Czech Technical University in Prague

Faculty of Electrical Engineering

MASTER'S THESIS

Convoy platooning using vehicle to vehicle communication

2015 Bc. MICHAL ŠVANDRLÍK

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Control Engineering

DIPLOMA THESIS ASSIGNMENT

Student: **Bc. Michal Švandrlík**

Study programme: Cybernetics and Robotics
Specialisation: Systems and Control

Title of Diploma Thesis: **Convoy platooning using vehicle to vehicle communication**

Guidelines:

1. Explore current wireless networking technologies and their usability for vehicle to vehicle (V2V) communication.
2. Evaluate suitability of common communication devices for V2V communication and compare them with available devices for V2V.
3. Create a controller of convoy platooning which will use V2V communication.
4. Verify the influence of convoy controlled via V2V on traffic using simulation at normal and also poor traffic conditions.
5. Consider an effect of convoy length, percentual amount of vehicles using V2V, splitting and linking of convoy, traffic density, penetration rate of communication technology etc.

Bibliography/Sources:

[1] Olfati-Saber, R.; Fax, J.A.; Murray, R.M., Consensus and Cooperation in Networked Multi-Agent Systems, Proceedings of the IEEE , vol.95, no.1, pp.215,233, Jan. 2007
[2] Russell, Stuart, Peter Norvig, and Artificial Intelligence. A modern approach. Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs 25 (1995)
[3] Stanica, Razvan; Chaput, Emmanuel; Beylot, André-Luc: Simulation of vehicular ad-hoc networks: Challenges, review of tools and recommendations

Diploma Thesis Supervisor: doc. Jiří Vokřínek Ing., Ph.D.

Valid until the summer semester 2015/2016

prof. Ing. Michael Šebek, DrSc.
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, January 29, 2015

Prohlášení autora práce

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 7. 5. 2015

podpis

Poděkování

Tímto bych rád poděkoval všem lidem, kteří mě podporovali a bez kterých by tato práce nemohla vzniknout. Především bych chtěl poděkovat vedoucímu práce doc. Ing. Jiřímu Vokřínkovi PhD. za vedení, cenné rady a motivaci. Dále bych chtěl poděkovat zadavateli práce Ing. Josefu Fulemovi PhD. za příležitost pracovat na zajímavém a aktuálním tématu. Můj dík patří též Ing. Martinu Schaeferovi za ochotnou podporu při seznamování a práci se simulátorem. Nakonec bych chtěl poděkovat své rodině za pochopení, laskavý přístup a zázemí, které mi poskytli k tvorbě a studiu.

Abstract

This thesis topic is to study the impact of vehicle to vehicle communication (V2V) used for convoy platooning control and its performance in a traffic. The main goal of V2V communication is increasing safety of traffic by sharing position, velocity and other useful information. It also aims to prevent from traffic jams and reduce the number of accidents. Our task is to evaluate the influence of convoy control via V2V under different conditions and performing maneuvers like acceleration, deceleration and zipper. The metrics like V2V technology penetration, convoy length and traffic density are taken into account. The experiments are realized via computer simulation.

Keywords: *string stability, convoy, V2V, V2I, simulation, agent*

Abstrakt

Diplomová práce pojednává o vlivu technologie komunikace typu vozidlo – vozidlo (V2V) při řízení skupiny vozidel a porovnání jeho úspěšnosti. Hlavním cílem užití V2V je zvýšení bezpečnosti provozu pomocí sdílení polohy, rychlosti a dalších užitečných dat. Dále si klade za cíl snížení počtu dopravních zácp a nehod. Naším cílem je ohodnotit vliv řízení konvoje pomocí V2V za různých dopravních podmínek provádějící manévry akcelerace, brzdění a zipování. V potaz budou brány parametry jako délka konvoje, hustota provozu a počet vozidel vybavených technologií V2V.

Klíčová slova: *string stability, konvoj, V2V, V2I, simulace, agent*

Contents

List of Figures

List of Tables

Chapter 1

Introduction

At the beginning of 21st century there was a huge increase of active safety components used in all kinds of vehicles. Many devices were added into vehicles to extent their sensing capability. As an example we can name radar sensors used along with adaptive cruise control, camera with detection and recognition algorithms for sensing pedestrians etc. Only problem was that those devices had the same “range of vision” as the driver [4]. The vehicle to vehicle communication (V2V) is supposed to reduce this problem and extend drivers view. V2V devices will likely be deployed inside vehicles in few years in EU and USA. Some manufacturers already made their cars V2V capable (e. g. BMW). Some after-market devices are being developed to be used in cars without V2V technology already incorporated. The main goal of V2V communication is increasing safety of traffic, accident and traffic jam reduction.

Driving in convoy has many advantages. Namely traffic congestion reduction, fuel saving, increase of traffic safety etc. Nowadays cars equipped with adaptive cruise control are able to keep safe distance from previous car automatically. This control strategy is called leader – follower. However steering still depends on the driver. Moreover with increasing length of such convoy, the problems with string stability occurs. Such control strategy couldn't be used in dense traffic. This is an area where V2V could play an important role. While sharing information about current vehicle status, well designed controller could minimize oscillations of vehicles in convoy. Moreover if vehicles were able to move autonomously, driver could use time spent on road more efficiently for work, relaxation or entertainment.

This thesis couldn't be made without simulation tools. It is inexpensive way how to model traffic situations without using costly equipment and risk its destruction. Another reason of using simulations is national legislation and the fact, that V2V devices are still not available on the market. Other benefits of using simulations are repeatability and scalability of experiments.

1. 1. Requirements

This section presents requirements and specification of the assignment.

1. Explore current wireless networking technologies and their usability for vehicle to vehicle (V2V) communication.

2. Evaluate suitability of common communication devices for V2V communication and compare them with available devices for V2V.

3. Explore various simulation tools for traffic and communication simulations.

4. Create a controller of convoy platooning which will use V2V communication.

5. Verify the influence of convoy controlled via V2V using simulation at normal and also poor traffic conditions.

6. Consider an effect of convoy length, percentage amount of vehicles using V2V, splitting and linking of convoy, traffic density, penetration rate and other variables affecting a quality of communication technology etc.

1. 2. Thesis organization

In this section we provide a reader with a structure of this thesis. In Chapter 2, the Intelligent Transportation Systems (ITS) are presented and their main parts, which are vehicle to vehicle (V2V) and vehicle to infrastructure (V2I) communications. Aspects like communication reliability, efficiency will be also discussed there. We also introduce simulation tools for vehicular transportation and communication. The last but not least the opportunities for usage of V2V will be presented. Chapter 3 deals with convoy string stability. We also present implementation specifics of chosen simulator and our implementation of message management and control law. In Chapter 4 we define testing methodology and scenarios to be simulated and evaluate the results. Thesis is concluded in Chapter 5.

Chapter 2

ITS communication technology

In this chapter we will focus on technology behind “Intelligent Transportation Systems” (ITS). The communication standard will be explored and also its properties like efficiency, reliability and safety will be discussed. Another topic of this chapter will be dealing with a current research, where could ITS is involved. The last topic explores simulation tools.

2. 1. Communication for ITS

Generally the vehicular communication system is ad hoc network with vehicles and infrastructure nodes. The network operates in 5.9 GHz band with 30 MHz bandwidth for road safety and traffic management in Europe. Another 20 MHz should be assigned for business oriented applications. In USA there is 75 MHz bandwidth dedicated for ITS. The V2V or V2I device should be capable to communicate with others within a circle of 300 to 500 meters of radius.

A new Wireless LAN IEEE standard 802.11p for Wireless Access in Vehicular Environment (WAVE) was developed (USA). In EU there is corresponding standard called ITS-G5. Physical layer (PHY) of IEEE 802.11p is as an extension of IEEE 802.11a modified for extended communication range by adjusting transmitting power level and reducing the bandwidth from 20 MHz to 10 MHz only. Medium Access Control (MAC) is based on IEEE 802.11 standard with simplified version of “career sense multiple access with collision avoidance” (CSMA/CA) [5], [6].

2. 1. 1. European standard for ITS

The ETSI (European Telecommunication Standards Institute) started to work on European telecommunication standard for ITS in January 2008. The main goal was to develop standard for layers of the ITS reference model [5]. The basic structure of ITS model is shown on Figure 1. The standard has been divided into six layers and we will briefly discus their functionality

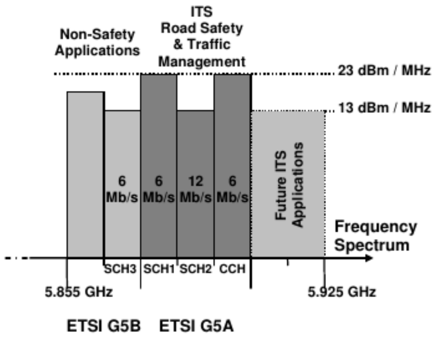


Figure 1: ITS frequency allocation in
Europe [6]

- Access – standards for the layer 1 and layer 2 of the OSI reference model (physical layer, link layer). The PHY secures physical connection to communication medium. The link layer can be divided to MAC, which manages the access to communication medium, and a logical link control sub-layer (LLC).

- Networking & Transport – standards for the layer 3 and layer 4 of the OSI reference model (network and transport). Network layer contains several possible networking models (e. g. GeoNetworking, different ways of IPv6, CALM FAST, etc.). Transport layer contains several different transport protocols (e. g. TCP, UDP, ITS specific, etc.). This layer uniting also contains its own management layer.

- Facilities – standards for functionality of OSI application, presentation (encoding, decoding, encryption) and session (e. g. inter–host communication) layers. This layer contains application, information, communication and session support and as previous ones its own management layer. Facilities may also include generic HMI support (presenting information to the driver), position and time support (provides information about geographical position of the ITS station and actual time), location referencing and time stamping of data, Local Dynamic Map (LDM – discussed later), message management (e. g. CAM and DNM messages – discussed later) and many others.

- Management layer – is management entity which communicates and manages other management sub-layers mentioned above. Also provides cross-interface management, inter-unit management communications, network, communication service, ITS application and station management etc.

