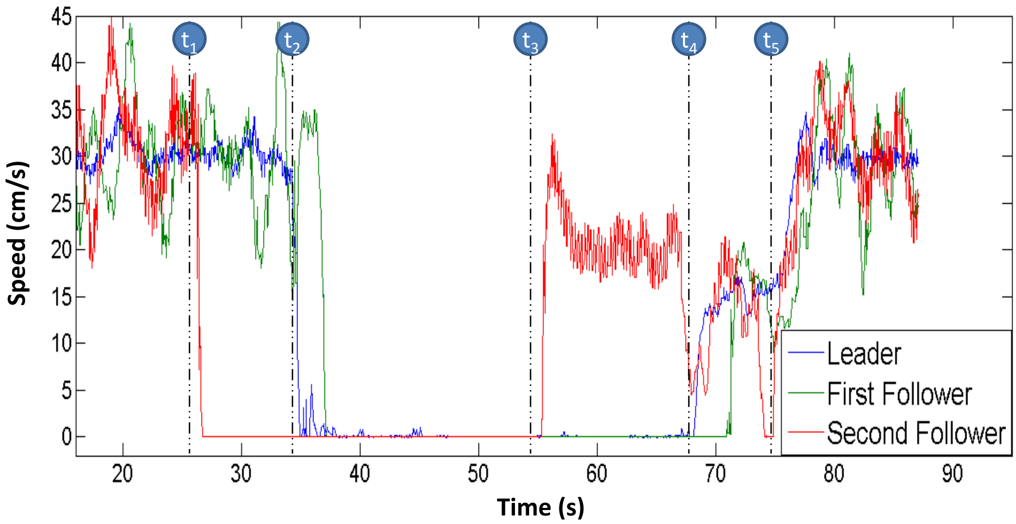
The platoon cannot cross the intersection with all its vehicles since another vehicle crosses the intersection at the same time. In this experiment the information of the existence of an another vehicle was provided manually and not through the intersection unit sensors. This procedure was necessary for this particular experiment since the intersection unit sensors have not always delivered a reproducible scenario.

With the algorithm explained before, the platoon calculates which vehicle should be stopped before the stop line. In this scenario the vehicles starting from the second follower should stop. Therefore, a SEPARATE command is initiated at t_1. This command is addressed to the second follower who changes his status to SPLIT, which causes cause him to stop, whereas the leader and the first follower pass the intersection untill t_2, at which they stop and wait.

The second follower is ready to pass the intersection at t_3 and starts the JOIN mode by moving with a predefined speed of 20 cm/s to join the platoon, which for its part starts to move at t_4. The vehicles are united again in one platoon at t_5 and drive forward with the previous speed 30 cm/s.



(a) The position



(b) The speed

Figure 5.15: Separate and join operations for a platoon

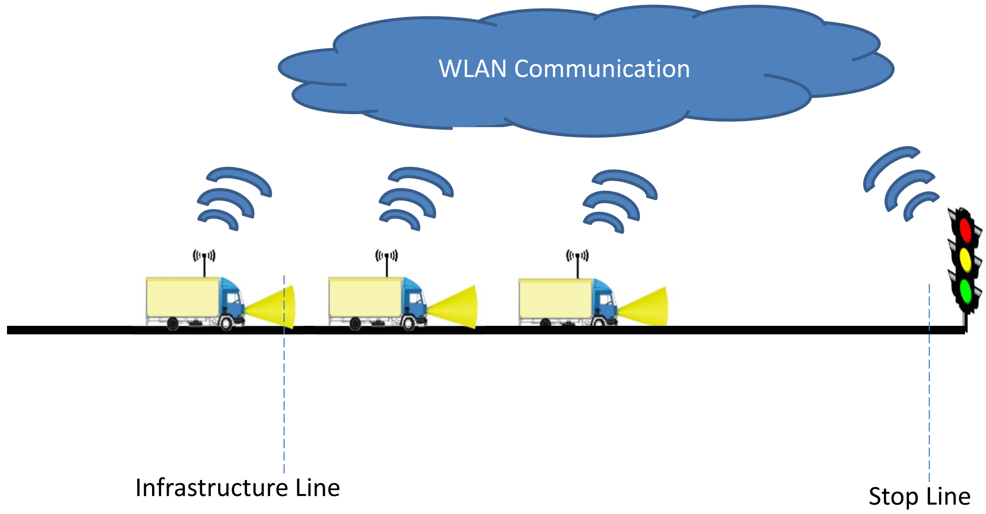


Figure 5.16: Communication with a traffic light

5.5 Intersection with Traffic Lights

Red light running violations are one of the main causes for severe injuries at intersections. In this section the behavior of a platoon approaching an intersection with a traffic light is investigated. However, in this work the safety at these kinds of intersections is considered only from the platoon side and not from other participants at the intersection.

We consider the safety factors in two ways: The platoon members should not collide with each other when crossing the intersection and no automated vehicle in the platoon should enter the intersection while the traffic light status is red. In addition to safety, we concentrate on the efficiency of driving the platoon through the intersection. This includes avoiding sharp changes in the acceleration of the platoon's vehicles.

5.5.1 Test Cases for a Signalized Intersection

The intersection management unit, which is equipped with a traffic light, provides information about the traffic light, such as its current status, time to the next signal phase, and the time cycle of the signals. Based on this information the platoon behavior will be optimized in order to cross the intersection efficiently and safely. For this system the assumption was made that the leader is also automated. This means that in order to make the platoon cross the intersection efficiently, a suggested velocity for the leader vehicle of the platoon should be sought.

In the following, some of the definitions, strategies and methods of choosing the platoon modes, considering the test cases for a signalized intersection, will be presented. These have partly been introduced in the bachelor thesis of Lars Tholen [Tho12].

Definitions

In the following some necessary notations which will be used in this section are introduced:

- Infrastructure line: This line simulates the communication radius. This means that only behind this line is the platoon able to initiate communication with the intersection. If the communication is enabled, the search for the best behavior for the platoon can be started.

- Norm velocity V_{norm}: This velocity is defined as the velocity at which the platoon reached and passed the infrastructure line.

- Range of velocity changes: If the system advises a new velocity, this should be in a certain range around the actual velocity to avoid sharp changes. This means that the new advisory velocity V_{new} should apply: $V_{new} \in [V_{old} - V_{old} * V_{Ratio}, V_{old} + V_{old} * V_{Ratio}]$.

- Minimum velocity V_{min}: This velocity is defined as the one that the vehicle should have at least in order to cross the intersection. This minimum speed can be ignored only when the vehicles in the platoon should cross the intersection after waiting at the stop line of the traffic light. The main reason to define this value is because the traffic flow would suffer greatly if the platoon is driven with a very low speed. In addition, we define this value since the drive of the platoon model is not very reliable at low speeds. Therefore the V_{min} is set to $10cm/s$ in most of our experiments.

- Maximum velocity V_{max}: This velocity should not be exceeded in any situation.

- Interval for a valid velocity: This is defined as the interval between the minimum and maximum limit for the velocity $[V_{min}, V_{max}]$. Therefore the velocity of a vehicle in the platoonV_i should apply: $V_i \in [V_{min}, V_{max}]$.

- Ambulance mode: This mode is considered to increase the safety at intersections. When an ambulance is moving towards an intersection it can communicate with the traffic light in order to manipulate its phases so that it can reach the intersection with a green phase, forcing other traffic lights in the intersection to go red to prevent any other vehicles from entering the intersection. This can be seen as an abrupt change of the traffic light phase, giving other autonomous vehicles not so much time to react. This behavior was taken into account while designing our system.

Strategies

A strategy is a maneuver that specifies a particular driving behavior of the platoon. These strategies are an extension of the platoon operations introduced in Subsection 5.3.3, so every strategy can be seen as a combination of conditions and appropriate platoon operations. Thus, the desired behavior of the platoon approaching a signalized intersection can be determined by choosing the valid strategy. All of the necessary strategies can be specified in a strategy profile. Each strategy in the strategies profile has a unique priority.

To keep this simple, we arranged the strategies in the strategies profile as follows: On the top of this profile the strategy with the highest priority is placed. The strategies with lower priority are then placed one after the other. Therefore, starting from the top of the strategy profile and going down sequentially, it must be checked if a valid strategy exists. Once a valid strategy is found, the search is ended and this strategy determines the behavior for the leader of the platoon before the entering the intersection.

Since our system should also be able to respond immediately to any change in the parameters of the traffic light (such as in the ambulance mode, which was already explained before) and the uncertainty in mechanical inertia of the scaled trucks (which can affect the calculation of the right time to reach the intersection), it should not be insisted on the initial strategy, calculated when the platoon first reached the infrastructure line, but to update strategy based on the new information periodically. Therefore every 100 *ms* it must be checked if the chosen strategy applied and is still valid, otherwise a search for new one should be conducted.

In the following, the different possible strategies are listed and explained shortly:

- Drive with the V_{norm}:

  This strategy checks if the platoon could pass the intersection during the green phase of the traffic light at a normal speed. If this applies, no change of the platoon behavior should be considered.

- Acceleration check:

  It is tested if with a gradual acceleration, starting with an acceleration of zero, the entire platoon could go through the next green phase of the traffic light. If a valid acceleration was found the search is stopped. However, if the resulting velocity due to this acceleration exceeds the allowed range of velocity changes, this strategy is considered as not valid.

  In this strategy we implemented two different scenarios, smooth acceleration and strong acceleration, which differ in their various limitations in terms of acceleration and velocity. The purpose of this differentiation is to avoid the extreme behavior. In smooth acceleration the maximum allowed acceleration is small and depends on the initial velocity of the platoon, while in strong acceleration these limits are independent of the original platoon speed. This would then be possible if the platoon is far away from the intersection area.

- Slowing down:

  If the platoon is not able to pass the intersection with its current speed, the platoon should slow down in order to reach the next green phase without stopping. This strategy is almost the same as the acceleration, but here the acceleration is negative, leading to vehicles slowing down their movement. Another difference is that by choosing the appropriate acceleration, no graduate change in acceleration is needed, just a search for a value in the acceleration interval is still valid.

- Split:

  This strategy divides the platoon if not all platoon members can pass the intersection due to the green phase of the traffic light. If the platoon can pass the intersection, it is not usual to split the platoon and lose the advantages of the effect of less air resistance. But if the whole platoon should wait at the traffic light, it may be useful to try to get as many of the vehicles through the intersection, while the other vehicles wait at the intersection.