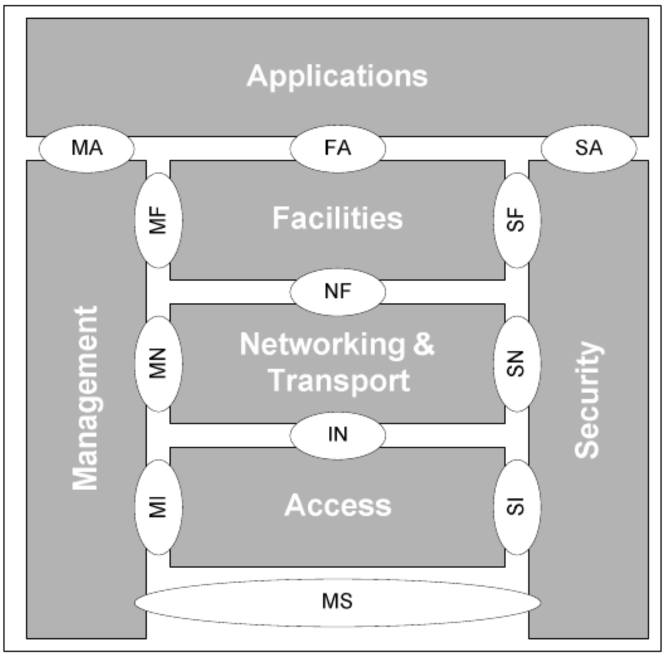


Figure 2: ITS communication reference model [8]

- Security layer – contains 25 security functionalities related to ITS communication protocol stack (e. g. firewall, authentication, authorization and profile management, hardware security modules etc.).

Detailed information about communication protocols can be found in [8], namely in standard ETSI EN 302 665 V1.1.1 – Intelligent Transport Systems, Communications Architecture.

2. 1. 2. Message management

As mentioned before, the ETSI standard for ITS supports message management at facility layer. There are two main types of standard messages proposed for use in ITS.

- The Co-operative Awareness Messages (CAM) – short messages exchanged in the ITS network, which are constantly broadcasted by all ITS stations in network (frequency between 10 Hz to 0.2 Hz, depends on senders state change and channel load). This message contains status and attribute information of the originating ITS station. In case of vehicle, information consists of status information like position, velocity, heading, motion state, etc. The attribute information contains parameters of vehicle namely type, dimensions, role in traffic. Those information aim at increasing awareness between individual vehicles and support cooperative performance of vehicles using road network. Length of CAM is around 100 – 200 bytes [6].

- The Decentralized environmental Notification Messages (DNM) – short messages that are broadcasted in order to alert road users to an event. This event suppose to report some hazardous situation (e. g. immobilized vehicle, bad weather, traffic problems, dangerous drivers etc.). When the event is detected, the ITS station (denoted as originating ITS st.) transmits DNM to spread the information about event to other ITS stations inside the affected area. DNM can be forwarded by another ITS (denoted as forwarding IST st.) to reach to every vehicle in specified area. Message may be either sent once or be repeated, until the event is terminated or predefined time expires. Length of the DNM depends on message content so it may vary.

These messages will help driver to react properly to the current traffic situation through creating some driver awareness or even act directly using control systems [5], [8].

2. 1. 3. Local Dynamic Map

A Local Dynamic Map (LDM) is a facility in cooperative ITS. It supports applications by maintaining information on objects influencing or influenced by road traffic [8]. The LDM can provide storage of information about surrounding static or moving object equipped with ITS station (e. g. vehicles, roadside stations, pedestrians with personal ITS st. etc.). The main sources of the data for LDM are CAM and DNM services and on-board sensors. LDM provides access to stored information to any authorized application which request the data. The information stored in LDM can be accessed in form of so called LDM Data Object. LDM also provides additional functionalities such as:

- Registration/Deregistration of LDM data providers/subscribers via security layer.
- Subscribe/Unsubscribe for notifications.
- Information retention by applying rules, e.g. based on time and/or location.
- Prioritization of requests [8].

In summary the LDM manages and updates data about local environment and provides secured access to information contained in data. For our purposes, this is one of the most important services provided by standard for ITS.

2. 2. Efficiency, reliability and safety

However there are some doubts about efficiency of new standard. There are two main issues mentioned in many papers. The first one is poor performance at PHY layer and the second is absence of an upper bound of channel access delay for time critical data at MAC layer. In this section we will briefly discuss both problems.

2. 2. 1. Physical layer performance

Since based on older IEEE 802.11a, which was originally created for indoor use and static environment, the performance of IEEE 802.11p in vehicular dynamic environment might be lower than usual. The main issue seems to be limited bandwidth for safety applications. As mentioned before, in the USA, there is 75 MHz of bandwidth dedicated for ITS services, but only 10 MHz are intended to critical road safety applications. The same situation is in EU where 10 MHz out of 30 MHz are reserved for control channel (CCH) [9]. The 802.11p PHY is based on Orthogonal Frequency Division Multiplexing (OFDM) which allows to communicate at variety of rates (from 3 to 27 Mb/s) dependent on used signal modulation and its coding rate [6]. The frequency allocation for EU is shown in Figure 1. A limited bandwidth of CCH is the main topic of objection in various papers.

Iulia Ivan et al. made research in this field and summarized it to [6]. Authors used a set of simulations to analyze IEEE 802.11p physical performance. The metrics of performance are package error rate (PER) and signal to noise ratio (SNR). Expected result was that with increasing transmission speed also PER increases and for its improvement the greater SNR is required. With increasing speed of communicating vehicle the situation gets even worse thanks to Doppler effect and other channel changes. The last issue is the packet length where larger packet is more probable to get lost or being received damaged.

2. 2. 2. MAC layer reliability

In safety applications for traffic safety and management, the real time access to data describing surrounding environment is critical. Especially while driving on the highway, where a few seconds of delay can cause significant difference between received data and real state. The main issue is that there is no upper bound for channel access delay in CSMA/CA. Transmitting device would always delay package when the channel is detected as busy. There might appear situation that package would not be sent before next package arrives. Another issue is only half duplex characteristic of the channel. The station couldn't receive message while transmitting. This could lead to significant delay where the vehicle would lost an information about the others.

In [9] authors discuss these problems and compare CSMA/CA with another existing protocol called “self-organizing time division multiple access” (STDMA) in urban highway scenario. STDMA protocol was originally proposed for maritime traffic. It has a synchronized time slot structure and ensures a predictable channel access delay even in a congested network. The station listens to channel activity and searches for free slots in selection interval (SI). Then unit choose some slots to send its message. The channel access delay is determined by the SI length [9]. The results are that STDMA is slightly better at limited bandwidth (especially in 10 MHz used by IEEE 802.11p) but with increasing bandwidth the CSMA/CA outperform STDMA. Authors came to 80 MHz of bandwidth to be sufficient to achieve 99 % reliability of the system according to their simulations.

2. 2. 3. Possible replacement with LTE

Another proposed solution is replacing IEEE 802.11p with Long Term Evolution (LTE) standard by 3GPP (3rd Generation Partnership Program). Using widespread existing infrastructure of the LTE a better reliability and scalability could be achieved. Studies [12] and [13] deals with comparison of IEEE 802.11p and LTE communication in non line of sight (NLOS) situations, typical for cities, using simulation.

In [13] authors focus on physical layer performance in situation where vehicle is in NLOS position and enters the intersection. The metrics of performance is block error rate (BLER) and communication is considered reliable when BLER is lower than 0.1. The Figure 3 shows simulation results where LTE outperforms IEEE 802.11p in communication range beyond the line of sight.

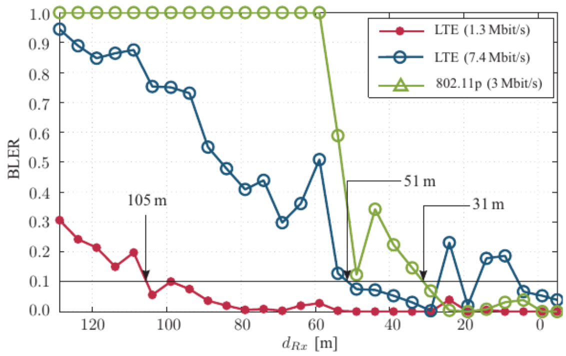


Figure 3: Simulation results for BLER [13]

In contrast to [13], authors of [12] focused on different conditions and investigated influence of a traffic density and different transmission frequencies. They used three metrics to examine the performance of communication technologies.

- End-to-End Delay, computed as the sum of all mean delays for each vehicle, normalized over the total number of flows in the network. Where, mean delay is defined as the ratio between the sums of all delays and the total number of received packets.
- Packet Delivery Ratio (PDR), computed as the ratio between the number of received packets and the transmitted packets during the simulation time.
- Throughput, is defined as sum of received data frame bytes at the destinations, averaged over the total number flows in the network.

The simulation of traffic was provided by SUMO (Simulation of Urban MObility) simulator and communication is simulated using ns3 simulator. The road network represents urban scenario (5x5 Manhattan grid – 25 blocks). The simulation results are displayed on Figure 4 and Figure 5.