  If this strategy is performed, a SEPARATE operation as explained in Subsection 5.3.3 should be subsequently commanded. However, we extended this operation to enable different capabilities of the first part of separated vehicles. Before entering the intersection, the first vehicle can either drive with a constant speed, or it can accelerate to go through the intersection as explained in 'Acceleration check' strategy.

- Stop at the traffic light:

  The platoon slows down so that it comes to a standstill at the traffic light. Then the strategy is set to wait at traffic lights.

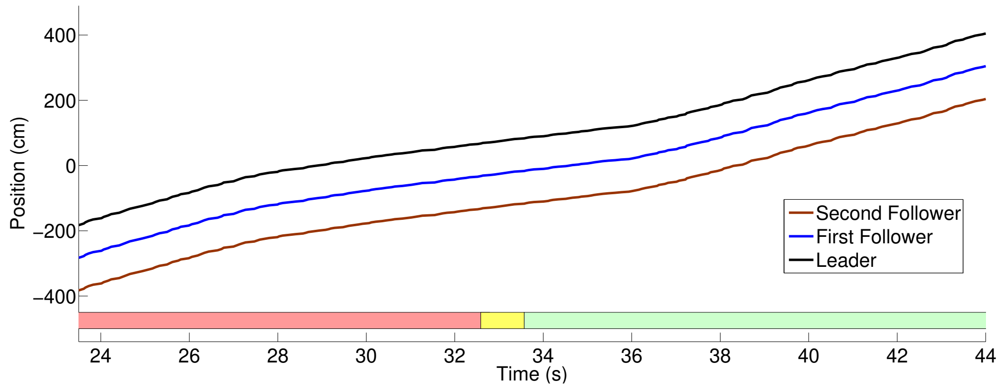
- Waiting at the traffic light:

  This strategy serves only after a complete stop before the traffic light. When the traffic light displays green again the platoon should drive immediately so there is enough time to pass the entire platoon through the intersection. The strategy 'Acceleration check' is then applied again.

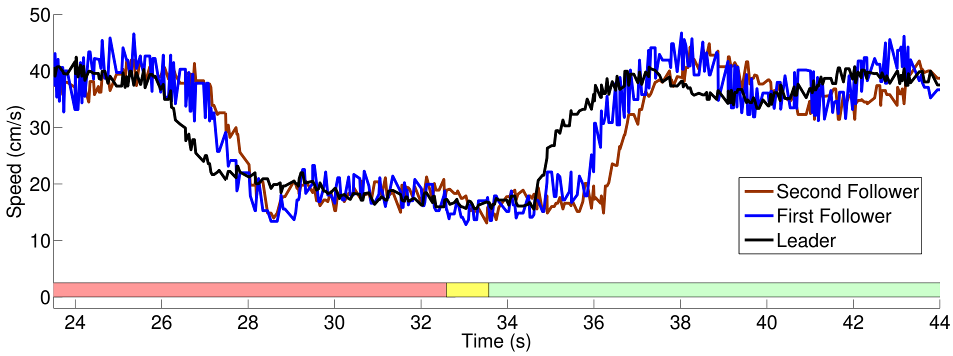
Choosing The Platoon Strategy

As explained in Subsection 5.3.3 the decision can be made from either the intersection management unit or the platoon leader. For choosing the best platoon behavior it makes sense to check first if the platoon could pass the intersection with a constant speed without changing its velocity. This means that the V_{norm} is maintained and not changed. This is achieved by putting the strategy 'Constant speed' on top of the strategies profile. To test the implemented system, it was chosen for many experiments to put the 'Stop at the traffic light' strategy at the bottom of the strategy list, although it is considered to be the safest strategy in the strategy profile.

As explained in Chapter 3 the implemented controller regulates the speed of the vehicle by giving it the desired speed values, not the acceleration ones. Observations have shown that the inertia of the trucks is relatively small, so they can accelerate very quickly. For this reason, the desired acceleration/deceleration was achieved by changing the desired vehicle velocity to a higher/lower value. The truck then tries to adjust its speed quickly to that specified value. Since the strategies are tested every 100 ms, the check if the platoon should go back to the V_{norm} is done permanently.



(a) The position



(b) The speed

Figure 5.17: The platoon approaches a signalized intersection: Slow down

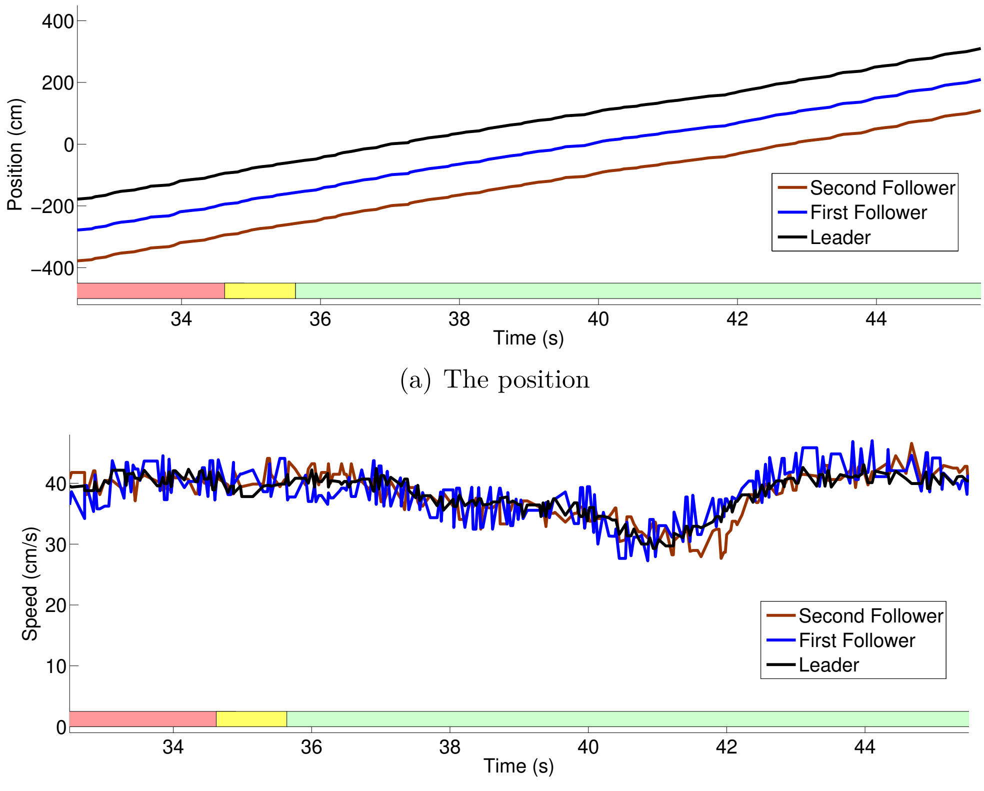
5.5.2 Experiments and Results

In these experiments, the platoon consists of three trucks: a leader and two followers. The distance between the platoon vehicles should be maintained at 40 cm. The length of the trucks l is 60 cm. The infrastructure line is at position Y_{IL}= -200 cm, and the stop line is at position Y_{SL}=15 cm. Depending on the traffic light phase and the time at which the platoon reaches the intersection, different scenarios were tested. Three of them will be explained below.

1. Slow down:

   Figure 5.17 shows the results of an experiment where the platoon is driving with a speed of 40 cm/s and reaches the intersection at a red phase. In order to avoid stopping and to cross the intersection efficiently, it reduces its speed and passes with all its members in the following green phase with a constant velocity of 20 cm/s. After passing the intersection the platoon velocity changes in order to reach its former speed of 40 cm/s.

2. Drive in PASS mode:



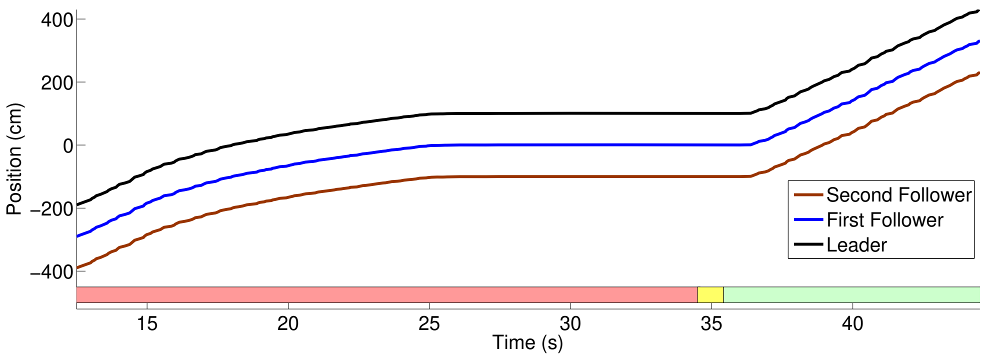
(a) The position

(b) The speed

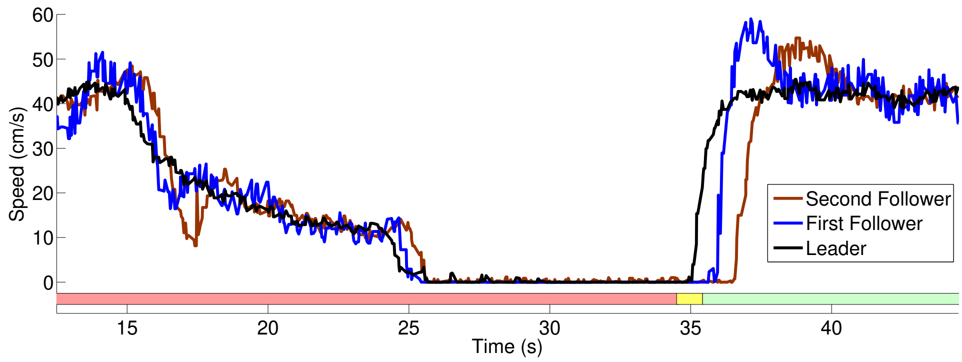
Figure 5.18: The platoon approaches a signalized intersection: Drive in PASS mode

In this experiment the platoon reaches the intersection while the traffic light has a green phase. Therefore, the platoon will not change its speed and continues driving with a speed of 40 cm/s.