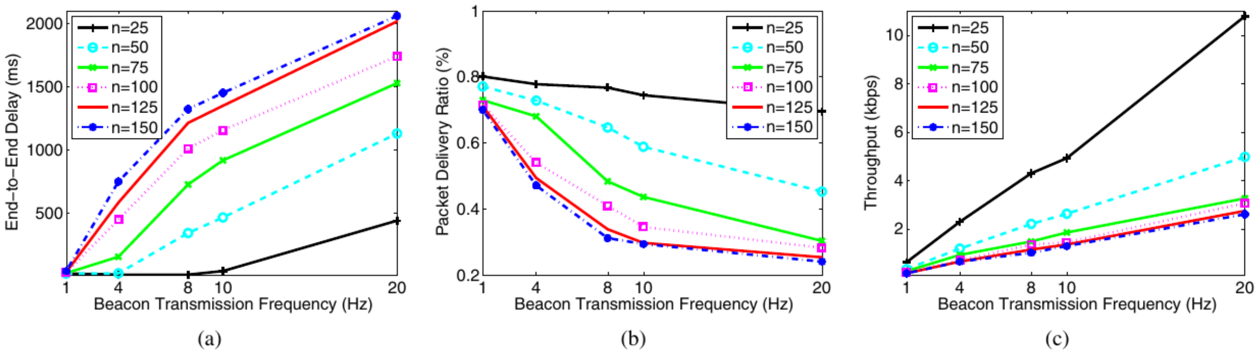


Figure 4: Simulated characteristics for IEEE 802.11p [12]

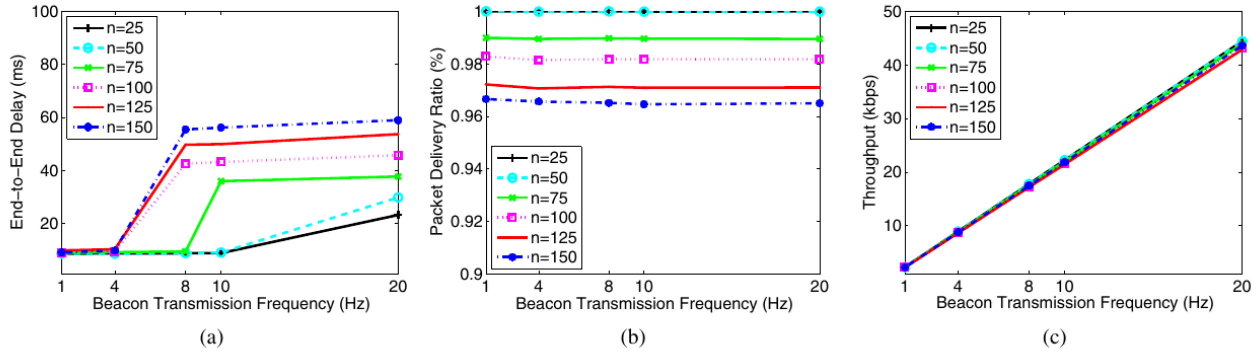


Figure 5: Simulated characteristics for LTE [12]

As we can see, LTE can reliably satisfy low End-to-End delay under 100 ms and also PDR is very close to 1 even for larger amount of vehicles involved in simulation. The main factor of good performance of LTE is due to assisted scheduling and access control which is not present in infrastructure-less IEEE 802.11p.

From results of these two papers we can conclude, that LTE communication technology performs a lot better in urban scenarios and should be considered as significant competitor to IEEE 802.11p in such environment.

2. 3. Domain of use of the ITS communication services

The main purpose of this network is driver assistance by enlarging the field of view of the driver. An information about the state (e.g. speed, velocity, road conditions...) of nearby vehicles would be obtained using CAM and LDM and be processed by ITS unit and used as driving assistance application to help driver with orientation in traffic. The safety warnings and traffic informations would be delivered via DNM either by store and forward mechanism using only cars with V2V units or broadcasting information from roadside units.

Those information are essential for safety applications to help preventing collisions in dense traffic (e. g. in big town center, exits on the highway, left hand turns at crossroad). The knowledge of current traffic situation ahead a planned route also could be provided by V2V and V2I using information from other participants of the traffic and infrastructure nodes as well. We will mention some examples of subjects of current research.

- **The left turn assistant - BMW**

The car manufacturer BMW introduced their left-turn assistant which should make left turns more safe. It incorporates fusion of information from sensors, camera and position sharing via V2V. The V2V range is approximately 250 m. Stream from camera is used to read road markers to detect left-turn lane and GPS is used to match the position with its intersection map. There are laser scanners at front of the vehicle which maps forward area and V2V communication searches for unseen vehicles [11].

- **Convoy driving - Grand Cooperative Driving Challenge (GCDC)**

This project took place in Netherland in 2011. There were 11 teams competing in several traffic scenarios. The longitudinal control of the convoy was demanded and maneuvers like joining, splitting of convoy. Convoy is assumed to be heterogeneous, consisting of cars and trucks. Car manufacturers like KHT and Scania were involved (Scoop project) also Volvo and many others. Next round of GCDC will be held in 2016.

2. 4. Simulation tools

Whereas real-life testing of convoy platooning would be problematic and expensive, the computer simulation can be used to produce experimental data. Simulators also allow us to easily adapt testing scenario as we need with large scalability and repeatability. Basically every research project related to V2V or CACC relies on simulation. In this section we introduce few simulators which could suit our needs.

SUMO – Simulator of Urban Mobility

SUMO is free and open-source traffic simulator written in C++ language. SUMO has been created and updated by Institute of Transportation Systems in Berlin since 2001. It is capable of simulating vehicles, public transport and even pedestrians explicitly. Maps can be imported in several formats from Open Street Map. Also provides tools such as visualization, route finding, emission calculation and traffic control. SUMO is number one choice for most projects dealing with V2V. Here is a list of few examples, where SUMO has been used.

- Evaluate the performance of traffic lights, including the evaluation of modern algorithms up to the evaluation of weekly timing plans.

- Vehicle route choice has been investigated, including the development of new methods, the evaluation of eco-aware routing based on pollutant emission, and investigations on network-wide influences of autonomous route choice.

- SUMO was used to provide traffic forecasts for authorities of the City of Cologne during the Pope's visit in 2005 and during the Soccer World Cup 2006 [15].

VEINS

Veins is free open-source framework which joints SUMO simulator for traffic simulation and OMNeT++ as network simulator. Both simulators are running in parallel, connected via TCP socket. It's complex V2V simulator which allows to incorporate fully detailed standardized communication models (e.g. IEEE 802.11p, LTE, IEEE 1609.4) including QoS channel access, shadowing of moving and stationary objects, noise and interference of signal. Here we provide short list of features included in VEINS

- Allows online re-configuration and re-routing of vehicles in reaction to network packets.

- Can employ validated, computationally inexpensive models of shadowing effects caused by buildings as well as by vehicles.

- Supplies data sources for a wide range of metrics, including travel time and emissions.

- Can simulate city block level simulations in real time on a single workstation [16].

VSimRTI

VsimRTI stands for "V2X Simulation Runtime Infrastructure" and it is universal framework which is capable to joint traffic simulator with network simulator. Unlike VEINS you can choose which simulator you want to use. There are a lots of network simulators you can involve in simulation of network (ns-3, OMNeT++, SWANS etc.). VsimRTI can simulate both ad hoc and cellular networks. Here we provide short list of features included in VsinRTI

- Realistic station positioning – provides information about position formated in WGS-84 GPS coordinates to simulate real navigation system.
- Supports ITS basic messages CAM and DNM defined by ETSI. Also provide LDM management and database services.
- Stations are able to detect events and perform maneuvers as reaction. Different capabilities of individual stations can be specified. Different types of vehicles, roadside units and intelligent traffic lights are supported.
- Variety of addressing schemes for ad-hoc such as unicast, broadcast, geocast as well as infrastructure-based cellular networks.

The disadvantage of VSimRTI is, that it is not open-source, but is licensed [17].

Alite and AgentDrive

Alite is a software toolkit helping with particular implementation steps during construction of multi-agent simulations and multi-agent systems in general. The goals of the toolkit are to provide highly modular, variable, and open set of functionalities defined by clear and simple API. Alite is developed by Agent Technology Center at Czech Technical University in Prague. The AgentDrive project, which uses Alite API, is a consolidated simulation framework for realistic vehicles simulation enabling testing of agent-based algorithms for route planning, navigation, cooperative driving, traffic optimization, and others vehicle coordination and cooperation methods [18]. Alite and AgentDrive are written in Java programming language.

There are only few simulators mentioned but those are the most used and their projects are still running. Many other projects had ended and there were no follow-up project to develop previous work.

Chapter 3

Convoy control and algorithm implementation

In the first part of this chapter we discuss string stability of convoy. In the second part we explore implementation specifics for our chosen simulator. In the third part we introduce our implementation of communication, message management and motion control.

3. 1. String stability

The manufacturers started to equip their cars with adaptive cruise control (ACC) nowadays. ACC gathers information from radar sensor to obtain distance from preceding vehicle and uses it to keep desired velocity with respect to a safe distance. The desired distance may be a function of vehicle velocity. This is usually called "constant time-headway spacing". Despite the fact, that ACC can improve safety and efficiency of the traffic it can't be used to control a convoy of vehicles. The main issue of ACC is lack of string stability improvement.

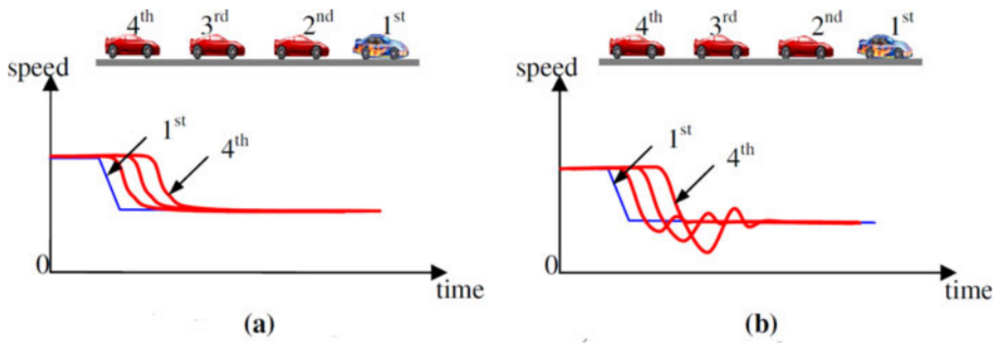


Figure 6: String stability (a) stable, (b) unstable [14]

A string of vehicles is said to be "string stable" when any non-zero position, speed and acceleration errors of an individual vehicle in the string do not amplify when they propagate upstream [14]. An example of string stability is shown in Figure 6. It is obvious that such unstable string of vehicles could cause traffic jam or even accident. A concept of an extension of the ACC called Cooperative ACC (CACC) was specified to provide desired string stability. CACC uses information about distance obtained by on-board radar, but also gathers information about surroundings using V2V communication. Especially information about velocity and acceleration of preceding vehicle can be used to improve string stability of convoy.

3. 2. Simulator specifics

We have chosen Alite simulator with AgentDrive framework for purposes of this thesis. The main reason for this choice was fact that it was developed within our university. There was an opportunity of direct contact with the research group working with it in case of having some troubles with implementation. In this section we explore simulator specifics, that must be taken into account while implementing algorithm and its limits.

3. 2. 1. Simulator coupling

AgentDrive is framework especially developed for vehicular simulation. It can be coupled with simulators such as “OpenDS” for realistic driving experience or, in our case, much simpler “Simulator-lite”, which visualizes traffic in 2D only. A reasons for this choice were technical support from colleagues at university, ability of changing source code and low CPU/GPU requirements. Also visualization of traffic is only for quick evaluation purposes. Detailed evaluation of performance is held in next chapter. Comparison of individual simulators visualization is shown in Figure 7.

Both AgentDrive and Simulator-lite (“simulator”) run in parallel. The simulator performs vehicle movement simulation based on physical model of vehicle. Simulator provides data about vehicles in periodical interval. AgentDrive goes through the list of agents present in simulation and assign fresh information from simulator to individual agents. Agents then perform their own logic to achieve desired behavior. Then it sends list of actions to be processed by the simulator. In our case action contains so called “waypoints” which represents points on the map to be driven through. Simulator obtains this information and performs steps defined by AgentDrive.

Figure 7: Traffic simulator OpenDS (left) and Simulator-lite (right)

3. 2. 2. Road description

AgentDrive uses map description taken from SUMO simulator. Basic representation of SUMO network is a directed graph. Nodes, usually named "junctions" in SUMO context, represent intersections, and "edges" roads or streets. Edges are unidirectional. Specifically, the SUMO network contains the following information:

- every street (edge) as a collection of lanes, including the position, shape and speed limit of every lane,
- traffic light logics referenced by junctions,
- junctions, including their right of way regulation,
- connections between lanes at junctions (nodes).

Files are generated by SUMO tools “NETCONVERT” or “NETGENERATE” and stored in .XML file format. Maps can be generated from various sources e. g. “Open Street Map” or can be defined by user as abstract road network [15]. The information about number of lanes and their direction is the most important part for us.

3. 2. 3. Basic AgentDrive tools

AgentDrive framework contains set of tools which are used to provide agent with information and ability to affect its state. In this section we bring the list of those tools because it will ease understanding of upcoming implementation description.

- RoadObject – describes vehicle with its basic set of data. It contains vehicle unique ID within the simulation. Other important item is “update time” which contains time when AgentDrive obtains the data from simulator. The most important data from RoadObject are position of the vehicle (3D point) and its velocity (3D vector). RoadObject also contains information about lane number, where vehicle is present.

- Sensor – it is universal object to detect vehicles and current situation on the road. It has ability to detect state of vehicle itself and return it in form of RoadObject. An information about every car can be obtained as set of RoadObjects and also state of specified vehicle can be sensed. Sensor is able to retrieve information about road and obstacles.

- Navigator – holds an information about route to be driven. It can obtain the whole route plan, or just part which wasn't driven through yet. Other important feature is changing lane to left and right.

- Action – contains information about steps to be performed by individual vehicle. Every action specifies vehicle velocity and “waypoint” to drive through.

- Actuator – actuator execute actions defined by agent logic.

3. 3. Controller specifics

Basic ACC controller relies only on distance obtained by radar sensor. Usually the desired distance d_D between vehicles is computed from time-headway t_H set by user, desired vehicle velocity v and stand-still distance d_{SS}, which represents distance between vehicles while not moving. All is summed up in equation

$$d_D = d_{SS} + v \cdot t_H \qquad (1)$$

When ACC have actual and desired distance between vehicles it can apply control action to “engine control unit” (ECU) which accelerates/decelerates. Hence we can describe control action of ACC as a general function u_{ACC} of distance $d_{i,i-1}$ between vehicles 'i' and 'i-1' as $u_{ACC} = f(x_i, d_{i,i-1})$, where x_i represents i^{th} vehicle state. However, common ACC cannot be used in convoy driving because of lack of string stability, as mentioned before. But when we extend ACC with V2V unit, we could gather more information about preceding vehicle than distance. We will examine a case, where we use distance and also difference of vehicles velocities as contribution to control action. The preceding vehicles velocity is obtained from communication only. It could be possible to estimate velocity by differentiating distance data from radar but measurement noise could corrupt the process causing unpredictable behavior. Considered control action would be a general function u_{CACC} of distance $d_{i,i-1}$ and relative velocity v_R between vehicles 'i' and 'i-1' as $u_{CACC}=f(x_i, v_R, d_{i,i-1})$, $v_R=v_i-v_{i-1}$ where x_i represents i^{th} vehicle state.

In our controller implementation we have to consider a difference between real (C)ACC and our simulation. In real life ACC a control action affects acceleration of the vehicle. In simulator we don't have direct access to acceleration. “RouteAgent” has its own internal regulator, which adjust acceleration to achieve prescribed velocity. As we can only affect velocity of the vehicle, so our control may act a little different. We will be modifying velocity by function u_{CACC} in form

$$u_{CACC}=f(x_i, v_R, d_{i,i-1})=-k_P \cdot d_{i,i-1}-k_V \cdot v_R \qquad (2)$$

Then velocity sent to our agent v_{SENT} will be

$$v_{SENT}=v_{SET}+u_{CACC} \qquad (3)$$

where v_{SET} denotes desired velocity of our agent. This situation brings some limitations which will be concluded in separate section.

3. 4. Implementation details

In this section we introduce implementation of communication system and control law at large. We also explain variety of options which our solution provides. We build our own agent by modifying existing agent called “Route agent”. Route agent is designed to follow road in defined velocity. Our agent is able to communicate with others so we called it “ComAgent”. The schematic design is shown on Figure 8 and its functionality is described below.

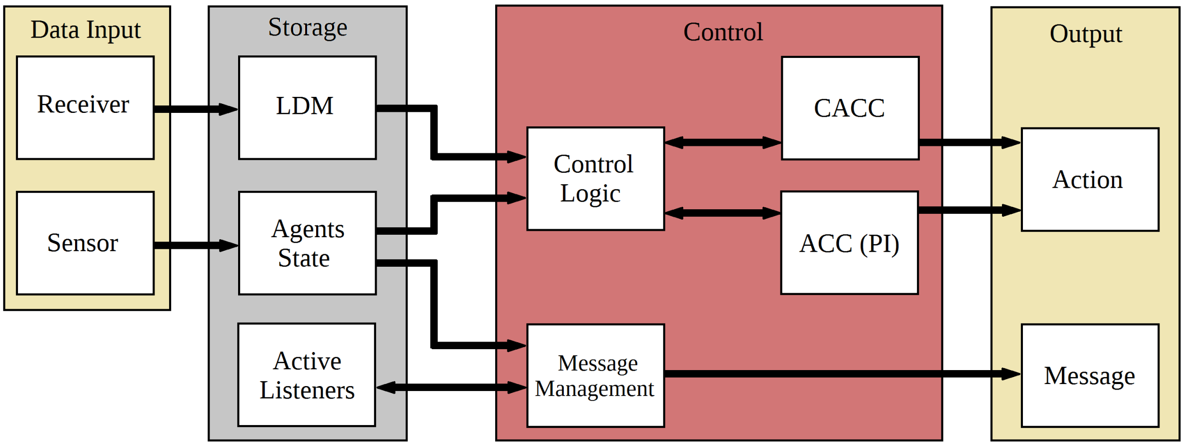


Figure 8: Schematic design of controller

3. 4. 1. Messages

We decided to make a message transmission as simple as possible. Messages are implemented in form of class called "ComMessage". Content of message is as follows

- Message type – indicates what type of message is. Basic type is CAM according to ETSI standard for ITS (see 2. 1. 2. Message management). All types of messages available must be defined in enumerate type we called "MesType". Message type is used for simplifying message processing by receiver.

- Sender ID – agent who sends message adds its ID within simulation. Used for identification during message processing.

- Receiver ID – identification number of agent for unicast messages. This ID is not used when message is broadcasted.

- Content – short string of characters for additional information.

- RoadObject – contains information about transmitting vehicle in form of RoadObject.

Transmission of messages is provided by events. Special event was created for this purpose. This event contains message and its source, which is in our case ComAgent. Agent is not only source of an event, but it must be also able to detect such event. We created listener interface "ComListener" which our agent has to implement to be able to listen and process an event with message.

3. 4. 2. CAM message management implementation

CAM message is broadcasted during every communication cycle between AgentDrive and simulator. Period of communication cycle is fixed to 200 ms. Event is fired by calling function “fireEvent” and message is transmitted. Latency of the channel is represented by stacking our events into list in form of FIFO queue which has limited length that matches to multiples of communication interval length. Minimal latency is one communication period long.

Message is received by function “receiveMessage”, which is overridden from interface “ComListener”. This function allows to obtain message content and it also can affect packet delivery ratio (PDR). PDR is implemented as random variable which is drawn from uniform distribution $U(0, 1)$. This value is then compared with fixed threshold of PDR and if the random number is lower than threshold, message is accepted and saved. Otherwise the message is ignored. Messages are stored in agents local storage which works as Local Dynamic Map. Messages in LDM are bounded to communicating agents in “Linked Hash Map” where agent serves as a key and message is respective value. When a new message is received, an old message is replaced by the new one so we have an access to fresh data only.

Every agent holds a list of agents which can obtain message from it. List of listeners is created during initialization of simulation. Individual agents can be marked as non-active. This option is used to limit the transmitter range. Agent performs this action automatically by calling function “manageMyListeners”, which marks all agents as non-active when they are farther than fixed distance. When non-active agent reaches transmitting distance it is added back to the list of active agents.

3. 4. 3. Zipper messages management

Our agents have embedded functionality for autonomous merging from two lanes to one. The first we describe situation from point of view of agent in ongoing lane. When driving in two separate lanes, vehicles keep their distance in lane, but ignores vehicles in other lane. After receiving request for joining from the other lane, agent switches into mode, where it keeps desired distance but according to vehicle in other lane if this is closer than vehicle in his lane. Agent also confirms that it received message with request and that it starts acting properly to make safe zipper maneuver possible. When there is enough space for agent from ended lane to switch lane, agent in ongoing lane sends message with permission to join ongoing lane. Agent also waits for confirmation about receiving message by other agent. If message is not confirmed in configured number of communication cycles it sends message again.

Now we describe situation from point of view of agent in ended lane. When needed, agent sends message with request for switching to other lane. Agent waits for confirmation about receiving message by other agent. If request message is not confirmed in configured number of communication cycles it sends message again. Distance keeping is performed in same way. When permission to join the ongoing lane is granted and agent has safe distance to preceding vehicle, it send confirmation and performs a maneuver. Permission of maneuver is necessary because agent needs to know whether agent in ongoing lane provides enough space for safe maneuver.