3. Stop at the intersection: Since the platoon with all its members cannot cross the intersection in the green phase, the platoon should stop and wait at the intersection. Figure 5.19 shows the results of this experiment where a platoon reaches the intersection with a speed of 40 cm/s at a red phase. The change of speed is done gradually, firstly to 20 cm/s and then to 0 cm/s. After waiting for the phase of the traffic light to change to green, the platoon starts to move again with its previous speed of 40 cm/s. In this experiment it was not possible for the platoon to just slow down to avoid a stop maneuver, like in the first experiment. This is because the speed of the vehicle would then have gone under the defined permitted vehicle speed of 5 cm/s .



(a) The position



(b) The speed

Figure 5.19: The platoon approaches a signalized intersection: Stop and then drive

6 A Platoon Crossing an Intersection: Simulation and Verification

As explained before in this thesis, the intersection is a vital part of the road. The behavior of the vehicles in the intersection area should be reliable, otherwise a collision could take place. For autonomous vehicles, the behavior can be checked by verifying the control strategies that assure collision-free performance for these vehicles. Simulation and verification are important methods for testing different processes and the effectiveness of the developed systems.

In this chapter the simulation environment, which was developed in order to test the platoon behavior of the platform at different types of intersections, will be introduced. This simulation environment allowed to perform two kinds of experiments. On one side the testing of collision avoidance algorithms is possible, and the testing of the efficiency of this crossing scenario by observing the energy consumption and the reduction of CO_2 could be done on the other side. In addition, this chapter will discuss the formal verification of the platoon behavior in cases of potential collisions, where a split of the platoon has to be induced, when a the platoon crosses an intersection while other traffic members cross the intersection at the same time.

External Contributions

In Section 6.2 some simulations were done in oder to investigate the effect of platoon driving through a signalized intersection on fuel consumption and the CO_2 emissions. These simulations are part of the bachelor thesis of Lars Tholen [Tho12]. The formal verification of a platoon crossing an unsignalized intersection, presented in Subsection 6.3, was developed in cooperation with Ibtissem Ben Makhlouf.

6.1 Simulation Environment

Simulation has a number of significant advantages, since with relatively little effort different scenarios can be evaluated. The simulation system made use of a formal programming approach of Object-Oriented Programming (OOP) in MATLAB. The reason for choosing OOP was that this method can manage software complexity efficiently. Since in our platform MATLAB, including a number of its products, was used in a large part in our software development, we continue to code and run it in this simulation environment.

To make the simulation more illustrative we included it into the control and monitoring GUI, which is already explained in Subsection 3.2.2. As can be seen in Figure 6.1 we have extended the GUI so that it can simulate the movement of the platoon of vehicles through the intersection. This environment enables testing signalized as well as unsignalized intersections.

The main modules in the simulation system are: control of the movement of the vehicles and the intersection management.

6.1.1 Vehicles Dynamic

Two mathematical models of vehicles crossing the intersection have been defined:

- Platoon model:

  For controlling the movement of the platoon in longitudinal dynamics we modified the controller presented in [MDAK11] in order to include the position as well. The dynamics of the whole platoon can thus be summarized as a state space form with the state vector $x_p = [\cdots p_i, \dot{p}_i, \ddot{p}_i \cdots]^T$, where p_i represents the position, $\dot{p}_i$ the velocity and $\ddot{p}_i$ the acceleration, and $i \in [1, n]$ where n is the number of the vehicles in the platoon. The leading vehicle's acceleration $a_L = \ddot{p}_1$ enters the dynamics as a disturbance to the system and thus leads to the following linear state-space description

$$\dot{x_p} = Ax_p + Ba_L. \tag{6.1}$$

- Vehicle model:

  Other vehicles that are not part of the automated platoon are also modeled as a state-space model as follows:

$$\dot{P}_{vehicle} = V_{vehicle}, \tag{6.2}$$

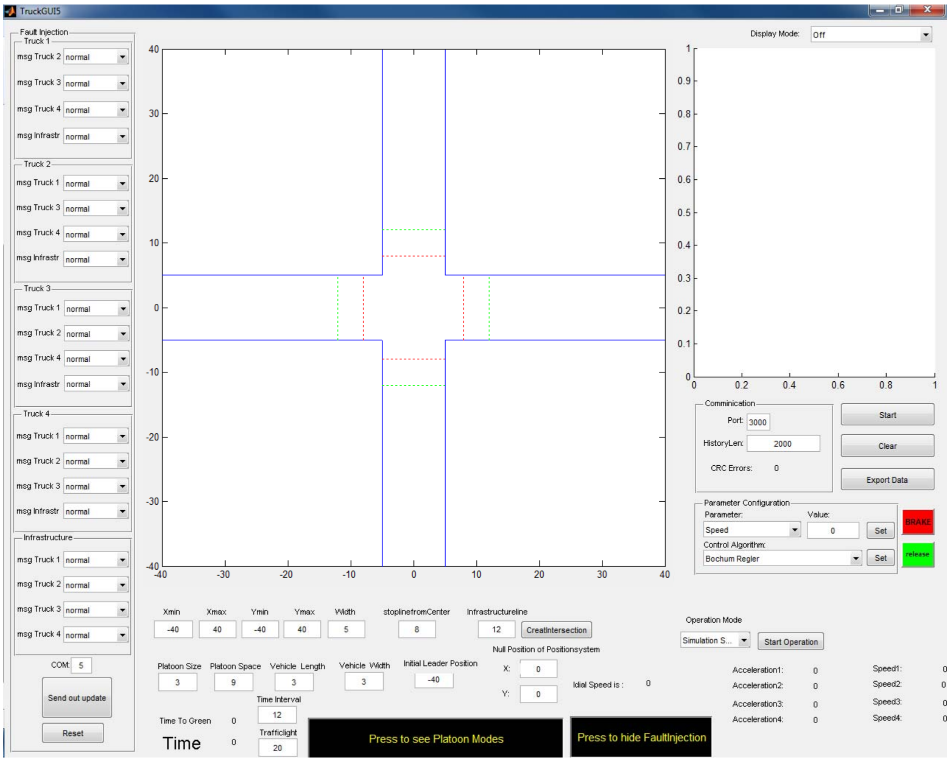
$$\dot{V}_{vehicle} = a_{vehicle}. \tag{6.3}$$

  where $P_{vehicle}$, $V_{vehicle}$ and $a_{vehicle}$ are the position, velocity and acceleration of the vehicle, respectively.

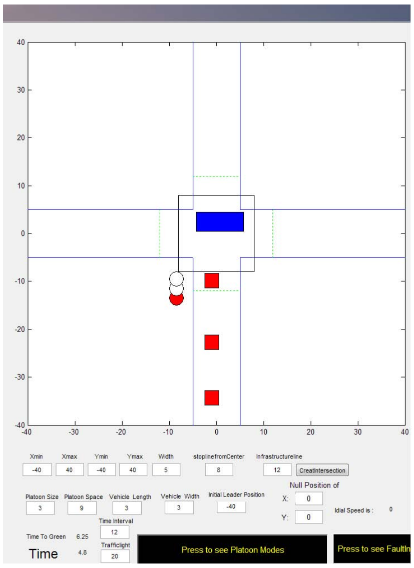
Before starting the simulation a number of parameters need to be adjusted. This includes defining the number of vehicles in the automated platoon n as well as the initial values for its vehicles and for other vehicles in the simulation. This is done by defining the initial conditions for the state-space models such the position, velocity and acceleration.

6.1.2 Intersection Management

We differentiate between two types intersections: with and without a traffic light. For intersections without traffic lights, the algorithm for collision avoidance, which is described previously in Section 5.4, was simulated. In addition, the simulation of intersections with traffic lights, as detailed in Subsection 5.5, we use this simulation environment in order to investigate the fuel consumption and the CO_2 emissions.



(a) Intersection environment



(b) Intersection with a traffic light

Figure 6.1: Intersection GUI

6.2 Simulating a Signalized Intersection

Knowing extra parameters of the driving process could be exploited to improve fuel consumption and CO_2 emissions of vehicles. These parameters could be the traffic flow status and the state of the traffic lights in the investigated area. Some methods for improving the traffic flow of the platoon, using methods of traffic light optimizations, have already been explained in Subsection 2.3.3. In this section we investigate the efficiency of a platoon of automated vehicles while crossing an intersection with a traffic light in terms of fuel consumption and the CO_2 emissions by using simulations for different driving scenarios. Large parts of this section are part of the bachelor thesis of Lars Tholen [Tho12].

6.2.1 Energy model

Since we want to evaluate the efficiency in term, of fuel consumption or the CO_2 emissions, it is necessary to have an energy model which provides reliable values for the platoon's driving situation. Although there are a number of research works which deal with achieving energy-efficient driving in automotive transport they in general use an energy model which does not take the behavior of driving in platoon formation and its effect on the energy savings into account. In addition, these energy models cannot be generalized easily.

Therefore, for the simulations a simple energy model based on theoretical physical laws is introduced. It takes into account the essential benefit factors of a platoon such as the reduced air resistance. Many formulas and driving parameters used in this energy model were used in [41].

The approach starts to determine the driving resistance which acts on the vehicle. After that, other variables such as the energy, fuel consumption and CO_2 emissions can be calculated. The driving resistance F_{drive} can be seen as the sum of four forces:

1. F_R is the friction force.
2. F_a is the acceleration resistance.
3. F_{roll} is the rolling friction of the vehicle.
4. F_{air} is the air resistance: This resistance opposes in general the actual vehicle movement. Its value depends primarily on the vehicle speed. However, in a certain formation, like a platoon, the air resistance can be reduced, as already explained in Section 2.2.2. Therefore we consider that in our energy model so that a platoon needs less fuel than if all vehicles were driving separately. Using the results of the simulations(mentioned in the work of Tsugawa and Kato [98] within the Energy ITS project) which investigate the aerodynamic drag, we modified the energy model. The value of this resistance is adjusted depending on the position of the vehicle in the platoon formation. This means that for the first and the last truck, the force is reduced to 20% of the mean value, whereas as the vehicles in the middle have a resistance reduced by 50%.

The driving resistance is changing during the movement because of accelerating and braking. To calculate the energy, CO_2 emission and fuel consumption for the whole investigated road area, the trajectory is divided into small sections s_i in meters. It can be assumed that in these sections the forces are held and do not change. Using the driving resistance F_{drive} in Newton, other important parameters for the simulations are calculated such as:

- The Energy E_v in Joule: This can be calculated by using the formula:

$$E_v = s_i \cdot F_{drive}. \tag{6.4}$$

- The Fuel consumption F_c: Since trucks usually have a diesel engine, all subsequent calculations are considered only for diesel fuel. The formula may be adopted for other types of fuel. It must only be replaced by fuel-specific values. The consumption in liters can then be calculated as:

$$F_c = 100 \cdot \frac{E_v}{H_C \cdot E_{cf}}, \tag{6.5}$$

  where H_C is the heating consumption in J/l and E_{cf} is the energy conversion efficiency in percent.

- The amount of CO_2 emission: This value is calculated by multiplying the amount of CO_2 per liter of the fuel with the F_c. For the amount of CO_2 that is produced by burning one liter of diesel fuel the value $2,62kg/l$ was considered in the following calculation [3, 6, 66, 70, 85]. At first glance, this value appears to be much too high, but knowing that CO_2 is produced only by a reaction with air, this high value becomes acceptable. This means that the following formula is used to calculate the CO_2:

$$CO_2 = 2.62 \cdot F_c. \tag{6.6}$$

6.2.2 Collected Data

The data which is used in the simulation tests is summarized in Table 6.1. In this table, the collected data are categorized depending on their characteristics. The underlined values represent the values which should be initiated before starting the simulation. The other values represent the calculated ones during the simulation. Having this information for each vehicle in the platoon, the total fuel consumption and the required energy of the platoon can be calculated by taking the sum of all the vehicles in the platoon. The total fuel consumption of the platoon is calculated per $100km$.

6.2.3 Simulation Evaluations

In this simulation environment, the platoon can change its highway mode according to the strategies which are already explained in Subsection 5.5.1. In contrast to the

Table 6.1: Table of collected data

Time (s)	Locations (m)
• Initial traffic light phase • Traffic light cycle times • Current time • Time offset • Current light phase • Time to traffic light turns red • Time to traffic light turns green	• Infra-structure line • Stop line • Vehicle start position • Vehicle current position • Vehicle traveled distance since the last step • Total vehicle traveled distance
Acceleration (m/s^2)	**Velocity (m/s)**
• Vehicle initial acceleration • Current vehicle acceleration	• Vehicle initial velocity • Current vehicle velocity
Fuel Consumption (l)and Energy (J/l) for each vehicle	**Forces (J) for each vehicle**
• Actual fuel consumption • Total fuel consumption for a section • Actual required energy • Total required energy for a section	• Driving resistance • Acceleration resistance / Grade resistance • Rolling resistance • Air resistance

implementation in the scaled vehicle, the choice of the appropriate strategy in the strategy profile is done only one time; once the platoon crosses the intersection line. The reason of not repeating the search for a new strategy is because the simulation's parameters would not be changed during the simulation, not like the real environment tests.

A number of simulation modes was chosen in order to investigate the effects of changing the different simulation's parameters. These modes are briefly explained below:

1. The traffic light phases vary gradually. This means that the length of each phase changes while the ratio of the individual phases to each other remains constant.

2. The arrival time of the platoon at the intersection varies gradually. Here, the traffic light always starts at the same phase, but the initial velocity/position of the platoon is changed. This mode observes how the platoon switches between its different strategies, which has been explained in Subsection 5.5.1.

3. The infrastructure line is gradually changed while the other parameters remain the same. This mode observes how changing the time of starting the communication between the traffic light and the platoon can effect the movement behavior. A series of tests was performed where, for example, the infrastructure line is changing in $5m$ increments from $50m$ to $250m$.

4. The infrastructure line is changed gradually while a random phase shift of the traffic light is chosen. The random phase shift of the traffic light is important in order to present a realistic scenario, so it cannot be predicted at which phase the platoon reaches the intersection.

The simulation with these different modes generates a lot of data that can be evaluated. In the following, we show some of the simulation's results.

Forces

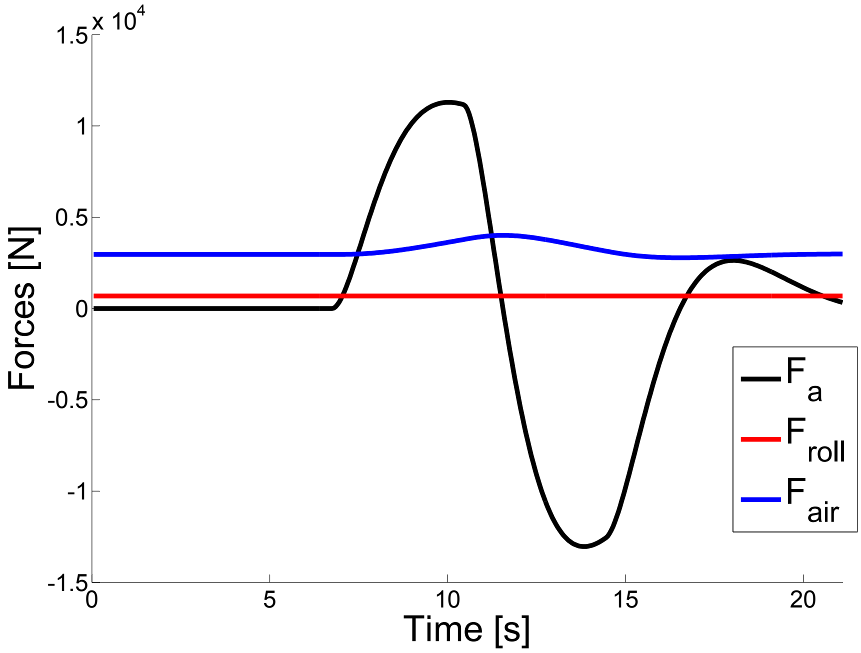
Since the forces are the reason why energy must be spent and they are therefore the cause of the fuel consumption, it is important to examine how the various forces evolve depending on the platoon behavior. If the platoon is driving with a constant speed the biggest value of forces goes to air resistance, while the acceleration resistance at a uniform velocity is zero. Figure 6.2(a) shows the values of the forces if the platoon velocity is changed from 100 km/h to 115 km/h and then decelerated back to 100 km/h. This figure shows how the forces relate to each other. During the acceleration phase the acceleration resistance is four times higher than the air resistance, although the air resistance also increases with the rise of the velocity. During a braking operation, the acceleration resistance has a negative value while the air resistance reduces.

Fuel Consumption and CO_2 Emission

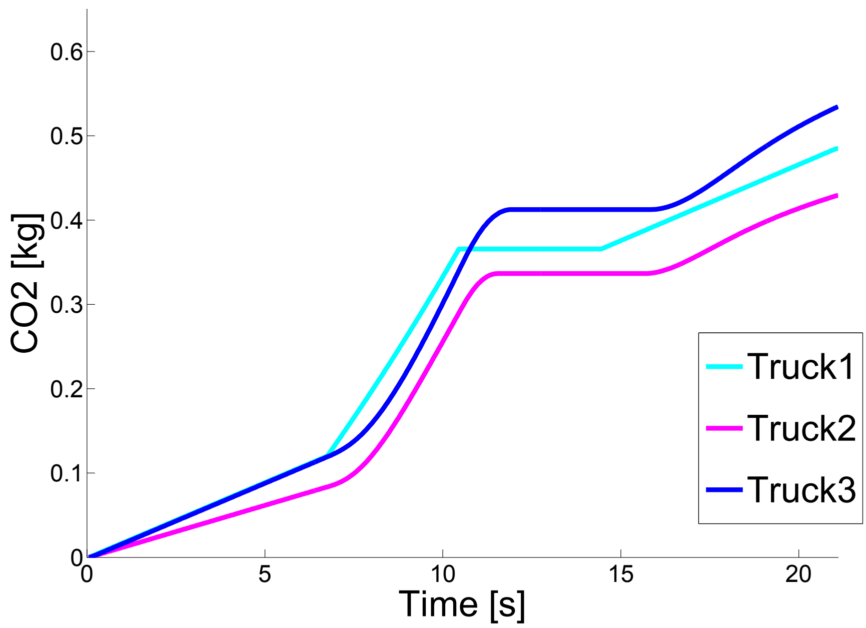
The ideal situation for the platoon is that the speed does not change when the platoon crosses an intersection with a traffic light. The value of fuel consumption in this scenario can be used as a base value to compare it with values of other platoon behaviors. In Figure 6.2(b) the fuel consumption of a platoon with three vehicles, a leader (truck 1) and two followers (truck 2, truck 3), is shown. The velocity of the platoon leader has been changed, like the experiment before, from 100 km/h to 115 km/h and then decelerated back to 100 km/h. The figure shows differences in fuel consumption, which can be explained by different air resistances. Thus, the middle truck consumes the least fuel.

Influence of the Infrastructure Line on the Fuel Consumption

The infrastructure line is the point at which the communication between the platoon and the infrastructure is initiated. In this experiment the position of this line is changed gradually. The behavior of the platoon changes in correspondence to its strategy profile. The platoon in this experiment has the following strategy profile: constant speed, slow braking, slow speed, split (constant first part), split (part acceleration), heavy braking, fast acceleration, and wait at traffic lights (priority in descending order from left to right). Figure 6.3 shows the fuel consumption with different values of the infrastructure line. The effects on the fuel consumption can be reduced to 14%. Other strategy profiles also led to a similar behavior.