3. 4. 4. Velocity controller

We suppose that modern vehicle equipped with V2V unit also has on-board radar to measure distance between our vehicle and the previous one. In our simulation case, we use sensor to detect other vehicles on the road. Sensor returns information about vehicles in form of collection of RoadObjects. Since using all information from our sensor could be considered as cheating, we only search for the nearest vehicle in front of us. We can choose if we want information about the nearest car in general or within our lane only.

Obtaining the distance is the second operation made in every control cycle right after sensing of an agents own state. Then a set of waypoints is generated using navigator. This procedure is taken from original “RouteAgent”. We reduced the number of waypoints to one for simplicity and for reducing computation time. Afterwards agent searches through LDM to find a message, which corresponds with preceding agent. When the message is found, information about agents status is collected, especially its velocity. Special flag is set to show that message was successfully found.

Then the control action is computed. If we have fresh information about preceding vehicles velocity, we use control action u_{CACC} according to (3). Otherwise an alternative controller is used and agent acts like equipped only with ACC generating action u_{ACC}. Controllers for ACC may vary. In both cases the control action is stored as u_{OLD}. When control action is computed, the action is generated and added to action list, which is sent to simulator.

For our purposes we designed modified PI regulator as ACC controller which is designed as follows. If there is no fresh message from preceding agent, PI controller creates action processing information form radar and previous action performed u_{OLD} . Action can be summarized as

$$u_{ACC}=f\left(x_i, d_{i,i-1}, u_{OLD}\right)=-k_P \cdot d_{i,i-1}-k_I \cdot u_{OLD} \tag{4}$$

Final action sent to the simulator can be expressed as

$$v_{SENT}=v_{SET}+u_{ACC} \tag{5}$$

During all experiments, the controller is set as follows

$$k_P=0.8,\ \ k_I=0.05,\ \ k_V=1$$

3. 5. Controller limitations

In this section we summarize limitations of our controller. The greatest issue is localization of agents. Whereas the radar data are represented by set of RoadObjects we can derive distance to preceding car only by looking for a vehicle with the lowest distance. A similar problem is matching agents to received messages. There we also need agents exact position and also our exact position to to estimate preceding agents position and compare with position contained in message.

Another issue is agents velocity control. As we don't have access to affect vehicles acceleration directly, we have to send velocity set-points to agent. This kind of solution takes away possibility of absolute control of the vehicle and causes strange behavior which can be observed during experiments in Chapter 4.

Chapter 4

Performance evaluation

In this chapter we introduce scenarios, where performance of our controller will be tested. We also have to define testing methodology. The last we provide results of simulation tests and their evaluation.

4. 1. Test scenarios

There are three basic scenarios, which will help us examine our controller performance. We have chosen those which are the most usual events in daily traffic. List of scenarios with individual details is listed below.

- **Accelerating from zero velocity to given reference**

This situation is common crowded cities. Especially at traffic light we want highest possible amount of vehicles to pass through, before the red lite goes on. Ability to gain desired velocity in least time possible while keeping safe distance is main issue in this case.

- **Decelerating from common allowed velocity to the one close to zero**

Fast and safe decelerating is one of the most common issue at any type of road. Not keeping a safe distance is one of the most frequent cause of accidents. In this case we demand 100 % reliability of our controller for keeping safe distance to prevent collisions.

- **Zipper maneuver from two lanes into one**

Merging two lanes into one or simply switching between lanes is most common at highways. This maneuver must be made with caution, especially driver must watch blinds spots in mirrors. Another issue is that many drivers don't act properly during merging e. g. they don't make enough space for safe maneuver or even don't let the vehicle in ended lane to merge. While performing autonomous merging, communication between agents must be flawless.

4. 2. Testing methodology

In this section we introduce the metrics which we use to evaluate performance of our control. Section is divided into three subsections where the first one deals with common examined issues and the second one with those specific for individual scenarios. The last subsection describes variable parameters used in simulations and which influence will be examined.

4. 2. 1. Common performance metrics

- **Safe distance**

The first and one of the most important metric is safe distance. During all experiments we will carefully observe distance between individual agents. We have chosen stand-still distance d_{SS} to be equal to 7 meters. We must consider the fact, that vehicles position doesn't include length of the vehicle so real stand-still distance is approximately 2 meters (we suppose vehicle length around 5 meters). If the distance between vehicles is lower than d_{SS}, the control will be assumed as inadmissible. If the distance between vehicles is lower than 5 meters, we consider that accident occurred.

- **String stability**

The second phenomenon to be observed is string stability. As defined earlier, a string of vehicles is said to be "string stable" when any non-zero position, speed and acceleration errors of an individual vehicle in the string do not amplify when they propagate upstream.

- **Velocity overshoot**

In this case the vehicle velocity will be observed with special attention to amplitude of overshoots. We consider acceptable overshoots of 5 % from the reference velocity to accept control as safe.

- **Settling time**

The third metric we want to observe is how fast the vehicle reaches desired velocity after reference changes. We expect some linear dependency with order in convoy. Another way to analyze it is to compute mean square error to the reference vehicle velocity or variation of difference from reference velocity. We say that settling time is the moment when actual velocity reaches interval of ±2 % around reference velocity and stays within this interval.

4. 2. 2. Individual performance metrics

Zipper maneuver scenario will also contain maximal time which was needed for vehicles to perform maneuver and also driven distance between sending zipper request message and switching lanes. Those data will help us to determine minimal transmission range for V2V or V2I communication devices.

Another important parameter is communication range of V2V unit. Because we are using control for distance from preceding vehicle, this parameter might not affect quality of control much. But it could be important when some vehicle is transmitting message, that it broke and it acts as an obstacle in current lane. We will estimate minimal value of this parameter from zipper maneuver experiments.

4. 2. 3. Variable simulation parameters

In this section we summarize parameters which we can vary within the settings of simulator and were described above. The first variable parameter is packet delivery ratio (PDR) which represents possible loss of the message or its damage during traveling in channel. It is modeled as random variable drawn from uniform distribution $U(0, 1)$ which is then compared with fixed threshold of PDR.

The second variable parameter is penetration rate. It means percentage of agents in traffic which are capable to communicate with others and use benefits of CACC. Agents which are not capable of communication are using plain ACC.

The third variable parameter is latency of the channel. A standard value for latency is time between simulator data exchange and it is present in every simulated experiment. This value can be easily extended by random multiplication of this value.

4. 3. Simulations and evaluation

In this section we will demonstrate a set of experiments to examine our controllers performance under different conditions created by varying simulation parameters. In the first example we will show behavior of convoy controlled by ACC with PI control to illustrate string instability. Then others experiments will be made with our implemented CACC. All experiments are performed by our designed control system implemented into agent based AgentDrive framework. All experiments have the same time-headway equal to 1 s. All numerical computation of results are performed using our scripts in MATLAB software, which we specially designed for our experiments evaluation.

Used shortcuts for metrics

Minimal velocity (undershoot)	v_{MIN}	[%]
Maximal velocity (overshoot)	v_{MAX}	[%]
Velocity settling time	t_S	[s]
Maximal distance (overshoot)	d_{MAX}	[m]
Minimal distance (undershoot)	d_{MIN}	[m]
Minimal zip distance	d_{ZMIN}	[m]
Maximal zip length	l_{ZMAX}	[m]
Maximal zip maneuver duration	t_{ZMAX}	[s]
Maximal zip position	p_{ZMAX}	[m]
Position of the first car in ended lane when zipper starts	p_{ZSTART}	[m]

4. 3. 1. Experiment 1: ACC with PI controller

This experiment is demonstrating situation where convoy of vehicles has zero velocity at the beginning. Then reference velocity changes to 30 m/s and vehicles accelerate to achieve desired velocity. The simulation has following settings

- Reference velocity 30 m/s
- Reference distance 37 m
- Number of agents 30
- Simulation time 100 s
- Variable parameter none

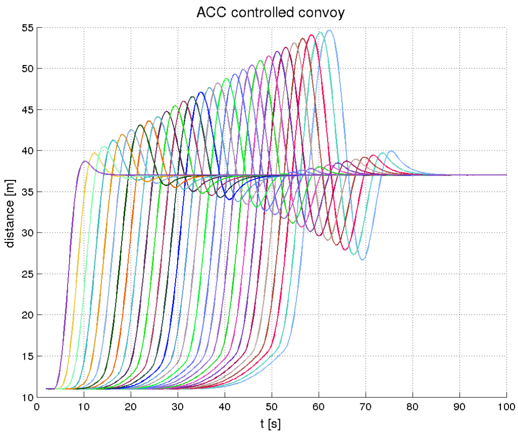


Figure 9: Exp. 1: Distance, PI only

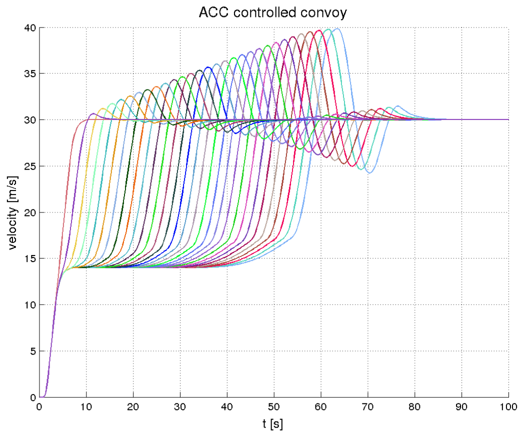


Figure 10: Exp. 1: Velocity, PI only

Table 1 below illustrates behavioral properties of chosen vehicles

vehicle number	1	5	10	20	30
v_{MAX} [%]	0	7.4	13.3	24.8	33
t_S [s]	8.4	20	33.6	55.2	79.2
d_{MAX} [m]	-	41	43.9	49.1	54.6
d_{MIN} [m]	-	36.6	35.41	32.6	27

Table 1: Experiment 1 results

As seen from simulation results, PI controller cannot provide string stability, because velocity and distance overshoots amplify when propagating through the convoy. There was no accident detected during experiment, but as we see growing trend of minimal distance, it might occur when there were more vehicles involved. When deceleration scenario with such PI regulator was tested[1], 7th vehicle of the convoy violated the safe distance and 10th vehicle crushed into previous one. We can conclude that such PI regulator is an example of control which cannot be used in convoy because of violating string stability resulting in dangerous behavior.