(a) The forces



(b) The CO_2 emission

Figure 6.2: Forces and emission during acceleration and braking

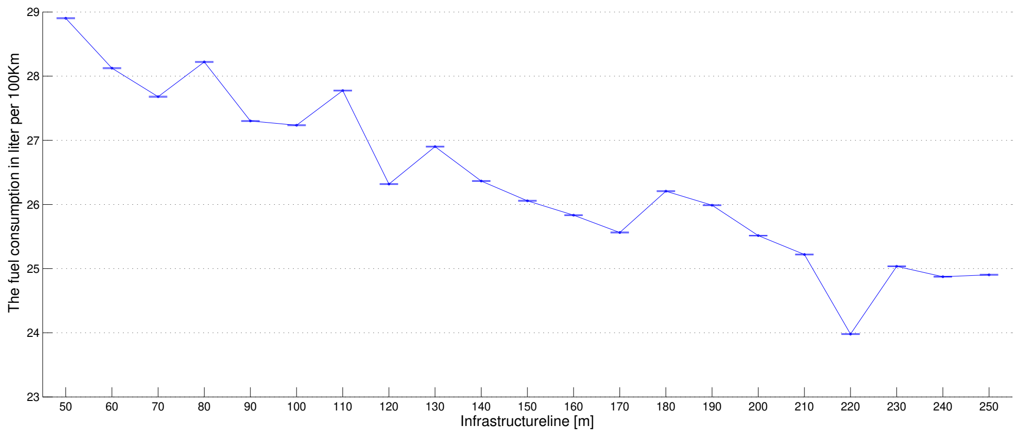


Figure 6.3: Influence of the infrastructure line on fuel consumption

6.3 Safety Verification of a Platoon Crossing an Unsignalized Intersection

When a platoon crosses an unsignalized intersection, safety is a very important factor to be investigated. The control of the behavior of the platoon should assure that no collision can occur between the platoon vehicles and other vehicles on one hand, and within the platoon vehicles on the other hand. Although the simulation environment, which was explained before, is able to simulate the collision avoidance algorithm, it provides no guarantee of the correctness of the algorithm. Therefore we are interested in using formal verification methods in order to proof the safety of the platoon while it crosses the intersection.

In Subsection 4.1.3 a number of research projects, which investigate and develop verification methods for intelligent transport systems, have been already mentioned. Since intersections are a vital part of the transportation system, there has also been some research dealing with the safety at these critical points. For example, the work of Loos and Platzer [61] shows how a system consisting of an intersection of two single lane roads and a traffic light has been modeled and verified. The authors use the tool KeYmaera in order to proof the safety requirement of the intersection control.

In Chapter 4 of this thesis, the safety verification of a cooperative platoon of vehicles driving in highway mode, was introduced. Methods to verify a longitudinal vehicle control algorithm were investigated. These verification methods were based on the reachability analysis of hybrid continuous linear systems with uncertain inputs. In this section we show how the same approach can be applied in order to investigate the decision-making process of the platoon while it crosses an intersection.

The new system of interest consists of:

- An intersection: The intersection for this study was chosen as a simple T-intersection with two lanes.

- A platoon: As explained in the simulation environment (section 6.1) the dynamics of the platoon is represented in Equation 6.1. By adding the index p (To represent the platoon) to the system matrices this equation can be rewritten as follows:

$$\dot{X}_p = A_p X_p + B_p a_L. \tag{6.7}$$

  We assume that via on-board sensors that each vehicle of the platoon can get its own information of position p_i, velocity v_i and acceleration a_i, with $i \in [1, n]$ where n is the number of the vehicles in the platoon.

- Other vehicles: For this study we consider a scenario where just one vehicle OV is driving on one lane while the platoon is on the other lane. This vehicle can also get its own information about its position P_{OV}, velocity V_{OV} and acceleration a_{OV}. The dynamics of this vehicle applies to Equations 6.2 and 6.3. The state-space model of this vehicle can thus be written as:

$$\dot{\begin{pmatrix} P_{OV} \\ V_{OV} \end{pmatrix}} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} P_{OV} \\ V_{OV} \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot a_{OV} \tag{6.8}$$

By enabling the platoon to get information on the vehicle OV, via communication or via an intersection unit, it is now possible to merge the above mentioned system elements into one model. Thus, the approach is to form a new state-space vector and a new verification condition in order to analyze the behavior of the whole system.

System Modeling

The new state-space model can be written as:

$$\underbrace{\dot{\begin{pmatrix} X_p \\ \hline P_{OV} \\ V_{OV} \end{pmatrix}}}_{X_{new}} = \underbrace{\left(\begin{array}{c|c} A_p & \begin{matrix} 00 \\ 00 \\ \vdots\vdots \end{matrix} \\ \hline \begin{matrix} 0\dots0 \\ 0\dots0 \end{matrix} & \begin{matrix} 10 \\ 00 \end{matrix} \end{array}\right)}_{A_{new}} \underbrace{\begin{pmatrix} X_p \\ \hline P_{OV} \\ V_{OV} \end{pmatrix}}_{X_{new}} + \underbrace{\begin{pmatrix} B_p \\ \hline 0 \\ 1 \end{pmatrix}}_{B_{new}} \underbrace{\begin{pmatrix} a_l \\ \hline a_{OV} \end{pmatrix}}_{a_{new}} \tag{6.9}$$

by rewriting the equation above we get:

$$\dot{X}_{new} = A_{new}\, X_{new} + B_{new}\, a_{new}. \tag{6.10}$$

Verification Condition

After formulating the system which should be verified, the next step is to define the verification condition. As previously defined in Section 5.4, the Infrastructure Stop Line (ISL) is the distance between the middle of the intersection and the point where the vehicle should stop. The goal of verification is to proof that at the same time there are no two vehicles in the intersection area. Thus, we want to proof that at any time the following statements are applied:

1. No collision between the platoon vehicles and the vehicle OV

$$\forall i \in [1, n] : \neg[(p_i \in [-ISL, ISL]) \wedge (P_{OV} \in [-ISL, ISL])]$$

2. No collision between the platoon vehicles themselves

$$\forall i \in [1, n-1] : (p_i - p_{i+1}) < SD$$

   where SD is a safe distance between the platoon vehicles

So the verification problem is now formulated as calculating the reachable sets of a linear time invariant system. This calculation and the methods used to realize it have been already introduced in Section 4.2 of this thesis.

Based on the verification the decision is made whether there is a collision or not. If a collision would occur the verification can also suggest other operations, such as stopping the platoon before the intersection line or to split the platoon at a distinct member of the platoon. In [BMDK13] the management of a platoon crossing an intersection was discussed. The collision decision was based on verification via reachability analysis of hybrid systems. Some case studies and results are pointed out.

7 Conclusion

The focus of this thesis was the development of methods for testing different scenarios for a group of vehicles moving in platoon formation. This work was motivated by the latest research work and practical issues in the automotive and robotics field on one side, and theoretical aspects on the other side.

In Chapter 1, the motivation for the work and the thesis structure was explained. Since we are interested in platoon technology as an example of an intelligent vehicle operation, Chapter 2 showed the status of research in the field of cooperative transport systems in detail. It concentrated particularly on the platooning system and explained the architecture and the advantages of such systems. The possibility of extending the potential of vehicle, by communicating with an infrastructure unit, giving the platoon the ability of crossing intersections safely and efficiently, was also introduced.

The description of the testing platform of scaled vehicles was addressed in Chapter3. Therein the system structure and the used hardware and software was explained. Communication within the platoon plays an important role in achieving the so-called platoon stability. In order to test that, the validation of two different control methods along with the achieved results were presented.

The safety verification using formal methods was explored in Chapter 4. This chapter concentrated on reachability analysis verification for a dynamic and hybrid system. This choice was made due to the fact that the movement of the platoon can be modeled as a dynamic system. To include the communication losses in the resulted model, it was also shown how the platoon can be modeled as a hybrid system. A new method to calculate the reachable states of linear systems with uncertain inputs under the assumption that the inputs are stepwise constant was also proposed. Here the original system with inputs is transformed into a system without inputs, thus the inputs need no longer be considered.

The extension of the platform, which enables the platoon to cross an intersection efficiently, was elaborated in Chapter 5. Due to the consideration of communication between the platoon and the infrastructure unit, the platoon can change its driving mode from driving on the highway to an intersection mode. It was necessary to implement an indoor position system which provides the vehicle's position to the platoon. This chapter also presented the hardware and software development of the intersection management unit and showed the necessary changes in the platoon software to perform the intersection crossing efficiently. Some test cases for a signalized/unsignalized intersection were presented. The simulation and the verification of a platoon crossing an intersection was introduced in Chapter 6.

Bibliography

[1] Cooperative Intersection Collision Avoidance Systems (CICAS) , www.its.dot.gov/cicas/.

[2] Safe Road Trains for the Environment(SARTRE), www.sartre-project.eu.

[3] *Policy Instruments for Achieving Environmentally Sustainable Transport.* OECD Publishing, 1998.

[4] World Report on Road Traffic Injury Prevention: Summary. Technical report, World Health Organization, Geneva, 2004.

[5] Motor vehicle traffic crash fatality counts and estimates of people injured for 2005. Technical report, National Highway Traffic Safety Administration, 2006.

[6] Sustainability report 2010 - environmental aspects- energy/global warming (p. 24-31). Technical report, Toyota, 2010. URL http://www.toyota.com.hk/images/sr10_p24_p31_tcm306.pdf.

[7] D. Althoff, M. Althoff, D. Wollherr, and M. Buss. Probabilistic collision state checker for crowded environments. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1492 –1498, may 2010.

[8] M. Althoff, O. Stursberg, and M. Buss. Verification of uncertain embedded systems by computing reachable sets based on zonotopes. In *Proc. of the 17th IFAC World Congress*, pages 5125–5130, 2008.

[9] M. Althoff, D. Althoff, D. Wollherr, and M. Buss. Safety verification of autonomous vehicles for coordinated evasive maneuvers. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 1078 –1083, june 2010.

[10] R. Alur, C. Courcoubetis, T. A. Henzinger, and P.-H. Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, editors, *Hybrid Systems (HSCC'92)*, volume 736 of *Lecture Notes in Computer Science*, pages 209–229. Springer-Verlag, 1993.

[11] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138:3–34, 1995.

[12] B. Asadi and A. Vahidi. Predictive cruise control: Utilizing upcoming traffic signal information for improving fuel economy and reducing trip time. *Control Systems Technology, IEEE Transactions on*, 19(3):707–714, 2011.

[13] E. Asarin, T. Dang, and O. Maler. d/dt: a verification tool for hybrid systems. In *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, volume 3, pages 2893 –2898 vol.3, 2001.

[14] K. Astrom and L. Rundqwist. Integrator windup and how to avoid it. In *Proceedings of the 1989 American Control Conference*, pages 1693–1698, 1989.

[15] I. Ben Makhlouf and S. Kowalewski. An evaluation of two recent reachability analysis tools for hybrid systems. In *2nd IFAC Conference on Analysis and Design of Hybrid Systems*, pages 377–382, 2006.

[16] I. Ben Makhlouf, S. Kowalewski, M. Chavez Grunewald, and D. Abel. Safety assessment of networked vehicle platoon controllers - practical experiences with available tools. In *3rd IFAC Conference on Analysis and Design of HybridSystems, Zaragoza, Spain*, pages 292–297, 2009.

[17] A. Benmimoun, J. Chen, and T. Suzuki. Design and practical evaluation of an intersection assistant in real world tests. In *Intelligent Vehicles Symposium, 2007 IEEE*, pages 606 –611, june 2007.

[18] R. Bishop. *Intelligent Vehicle Technology And Trends*. Artech House Publishers, 05 2005. ISBN 1580539114.

[19] S. Biswas, R. Tatchikou, and F. Dion. Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety. *Communications Magazine, IEEE*, 44(1):74–82, 2006.

[20] S. Brennan and A. Alleyne. Using a scale testbed: Controller design and evaluation. *Control Systems, IEEE*, 21(3):15 –26, jun 2001. ISSN 1066-033X.

[21] S. Brennan, A. Alleyne, and M. DePoorter. The illinois roadway simulator-a hardware-in-the-loop testbed for vehicle dynamics and control. In *American Control Conference, 1998. Proceedings of the 1998*, volume 1, pages 493 –497 vol.1, jun 1998.

[22] F. Browand, J. McArthur, and C. Radovich. Fuel saving achieved in the field test of two tandem trucks. Institute of transportation studies, research reports, working papers, proceedings, Institute of Transportation Studies, UC Berkeley, 2004.

[23] K. Chawla, G. Robins, and L. Zhang. Object localization using rfid. In *5th IEEE International Symposium on Wireless Pervasive Computing (ISWPC), 2010*, pages 301 –306, may 2010.

[24] A. Chutinan and B. Krogh. Computational techniques for hybrid system verification. *Automatic Control, IEEE Transactions on*, 48(1):64 – 75, jan 2003. ISSN 0018-9286.

[25] E. M. Clarke and J. M. Wing. Formal methods: State of the art and future directions. *ACM Computing Surveys*, 28:626–643, 1996.

[26] M. Conrad, I. Fey, M. Grochtmann, and T. Klein. Modellbasierte entwicklung eingebetteter fahrzeugsoftware bei daimlerchrysler. *Informatik Forschung und Entwicklung*, 20(1-2):31–41, 2004.

[27] S. Deutschle. Das KONVOI Projekt: Entwicklung und Untersuchung des Einsatzes von elektronisch gekoppelten Lkw-Konvois auf Autobahnen. *15. Aachener Kolloquium Fahrzeug- und Motorentechnik, Aachen*, 2006.

[28] K. Dresner and P. Stone. Turning the corner: improved intersection control for autonomous vehicles. In *Proc. IEEE Intelligent Vehicles Symp*, pages 423–428, 2005.

[29] K. Dresner and P. Stone. A multiagent approach to autonomous intersection management. *Journal of Artificial Intelligence Research*, 31:591–656, March 2008.

[30] P. Enge and P. Misra. Special issue on global positioning system. *Proceedings of the IEEE*, 87(1):3 –15, jan 1999. ISSN 0018-9219.

[31] A. Ferrara. Automatic pre-crash collision avoidance in cars. In *IEEE Intelligent Vehicles Symposium, 2004*, pages 133 – 138, june 2004.

[32] G. Frehse. PHAVer: Algorithmic verification of hybrid systems past hytech. In *HSCC 2005, LNCS 3414*, pages 258–273. Springer-Verlag, 2005.

[33] H. Fritz. Longitudinal and lateral control of heavy duty trucks for automated vehicle following in mixed traffic: experimental results from the chauffeur project. In *Proc Int. Conf. Contr. Applications*, pages 1348–1352, 1999.

[34] K. Fuerstenberg and B. Roessler. Intersection driver assistance system - results of the ec-project intersafe. In *IEEE Intelligent Vehicles Symposium, 2007*, pages 462 –467, june 2007.

[35] K. C. Fuerstenberg and B. Roessler. Advanced intersection safety - the ec project intersafe. In *Proc. IEEE Intelligent Vehicles Symp*, pages 89–93, 2006.

[36] A. Girard. Reachability of uncertain linear systems using zonotopes. In *Proceedings of the 8th international conference on Hybrid Systems: computation and control*, HSCC'05, pages 291–305, Berlin, Heidelberg, 2005. Springer-Verlag.

[37] C. L. Guernic and A. Girard. Reachability analysis of linear systems using support functions. *Nonlinear Analysis: Hybrid Systems,*, 4(2):250 – 262, 2010. ISSN 1751-570X.

[38] Z. Han and B. Krogh. Reachability analysis of hybrid control systems using reduced-order models. In *American Control Conference, 2004. Proceedings of the 2004*, volume 2, pages 1183 –1189 vol.2, 30 2004-july 2 2004.

[39] Z. Han and B. H. Krogh. Reachability analysis of large-scale affine systems using low-dimensional polytopes. In J. P. Hespanha and A. Tiwari, editors, *Hybrid Systems: Computation and Control, 9th International Workshop, HSCC 2006, Santa Barbara, CA, USA, March 29-31, 2006,*, volume 3927 of *Lecture Notes in Computer Science*, pages 287–301. Springer, 2006. ISBN 3-540-33170-0.

[40] T. Hecker, J. Zech, B. Schäufele, R. Gräfe, and I. Radusch. Model car testbed for development of v2x applications. *JCM*, 6(1):115–124, 2011.

[41] B. Heißing, M. Ersoy, and S. Gies. *Fahrwerkhandbuch: Grundlagen, Fahrdynamik, Komponenten, Systeme, Mechatronik, Perspektiven.* ATZ-MTZ Fachbuch. Vieweg+Teubner Verlag, 2011. ISBN 9783834808219.

[42] T. Henzinger, P. Ho, and H. Wong-Toi. Hytech: A model checker for hybrid systems. *Software Tools for Technology Transfer*, 1(1,2):110–122, 1997.

[43] T. Henzinger, P. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata. *Computer and Systems Sciences*, 1998.

[44] T. A. Henzinger and J. Sifakis. The embedded systems design challenge. In *Proceedings of the 14th International Symposium on Formal Methods (FM), Lecture Notes in Computer Science*, pages 1–15. Springer, 2006.

[45] P. Hoblet, J. O'Brien, R.T., and J. Piepmeier. Scale-model vehicle analysis for the design of a steering controller. In *System Theory, 2003. Proceedings of the 35th Southeastern Symposium on*, pages 201 – 205, 2003.

[46] A. S. Howell, A. R. Girard, J. K. Hedrick, and P. P. Varaiya. A real-time hierarchical software architecture for coordinated vehicle control. In *Automotive Software Architecture Workshop*, 2004.

[47] J. Huang, J. Ahmed, A. Kojic, and J.-P. Hathout. Control oriented modeling for enhanced yaw stability and vehicle steerability. In *American Control Conference, Proceedings of the 2004*, volume 4, pages 3405 –3410, july 2004.

[48] W. Jian, C. Xiumin, W. Yong, and Z. Rui. The design of autonomous smart car used in simulation of vehicle platoon. In *2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, pages 885–890, 2008.

[49] Y. Jiang, S. Li, and D. E. Shamo. Development of vehicle platoon distribution models and simulation of platoon movements indian rural corridors. *Joint Transportation Research Program Technical, Report Series, Purdue University, West Lafayette, Indiana*, 2003.

[50] L. Jing and P. Yang. A localization algorithm for mobile robots in rfid system. In *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on*, pages 2109 –2112, sept. 2007.

[51] W.-W. Kao. Integration of gps and dead-reckoning navigation systems. In *Vehicle Navigation and Information Systems Conference, 1991*, volume 2, pages 635 – 643, oct. 1991.

[52] D. Katzourakis, I. Papaefstathiou, and M. Lagoudakis. An open-source scaled automobile platform for fault-tolerant electronic stability control. *Instrumentation and Measurement, IEEE Transactions on*, 59(9):2303 –2314, sept. 2010. ISSN 0018-9456.

[53] J. Kolodko and L. Vlacic. Cooperative autonomous driving at the intelligent control systems laboratory. *Intelligent Systems, IEEE*, 18(4):8–11, 2003.

[54] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer. Multi-camera multi-person tracking for easyliving. In *Visual Surveillance, 2000. Proceedings. Third IEEE International Workshop on*, pages 3 –10, 2000.

[55] A. B. Kurzhanski and P. Varaiya. Ellipsoidal techniques for reachability analysis. In *HSCC'00: Proceedings of the Third International Workshop on Hybrid Systems: Computation and Control*, pages 202–214, London, UK, 2000. Springer-Verlag. ISBN 3-540-67259-1.

[56] A. M. Ladd, K. E. Bekris, A. Rudys, G. Marceau, L. E. Kavraki, and D. S. Wallach. Robotics-based location sensing using wireless ethernet. In *Proceedings of the 8th annual international conference on Mobile computing and networking*, MobiCom '02, pages 227–238, New York, NY, USA, 2002. ISBN 1-58113-486-X.

[57] G. Lafferriere, G. J. Pappas, and S. Yovine. A new class of decidable hybrid systems. In *Proceedings of the Second International Workshop on Hybrid Systems: Computation and Control*, HSCC '99, pages 137–151, London, UK, UK, 1999. Springer-Verlag. ISBN 3-540-65734-7.

[58] G. Lafferriere, G. J. Pappas, and S. Yovine. Symbolic reachability computation for families of linear vector fields. *Journal of Symbolic Computation*, 32:231–253, 2001.

[59] X. Liu, A. Goldsmith, S. Mahal, and J. Hedrick. Effects of communication delay on string stability in vehicle platoons. In *Intelligent Transportation Systems. Proceedings IEEE*, pages 625 –630, 2001.

[60] R. G. Longoria, A. Al-Sharif, and C. B. Patil. Scaled vehicle system dynamics and control: a case study in anti-lock braking. *International Journal Of Vehicle Autonomous Systems*, 2(1/2):18, 2004.

[61] S. Loos and A. Platzer. Safe intersections: At the crossing of hybrid systems and verification. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 1181 –1186, oct. 2011.

[62] H. Lu, S. Zhang, X. Liu, and X. Lin. Vehicle tracking using particle filter in wi-fi network. In *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd*, pages 1 –5, sept. 2010.