We can see on Figure 10 at the beginning of an experiment sudden acceleration performed by all vehicles at once. This behavior is caused by lack of direct acceleration control discussed in implementation part of Chapter 3. When velocity setpoint changes from zero to 30 m/s agents starts to accelerate in same manner. When they reach some velocity (in this case around 14 m/s) controller starts to push them back to handle proper distance. The same way is when decelerating. This behavior can be observed in most experiments.

1 Decelerating was performed from initial velocity 30 m/s to terminal velocity 5 m/s

4. 3. 2. Experiment 2: CACC in perfect conditions

This simulation aims to show a performance of our implementation of CACC controller in perfect conditions. Other experiments will be compared with this. PDR and technology penetration are set to 100 %. Communication channel is suppose to be perfect such that latency is only affected by length of the message.

4. 3. 2. 1. Accelerating from zero velocity

This experiment is demonstrating situation where convoy of vehicles has zero velocity at the beginning. Then reference velocity changes to 30 m/s and vehicles accelerate to achieve desired velocity. The simulation has following settings

- Reference velocity 30 m/s
- Reference distance 37 m
- Number of agents 30
- Simulation time 80 s
- Variable parameter none

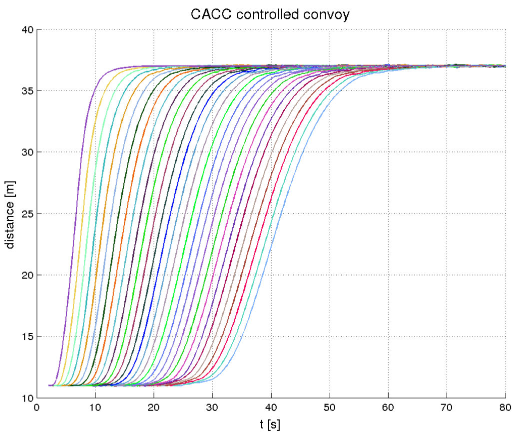


Figure 11: Exp. 2.1: Distance

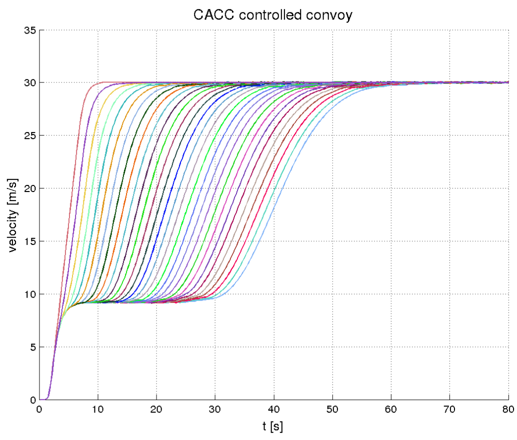


Figure 12: Exp. 2.1: Velocity

vehicle number	1	5	10	20	30
v_{MAX} [%]	0	0	0	0	0
t_S [s]	8.8	17.4	25.8	41.8	55.8
d_{MAX} [m]	-	37.05	37.09	37.09	37.03
d_{MIN} [m]	-	36.96	36.92	36.96	36.93

Table 2: Experiment 2.1 results

Table 2 shows results of simulation. Our CACC controller achieved perfect string stability with zero overshoot in velocity and distance from previous vehicle or with some insignificant perturbations. Also settling time is better for every vehicle, compared to PI control, thanks to suppression of oscillations.

4. 3. 2. 2. Decelerating to low velocity

This experiment is demonstrating situation where convoy of vehicles has stable velocity equal to 30 m/s at the beginning. Then reference velocity changes to 5 m/s and vehicles decelerate to achieve desired velocity. The simulation has following settings

- Reference velocity 5 m/s
- Reference distance 12 m
- Number of agents 30
- Simulation time 80 s
- Variable parameter none

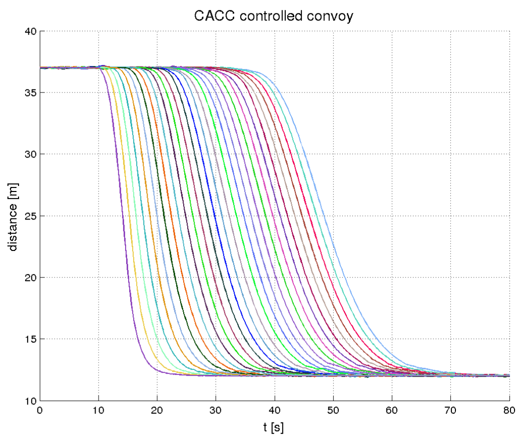


Figure 13: Exp. 2.2, Distance

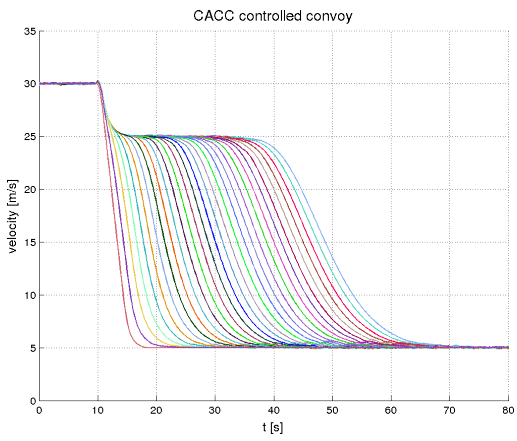


Figure 14: Exp. 2.2, Velocity

vehicle number	1	5	10	20	30
v_{MIN} [%]	0	1.9	2.9	1.8	2.2
t_S [s]	8	43.4	68.2	56	67.6
d_{MIN} [m]	-	11.98	11.98	11.93	11.99

Table 3: Experiment 2.2 results

From Table 3 can be seen that controller performed well. Desired distance between vehicles is achieved smoothly and is held without significant perturbations. Slightly higher perturbation in velocity can be observed and worse settling time, but it is due to much lower velocity set-point which causes all the relative errors to grow.

4. 3. 2. 3. Zipper maneuver

This experiment is demonstrating situation where there are two lanes occupied by vehicles driving with identical velocity. In defined time vehicles from right lane send request for zip maneuver. This experiment will help us to determine minimal time to perform maneuver and thus the minimal transmission range for V2V device. This value will be updated in further experiments. The simulation has following settings

- Reference velocity 30 m/s
- Reference distance 37 m
- Number of agents 30
- Simulation time 100 s
- Variable parameter none

v_{MAX} [%]	0.5
v_{MIN} [%]	50
t_S [s]	52.6
d_{ZMIN} [m]	21.8
l_{ZMAX} [m]	706
t_{ZMAX} [m]	40
p_{ZMAX} [m]	2169
$p_{Z,START}$ [m]	1980

Table 4: Experiment 2.3 results

The string stability was not violated during this experiment. Strange behavior presented one of agents which performed maneuver with distance to vehicle ahead only 21 meters while others did with minimal distance at least 30 meters. Maximal length of preparation phase to maneuver was 706 meters and took 40 seconds to complete. But if we look carefully at the last two rows of Table 4 we can see that space needed for all agents to zip to next lane was around 190 meters after the point where request has been sent.

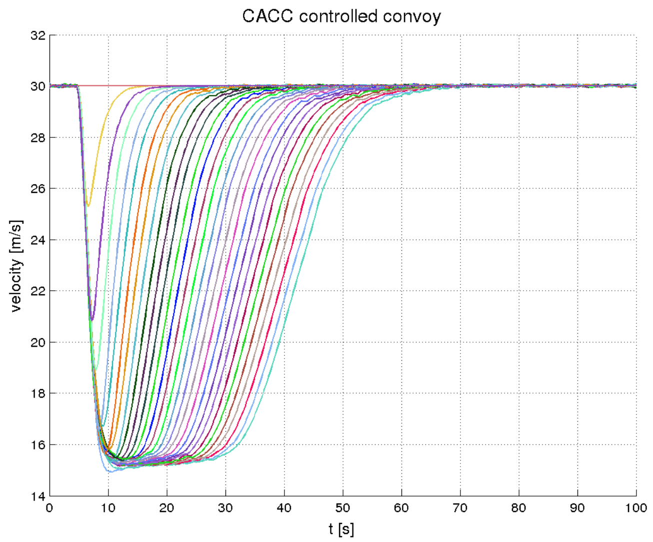


Figure 15: Exp. 2.3: Velocity

4. 3. 3. Experiment 3: CACC with varying PDR

This experiment aims to evaluate performance of our CACC controller under different values of packet delivery ratio. All three scenarios will be simulated and results will be compared to previous experiment with perfect conditions. At the end, we will determine minimal value of PDR which is suitable for safe operation of convoy control.

4. 3. 3. 1. Accelerating from zero velocity

This experiment is demonstrating situation where convoy of vehicles has zero velocity at the beginning. Then reference velocity changes to 30 m/s and vehicles accelerate to achieve desired velocity. In this experiment the performance is derived from the worst case in every metrics. The simulation has following settings

- Reference velocity 30 m/s
- Reference distance 37 m
- Number of agents 30
- Simulation time 80 s
- Variable parameter packet delivery ratio

PDR [%]	90	80	75	70	60	65	50
v_{MAX} [%]	0	0	0.4	1.8	2.6	3.3	7.3
t_S [s]	55	53.4	52.4	52.4	51.2	50.2	52.2
d_{MAX} [m]	37.08	37.07	37.06	38.3	38.3	38.9	40.8
d_{MIN} [m]	10.06	10	10.37	9.68	9.89	9.56	9.15

Table 5: Experiment 3.1 results

The biggest problem for our regulator was at the beginning where vehicles moved slowly. It had some perturbations in distance, but no accident occurred. When PDR has dropped under 65 % convoy started to have larger overshoots than our maximal value. String stability was not violated, however some disturbances occurred, they were dumped, not amplified when propagating upstream. The worst case with PDR equal to 50 % is shown in Figures below.