[63] X. Y. Lu and J. K. Hedrick. Heavy-duty vehicle modelling and longitudinal control. In *Vehicle System Dynamics*, volume 6, pages 653–669, 2005.

[64] X. Y. Lu, S. Shladover, and J. Hedrick. Heavy-duty truck control: short inter-vehicle distance following. In *American Control Conference, 2004. Proceedings of the 2004*, volume 5, pages 4722–4727 vol.5, 2004.

[65] J. Lygeros, D. N. Godbole, and S. Sastry. Verified hybrid controllers for automated vehicles. *IEEE Transactions on Automatic Control,*, 43(4):522–539, 1998.

[66] J.-L. Madre, M. André, C. Rizet, J. Leonardi, and P. Ottmann. Importance of the loading factor in transport co2 emissions. In *12th World Conference on Transport Research (WCTR), July 11-15, 2010, Lisbon, Portugal.*

[67] S. Mandava, K. Boriboonsomsin, and M. Barth. Arterial velocity planning based on traffic signal information under light traffic conditions. In *Intelligent Transportation Systems, 2009. ITSC '09. 12th International IEEE Conference on*, pages 1 –6, oct. 2009.

[68] J. P. Maschuw, G. C. Keßler, and D. Abel. LMI-based control of vehicle platoons for robust longitudinal guidance. In *IFAC World Congress 2008*, pages 12111–12116, Seoul, July 2008.

[69] D. McMahon, V. K. Narendran, D. Swaroop, J. Hedrick, K. Chang, and P. Devlin. Longitudinal vehicle controllers for IVHS: Theory and experiment. In *American Control Conference, 1992*, pages 1753–1757, 1992.

[70] G. Moitzi, T. Szalay, M. Schüller, H. Wagentristl, K. Refenner, H. Weingartmann, P. Liebhard, J. Boxberger, and A. Gronauer. Effects of tillage systems and mechanization on work time, fuel and energy consumption for cereal cropping in austria. *Agricultural Engineering International: CIGR Journal*, 15(4):94–101, 2013.

[71] C. Moler and C. V. Loan. Nineteen dubious ways to compute the exponential of a matrix. *SIAM Review*, 1978.

[72] N. Neuendorf and T. Bruns. The vehicle platoon controller in the decentralised, autonomous intersection management of vehicles. In *Proc. IEEE Int. Conf. Mechatronics ICM '04*, pages 375–380, 2004.

[73] H. Nishiuchi and T. Yoshii. A study of the signal control for the minimization of CO2 emission. *12th World Congress on Intelligent Transport Systems proceedings*, 2005.

[74] C. of the European Communities. Keep europe moving - sustainable mobility for our continent. Brussels, 2006.

[75] P. Papadimitratos, A. La Fortelle, K. Evenssen, R. Brignolo, and S. Cosenza. Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation. *Communications Magazine, IEEE*, 47(11):84 –95, november 2009. ISSN 0163-6804.

[76] L. Perko. *Differential Equations and Dynamical Systems.* Springer; 3rd edition, 2006.

[77] A. Puri and P. Varaiya. Decidability of hybrid systems with rectangular differential inclusions. In *Proceedings of the 6th International Conference on Computer Aided Verification (CAV 1994)*, pages 95–104. Springer-Verlag, 1994.

[78] R. Ramakers, K. Henning, S. Gies, D. Abel, and H. Max. Electronically coupled truck platoons on german highways. In *IEEE International Conference on Systems, Man and Cybernetics, 2009. SMC 2009.*, pages 2409 –2414, oct. 2009.

[79] S. Ratschan and Z. She. Safety verification of hybrid systems by constraint propagation based abstraction refinement. In M. Morari and L. Thiele, editors, *Hybrid Systems: Computation and Control, 8th International Workshop, HSCC 2005, Zurich, Switzerland, March 9-11, 2005, Proceedings*, volume 3414 of *Lecture Notes in Computer Science*, pages 573–589. Springer, 2005. ISBN 3-540-25108-1.

[80] S. Ratschan and Z. She. Constraints for continuous reachability in the verification of hybrid systems. In *Proc. 8th Int. Conf. on Artif. Intell. and Symb. Comp., AISC'2006*, number 4120 in LNCS, pages 196–210. Springer, 2006.

[81] T. Sakaguchi, A. Uno, S. Kato, and S. Tsugawa. Cooperative driving of automated vehicles with inter-vehicle communications. In *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*, pages 516 –521, 2000.

[82] J. Sasiadek and Q. Wang. Sensor fusion based on fuzzy kalman filtering for autonomous robot vehicle. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 4, pages 2970 –2975, 1999.

[83] A. Sassi, F. Charfi, L. Kamoun, Y. Elhillali, and A. Rivenq. The impact of mobility on the performance of v2x communication. In *Proc. 4th Int Logistics (LOGISTIQUA) Conf*, pages 434–438, 2011.

[84] F. Saust, O. Bley, R. Kutzner, J. M. Wille, B. Friedrich, and M. Maurer. Exploitability of vehicle related sensor data in cooperative systems. In *Proc. 13th Int Intelligent Transportation Systems (ITSC) IEEE Conf*, pages 1724–1729, 2010.

[85] A. Schmitt. *Beitrag zur NOx Emissionsminderung für Niedrig-Emissions-Fahrzeuganwendungen mittels Selektiver-Katalytischer-Reduktion*. PhD thesis, TU Darmstadt, 2013. URL http://tuprints.ulb.tu-darmstadt.de/3544/.

[86] D. Schrank and T. Lomax. The 2007 urban mobility report. *Transportation*, September, 2007.

[87] S. Sheikholeslam and C. Desoer. Longitudinal control of a platoon of vehicles with no communication of lead vehicle information: A system level study. *Vehicular Technology, IEEE Transactions on*, 42(4):546–554, 1993. ISSN 0018-9545.

[88] S. Shladover, C. Desoer, J. Hedrick, M. Tomizuka, J. Walrand, W.-B. Zhang, D. McMahon, H. Peng, S. Sheikholeslam, and N. McKeown. Automated vehicle control developments in the path program. *Vehicular Technology, IEEE Transactions on*, 40(1):114 –130, feb 1991. ISSN 0018-9545.

[89] S. E. Shladover. Longitudinal control of automated guideway transit vehicles within platoons. *Journal of Dynamic Systems, Measurement, and Control*, 100(4):302–310, 1978.

[90] M. Sichitiu and M. Kihl. Inter-vehicle communication systems: A survey. *Communications Surveys Tutorials, IEEE*, 10(2):88 –105, quarter 2008. ISSN 1553-877X.

[91] A. Smith, H. Balakrishnan, M. Goraczko, and N. B. Priyantha. Tracking Moving Devices with the Cricket Location System. In *2nd International Conference on Mobile Systems, Applications and Services (Mobisys 2004)*, Boston, MA, June 2004.

[92] A. Stollenwerk, A. Derks, S. Kowalewski, and F. Salewski. A modular, robust and open source microcontroller platformfor broad educational usage. In *Workshop on Embedded Systems Education (WESE10), Scottsdale,AZ, USA*, pages 48–54, October 2010.

[93] O. Stursberg, A. Fehnker, Z. Han, and B. H. Krogh. Verification of a cruise control system using counterexample-guided search. *Control Engineering Practice*, 12(10): 1269 – 1278, 2004. ISSN 0967-0661.

[94] D. Swaroop and J. Hedrick. Direct adaptive longitudinal control of vehicle platoons. In *Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on*, volume 1, pages 684 –689 vol.1, dec 1994.

[95] D. Swaroop and J. Hedrick. String stability of interconnected systems. *IEEE Transactions on Automatic Control*, 41:349–357, 1996.

[96] T. Tielert, M. Killat, H. Hartenstein, R. Luz, S. Hausberger, and T. Benz. The impact of traffic-light-to-vehicle communication on fuel consumption and emissions. In *Internet of Things (IOT), 2010*, pages 1 –8, dec. 2010.

[97] W. Travis, R. Whitehead, D. Bevly, and G. Flowers. Using scaled vehicles to investigate the influence of various properties on rollover propensity. In *American Control Conference, Proceedings of the 2004*, volume 4, pages 3381 –3386, july 2004.

[98] S. Tsugawa and S. Kato. Energy ITS: another application of vehicular communications. *Communications Magazine, IEEE*, 48(11):120 –126, november 2010. ISSN 0163-6804.

[99] S. Tsugawa, S. Kato, and K. Aoki. An automated truck platoon for energy saving. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference*, pages 4109 –4114, sept. 2011.

[100] A. van der Schaft and J. Schumacher. *An Introduction to Hybrid Dynamical Systems*. Lecture Notes in Control and Information Sciences. Springer, 2000. ISBN 9781852332334.

[101] R. Verma, D. Del Vecchio, and H. Fathy. Development of a scaled vehicle with longitudinal dynamics of an HMMWV for an ITS testbed. *Mechatronics, IEEE/ASME Transactions*, 13(1):46 –57, feb. 2008. ISSN 1083-4435.

[102] M. Wada, T. Yendo, T. Fujii, and M. Tanimoto. Road-to-vehicle communication using led traffic light. In *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pages 601 – 606, june 2005.

[103] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 10(1):91–102, Jan. 1992.

[104] G. Welch and G. Bishop. SCAAT: incremental tracking with incomplete information. In *SIGGRAPH*, pages 333–344, 1997.

[105] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4), Dec. 2006. ISSN 0360-0300.

[106] M. Youssef, A. Agrawala, and A. Udaya Shankar. Wlan location determination via clustering and probability distributions. In *Pervasive Computing and Communications, 2003. (PerCom 2003). Proceedings of the First IEEE International Conference on*, pages 143 – 150, march 2003.

Supervised Graduation Thesis

[Hil11] Benjamin Hilgers. Unterstützung eines fahrzeuginternen ortungssystems durch sensorfusion. Diplomarbeit, Chair of Computer Science 11 , RWTH Aachen, 2011.

[Pop12] Dominik Poppe. Erstellung und evaluation einer hardwareplattform zur präzisen ladezustandserfassung von blei-gel-akkumulatoren. Bachelor's thesis, Chair of Computer Science 11 , RWTH Aachen, 2012.

[Tar12] Ali Tarzan. Safe crossing of an intersection with a platoon of vehicles. Bachelor's thesis, Chair of Computer Science 11 , RWTH Aachen, 2012.