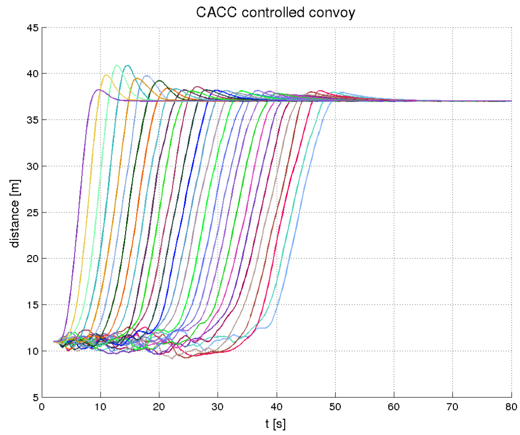


Figure 16: Exp. 3.1: Distance, PDR = 50 %

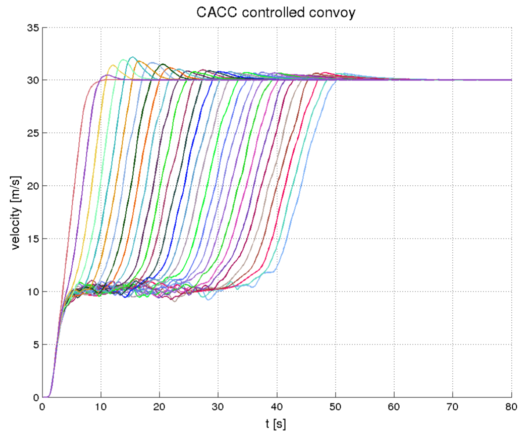


Figure 17: Exp. 3.1: Velocity, PDR = 50 %

4. 3. 3. 2. Decelerating to low velocity

This experiment is demonstrating situation where convoy of vehicles has stable velocity equal to 30 m/s at the beginning. Then reference velocity changes to 5 m/s and vehicles decelerate to achieve desired velocity. In this experiment the performance is derived from the worst case in every metrics. The simulation has following settings

- Reference velocity 5 m/s
- Reference distance 12 m
- Number of agents 30
- Simulation time 80 s
- Variable parameter packet delivery ratio

PDR [%]	90	80	75	70	60	50
v_{MIN} [%]	0	0	2.3	6.8	16.5	22.6
t_S [s]	70	69.6	69	67.6	66	59.2
d_{MIN} [m]	11.92	11.92	11.93	11.24	10.39	10.22

Table 6: Experiment 3.2 results

After dropping below 75 % of PDR control started to make velocity overshoots larger than our limit. But again, any disturbance in velocity and distance was successfully dumped while propagating upstream so string stability was kept. However bad disturbances at the beginning of the maneuver might appear as a place for improvement. No accident was detected during simulation. The worst case with PDR equal to 50 % is shown in Figures below

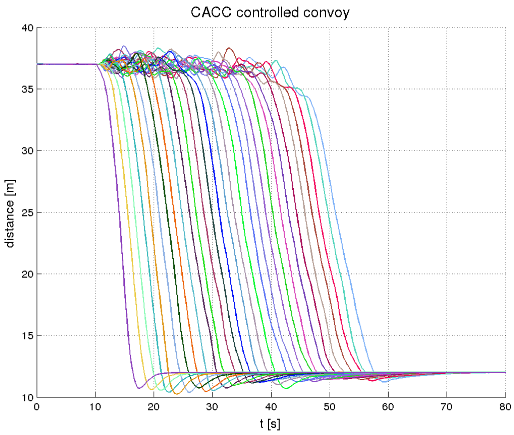


Figure 18: Exp. 3.2: Distance, PDR = 50 %

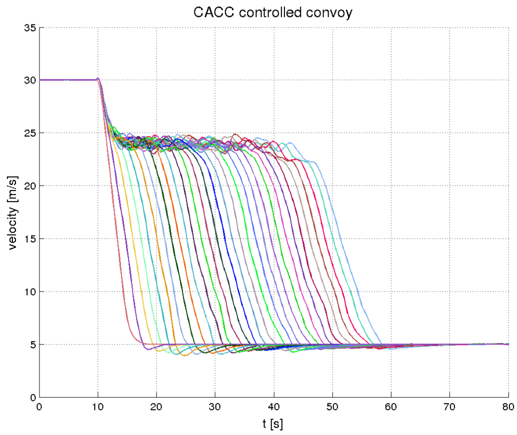


Figure 19: Exp. 3.2: Velocity, PDR = 50 %

4. 3. 3. 3. Zipper maneuver

This experiment is demonstrating situation where there are two lanes occupied by vehicles driving with identical velocity. In defined time vehicles from right lane send request for zip maneuver. This experiment will help us to determine minimal time to perform maneuver and thus the minimal transmission range for V2V device. This value will be updated in further experiments. The simulation has following settings

- Reference velocity 30 m/s
- Reference distance 37 m
- Number of agents 30
- Simulation time 100 s
- Variable parameter packet delivery ratio

PDR [%]	90	80	75
v_{MAX} [%]	0.6	5.8	33
v_{MIN} [%]	49.8	52.4	75
t_S [s]	52.6	49.4	71.2
d_{ZMIN} [m]	30.86	30.91	0.48
l_{ZMAX} [m]	749	721	2855
t_{ZMAX} [m]	41	38	103
p_{ZMAX} [m]	2183	2130	4599
$p_{Z,START}$ [m]	1988	1926	1929

Table 7: Experiment 3.3 results

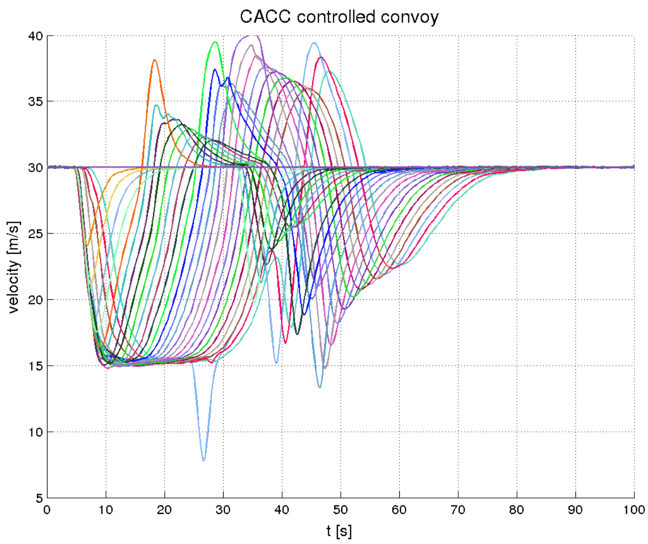


Figure 20: Exp. 3.3: Velocity, PDR = 75 %

Results show that this scenario is more prone to PDR than two previous. Overshoot limit was violated at PDR equal to 80 % and at 75 % of PDR some agents were not able to perform maneuver. All disturbances were dumped when propagating upstream, but agents behavior couldn't be called as decent. This might be caused by loss of confirmation messages which are sent only once.

4. 3. 3. 4. Conclusion of Experiment 3

Results of individual parts of experiment shows that for velocity control sufficient packet delivery ratio is 80 %. In case of zipper maneuver it is around 90 %. It might be decreased to 80 % as well, but it must had been preceded by algorithm improvement, especially part with special message management. If those improvements were made, algorithm could perform well even with transmission distance equal to 300 meters. Considering this distance, an algorithm must be extended with “store and froward” module to that every vehicle could obtain information about obstacle in shortest time possible. A problem with decreasing PDR could be solved with increasing message frequency, but it couldn't be verified because of fixed communication step of simulator. If we considered 500 m communication range we used and compared it with results in [9] (highway scenario) we could achieve 80 % PDR within 20 MHz bandwidth.

4. 3. 4. Experiment 4: CACC with varying technology penetration rate

This experiment aims to evaluate performance of our CACC controller under different values of technology penetration. All three scenarios will be simulated and results will be compared to experiment with perfect conditions. We suppose that penetration rate must be the highest possible, because if one agent isn't capable to communicate it will corrupt also agent behind.

4. 3. 4. 1. Accelerating from zero velocity

This experiment is demonstrating situation where convoy of vehicles has zero velocity at the beginning. Then reference velocity changes to 30 m/s and vehicles accelerate to achieve desired velocity. In this experiment the performance is derived from the worst case in every metrics. The simulation has following settings

- Reference velocity 30 m/s
- Reference distance 37 m
- Number of agents 30
- Simulation time 80 s
- Variable parameter technology penetration rate (TPR)

TPR [%]	95	90	85	80	75	70
v_{MAX} [%]	0.5	0.5	0.5	0.5	12.9	13.3
t_S [s]	54.8	54	53	53	52.6	50.4
d_{MAX} [m]	37.09	37.07	37.07	37.08	43.3	43.3
d_{MIN} [m]	9.18	9.25	9.18	9.2	9.09	9.19

Table 8: Experiment 4.1 results

In this experiment a behavior of the convoy got worse when penetration rate was lower than 80 %. Amplification of velocity can be observed from the fourth agent to the eighth agent, but effect is then dumped by vehicles with active CACC.

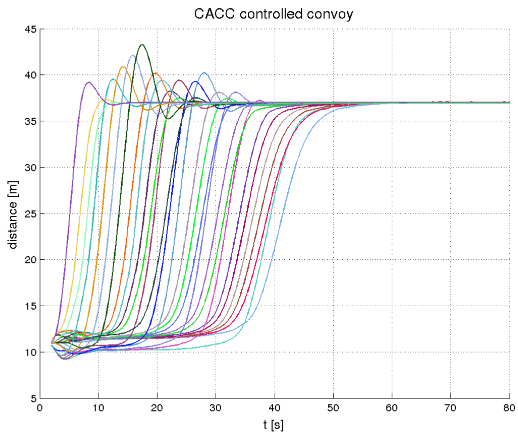


Figure 21: Exp. 4.1: Distance, TPR = 70 %

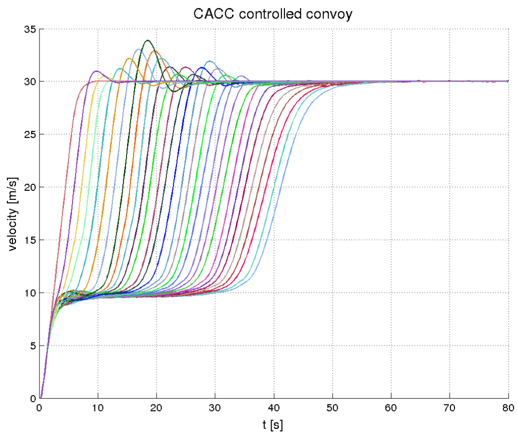


Figure 22: Exp. 4.1: Velocity, TPR = 70 %

4. 3. 4. 2. Decelerating to low velocity

This experiment is demonstrating situation where convoy of vehicles has stable velocity equal to 30 m/s at the beginning. Then reference velocity changes to 5 m/s and vehicles decelerate to achieve desired velocity. In this experiment the performance is derived from the worst case in every metrics. The simulation has following settings