[Tho12] Lars Tholen. Efficient crossing of a signalized intersection with a platoon of vehicles. Bachelor's thesis, Chair of Computer Science 11 , RWTH Aachen, 2012.

Own Publications

[BMDK12] Ibtissem Ben Makhlouf, Hilal Diab, and Stefan Kowalewski. Safety verification of a controlled cooperative platoonunder loss of communication using zonotopes. In *Inproceeding of the 4th IFAC Conference on Analysis and Design of Hybrid Systems (ADHS 12)*, pages 333–338, 2012.

[BMDK13] Ibtissem Ben Makhlouf, Hilal Diab, and Stefan Kowalewski. Reachability analysis for managing platoons at intersections. In *MED'13: the 21st Mediterranean Conference on Control andAutomation, Chania, Crete, Greece June 25-28*, 2013.

[BMMH+11] Ibtissem Ben Makhlouf, Jan Philipp Maschuw, Paul Hänsch, Hilal Diab, Stefan Kowalewski, and Dirk Abel. Safety verification of a cooperative vehicle platoon with uncertain inputs using zonotopes. In *18th IFAC World Congress, 2011, Milano, Italy*, pages 9769–9774. IFAC, 2011.

[CGBMD+10a] Martín Chávez Grunewald, Ibtissem Ben Makhlouf, Hilal Diab, Dirk Abel, and Stefan Kowalewski. On the effects of network delays on an energy-based controller. In *2nd IFAC Workshop on Distributed Estimation and Control inNetworked Systems(NecSys' 2010)*, pages 169–174, September 2010.

[CGBMD+10b] Martín Chávez Grunewald, Ibtissem Ben Makhlouf, Hilal Diab, Vicente Mut, Stefan Kowalewski, and Dirk Abel. Regelung und sicherheitsanalyse einer gruppe massenpunktfahrzeuge mit hilfe energiebasierter methoden. *Automatisierungstechnik*, 58(4):227–235, 2010.

[DBMK12] Hilal Diab, Ibtissem Ben Makhlouf, and Stefan Kowalewski. A platoon of vehicles approaching an intersection: A testing platform for safe intersections. In *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pages 1918–1923, 2012.

[DCGBM+10] Hilal Diab, Martín Chávez Grunewald, Ibtissem Ben Makhlouf, Dirk Abel, and Stefan Kowalewski. A testing platform for cooperative vehicle platoon controllers. In *13th International IEEE Conference on Intelligent Transportation Systems (ITSC 2010)*, Madeira Island, Portugal, September 2010.

[DFDK11] Christian Dernehl, Dominik Franke, Hilal Diab, and Stefan Kowalewski. An architecture with integrated image processing for autonomous micro aerial vehicles. In *International Micro Air Vehicle Conference (IMAV)*, pages 138 – 145. IMAV, 2011.

[HDMK13] Paul Hänsch, Hilal Diab, Ibtissem Ben Makhlouf, and Stefan Kowalewski. Reachability analysis of linear systems with stepwise constant inputs. *Electronic Notes in Theoretical Computer Science*, 297(0):61 – 74, 2013. Proceedings of the first workshop on Hybrid Autonomous Systems.

[MDAK11] Jan P. Maschuw, Hilal Diab, Dirk Abel, and Stefan Kowalewski. Control design for generalized platoon problems. *Automatisierungstechnik*, 59(12):721–729, 2011.

Aachener Informatik-Berichte

This list contains all technical reports published during the past three years. A complete list of reports dating back to 1987 is available from:

http://aib.informatik.rwth-aachen.de/

To obtain copies please consult the above URL or send your request to:

Informatik-Bibliothek, RWTH Aachen, Ahornstr. 55, 52056 Aachen,
Email: biblio@informatik.rwth-aachen.de

2012-01	Fachgruppe Informatik: Annual Report 2012
2012-02	Thomas Heer: Controlling Development Processes
2012-03	Arne Haber, Jan Oliver Ringert, Bernhard Rumpe: MontiArc - Architectural Modeling of Interactive Distributed and Cyber-Physical Systems
2012-04	Marcus Gelderie: Strategy Machines and their Complexity
2012-05	Thomas Ströder, Fabian Emmes, Jürgen Giesl, Peter Schneider-Kamp, and Carsten Fuhs: Automated Complexity Analysis for Prolog by Term Rewriting
2012-06	Marc Brockschmidt, Richard Musiol, Carsten Otto, Jürgen Giesl: Automated Termination Proofs for Java Programs with Cyclic Data
2012-07	André Egners, Björn Marschollek, and Ulrike Meyer: Hackers in Your Pocket: A Survey of Smartphone Security Across Platforms
2012-08	Hongfei Fu: Computing Game Metrics on Markov Decision Processes
2012-09	Dennis Guck, Tingting Han, Joost-Pieter Katoen, and Martin R. Neuhäußer: Quantitative Timed Analysis of Interactive Markov Chains
2012-10	Uwe Naumann and Johannes Lotz: Algorithmic Differentiation of Numerical Methods: Tangent-Linear and Adjoint Direct Solvers for Systems of Linear Equations
2012-12	Jürgen Giesl, Thomas Ströder, Peter Schneider-Kamp, Fabian Emmes, and Carsten Fuhs: Symbolic Evaluation Graphs and Term Rewriting — A General Methodology for Analyzing Logic Programs
2012-15	Uwe Naumann, Johannes Lotz, Klaus Leppkes, and Markus Towara: Algorithmic Differentiation of Numerical Methods: Tangent-Linear and Adjoint Solvers for Systems of Nonlinear Equations
2012-16	Georg Neugebauer and Ulrike Meyer: SMC-MuSe: A Framework for Secure Multi-Party Computation on MultiSets
2012-17	Viet Yen Nguyen: Trustworthy Spacecraft Design Using Formal Methods
2013-01 *	Fachgruppe Informatik: Annual Report 2013
2013-02	Michael Reke: Modellbasierte Entwicklung automobiler Steuerungssysteme in Klein- und mittelständischen Unternehmen
2013-03	Markus Towara and Uwe Naumann: A Discrete Adjoint Model for OpenFOAM

2013-04	Max Sagebaum, Nicolas R. Gauger, Uwe Naumann, Johannes Lotz, and Klaus Leppkes: Algorithmic Differentiation of a Complex C++ Code with Underlying Libraries
2013-05	Andreas Rausch and Marc Sihling: Software & Systems Engineering Essentials 2013
2013-06	Marc Brockschmidt, Byron Cook, and Carsten Fuhs: Better termination proving through cooperation
2013-07	André Stollenwerk: Ein modellbasiertes Sicherheitskonzept für die extrakorporale Lungenunterstützung
2013-08	Sebastian Junges, Ulrich Loup, Florian Corzilius and Erika Ábrahám: On Gröbner Bases in the Context of Satisfiability-Modulo-Theories Solving over the Real Numbers
2013-10	Joost-Pieter Katoen, Thomas Noll, Thomas Santen, Dirk Seifert, and Hao Wu: Performance Analysis of Computing Servers using Stochastic Petri Nets and Markov Automata
2013-12	Marc Brockschmidt, Fabian Emmes, Stephan Falke, Carsten Fuhs, and Jürgen Giesl: Alternating Runtime and Size Complexity Analysis of Integer Programs
2013-13	Michael Eggert, Roger Häußling, Martin Henze, Lars Hermerschmidt, René Hummen, Daniel Kerpen, Antonio Navarro Pérez, Bernhard Rumpe, Dirk Thißen, and Klaus Wehrle: SensorCloud: Towards the Interdisciplinary Development of a Trustworthy Platform for Globally Interconnected Sensors and Actuators
2013-14	Jörg Brauer: Automatic Abstraction for Bit-Vectors using Decision Procedures
2013-16	Carsten Otto: Java Program Analysis by Symbolic Execution
2013-19	Florian Schmidt, David Orlea, and Klaus Wehrle: Support for error tolerance in the Real-Time Transport Protocol
2013-20	Jacob Palczynski: Time-Continuous Behaviour Comparison Based on Abstract Models
2014-01 *	Fachgruppe Informatik: Annual Report 2014
2014-02	Daniel Merschen: Integration und Analyse von Artefakten in der modellbasierten Entwicklung eingebetteter Software
2014-03	Uwe Naumann, Klaus Leppkes, and Johannes Lotz: dco/c++ User Guide
2014-04	Namit Chaturvedi: Languages of Infinite Traces and Deterministic Asynchronous Automata
2014-05	Thomas Ströder, Jürgen Giesl, Marc Brockschmidt, Florian Frohn, Carsten Fuhs, Jera Hensel, and Peter Schneider-Kamp: Automated Termination Analysis for Programs with Pointer Arithmetic
2014-06	Esther Horbert, Germán Martín García, Simone Frintrop, and Bastian Leibe: Sequence Level Salient Object Proposals for Generic Object Detection in Video

2014-07	Niloofar Safiran, Johannes Lotz, and Uwe Naumann: Algorithmic Differentiation of Numerical Methods: Second-Order Tangent and Adjoint Solvers for Systems of Parametrized Nonlinear Equations
2014-08	Christina Jansen, Florian Göbe, and Thomas Noll: Generating Inductive Predicates for Symbolic Execution of Pointer-Manipulating Programs
2014-09	Thomas Ströder and Terrance Swift (Editors): Proceedings of the International Joint Workshop on Implementation of Constraint and Logic Programming Systems and Logic-based Methods in Programming Environments 2014
2014-14	Florian Schmidt, Matteo Ceriotti, Niklas Hauser, and Klaus Wehrle: HotBox: Testing Temperature Effects in Sensor Networks
2014-15	Dominique Gückel: Synthesis of State Space Generators for Model Checking Microcontroller Code
2014-16	Hongfei Fu: Verifying Probabilistic Systems: New Algorithms and Complexity Results
2015-01 *	Fachgruppe Informatik: Annual Report 2015
2015-05	Florian Frohn, Jürgen Giesl, Jera Hensel, Cornelius Aschermann, and Thomas Ströder: Inferring Lower Bounds for Runtime Complexity
2015-06	Thomas Ströder and Wolfgang Thomas (Editors): Proceedings of the Young Researchers' Conference "Frontiers of Formal Methods"

* These reports are only available as a printed version.
Please contact biblio@informatik.rwth-aachen.de to obtain copies.