- Reference velocity 5 m/s
- Reference distance 12 m
- Number of agents 30
- Simulation time 80 s
- Variable parameter technology penetration rate (TPR)

TPR [%]	95	90	85	80	75	70
v_{MIN} [%]	2.9	2.5	17.8	19.56	5.9	45
t_S [s]	70	69.8	69.8	68.6	69.4	69.6
d_{MIN} [m]	11.91	11.92	10.15	10.09	11.27	8.31

Table 9: Experiment 4.2 results

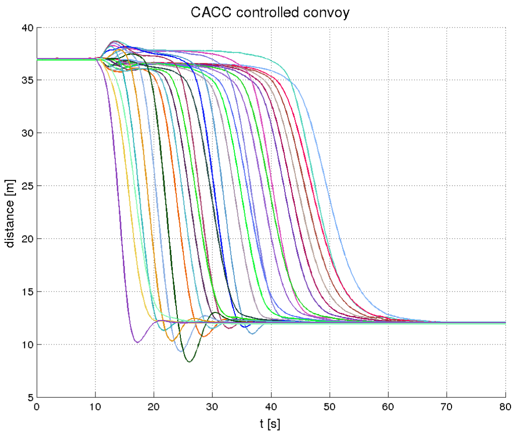


Figure 23: Exp. 4.2: Distance, TPR = 70 %

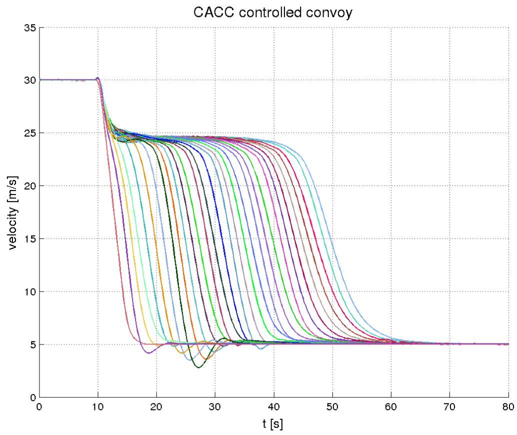


Figure 24: Exp. 4.2: Velocity, TPR = 70 %

String stability was kept even when dome amplifications have occurred, the were dumped while propagating upstream. Velocity overshoot was violated in this experiment when TPR dropped below 90 %. However performance seems to drop really fast, sometimes could happen, that lower penetration rate can have better results than a higher one. This can be seen from Table 9 where at 75 % of TPR we have better performance than in case with 85 % TPR. This is probably caused by a random selection of non-comunicating vehicles and a number of such vehicles in a row. Scenario with zipper maneuver was not included because of its dependency on 100 % communication ability.

4. 3. 4. 3. Conclusion of Experiment 4

In this experiment we discovered an interesting anomaly. It happened during decelerating part of experiment with TPR equal to 75 %. It shows, how tricky can be using function for random variable generation to simulate uniform distribution. Results also imply that penetration rate is more complex problem and is not only dependent of percentage of vehicles equipped with V2V device, but also their dispersion within individual lane. In our simulation we observed that four vehicles without communication ability can almost cause an accident (deceleration experiment with TPR = 70 %).

4. 3. 5. Experiment 5: CACC with varying channel latency

This experiment aims to evaluate performance of our CACC controller under different values of channel latency. All three scenarios will be simulated and results will be compared to experiment with perfect conditions. At the end, we will discuss minimal value of latency which is suitable for safe operation of convoy control.

4. 3. 5. 1. Accelerating from zero velocity

This experiment is demonstrating situation where convoy of vehicles has zero velocity at the beginning. Then reference velocity changes to 30 m/s and vehicles accelerate to achieve desired velocity. In this experiment the performance is derived from the worst case in every metrics. The simulation has following settings

- Reference velocity 30 m/s
- Reference distance 37 m
- Number of agents 30
- Simulation time 80 s
- Variable parameter latency

lat. [ms]	600	800	1000	1200	1400
v_{MAX} [%]	0.5	0.5	3.2	33	33
t_S [s]	53.8	50	38.6	59	69
d_{MAX} [m]	37.09	37.1	38.24	56	63
d_{MIN} [m]	10.79	10.79	10.8	10.91	10.9

Table 10: Experiment 5.1 results

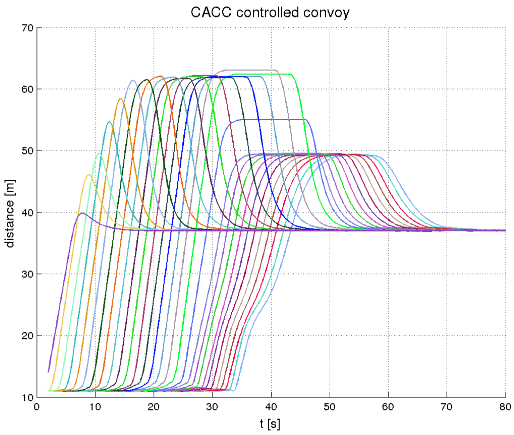


Figure 25: Exp. 5.1: Distance, lat. = 1400 ms

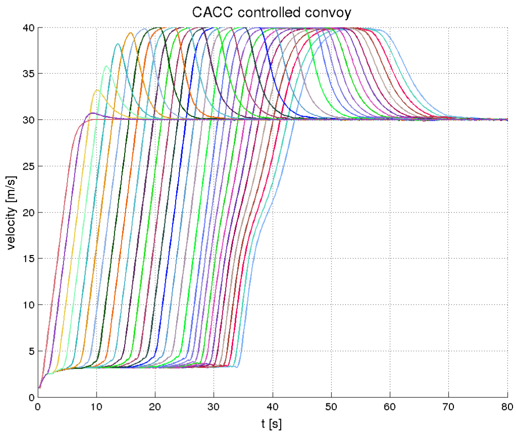


Figure 26: Exp. 5.1: Velocity, lat. = 1400 ms

Table 10 show results of experiment. String stability is violated when latency grows above 1000 ms. As we are limited with fixed communication period we cannot do denser scaling to observe this area more thoroughly. As we can see on Figure 26 the velocity reaches the upper limit of regulator which is 40 m/s.

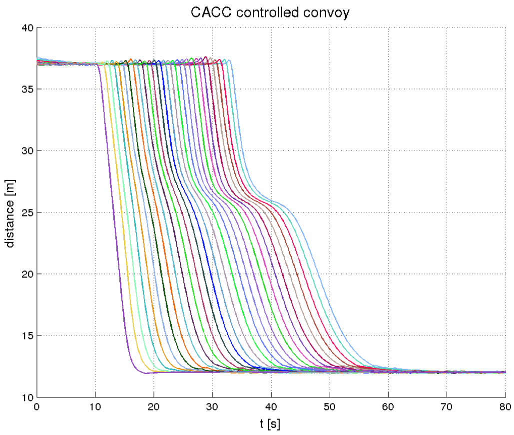
4. 3. 5. 2. Decelerating to low velocity

This experiment is demonstrating situation where convoy of vehicles has stable velocity equal to 30 m/s at the beginning. Then reference velocity changes to 5 m/s and vehicles decelerate to achieve desired velocity. In this experiment the performance is derived from the worst case in every metrics. The simulation has following settings

- Reference velocity 5 m/s
- Reference distance 12 m
- Number of agents 30
- Simulation time 80 s
- Variable parameter latency

lat. [ms]	600	800	1000	1200	1400
v_{MIN} [%]	3.2	3.4	2.9	2.6	2.9
t_S [s]	70	69.8	70	70	70
d_{MIN} [m]	11.92	11.9	11.92	11.94	11.9

Table 11: Experiment 5.2 results



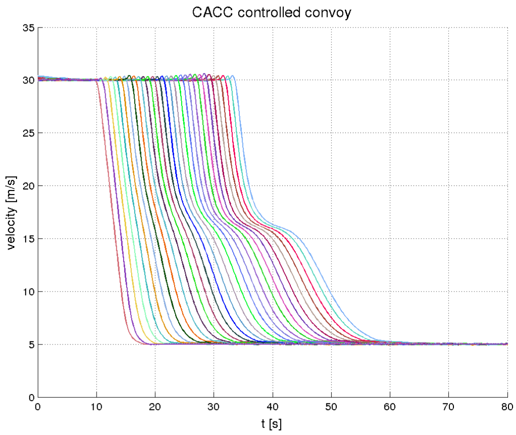


Figure 27: Exp. 5.2: Distance, lat. = 1400 ms   Figure 28: Exp. 5.2: Velocity, lat. = 1400 ms

Our controller performed much better during decelerating than when accelerating while in previous experiments otherwise. There were no significant overshoots in velocity. String stability was not violated even at highest latency examined.

4. 3. 5. 3. Zipper maneuver

This experiment is demonstrating situation where there are two lanes occupied by vehicles driving with identical velocity. In defined time vehicles from right lane send request for zip maneuver. This experiment will help us to determine minimal time to perform maneuver and thus the minimal transmission range for V2V device. This value will be updated in further experiments. The simulation has following settings

- Reference velocity 30 m/s
- Reference distance 37 m
- Number of agents 30
- Simulation time 100 s
- Variable parameter latency

lat. [ms]	600	800	1000	1200	1400	1600
v_{MAX} [%]	0.6	0.6	0.6	4.2	1.9	11
v_{MIN} [%]	50.7	49.8	76.4	50	50.8	51.7
t_S [s]	50	48.6	46.6	46.6	43.4	49
d_{ZMIN} [m]	30.8	30.9	31.2	33.4	32.2	32.2
l_{ZMAX} [m]	781	678	635	596	680	655
t_{ZMAX} [m]	40.6	39.6	40	33.2	39.2	38
p_{ZMAX} [m]	2170	2142	2146	2161	2188	2197
$p_{Z,START}$ [m]	1980	1981	1992	1968	1982	1979

Table 12: Experiment 5.3 results

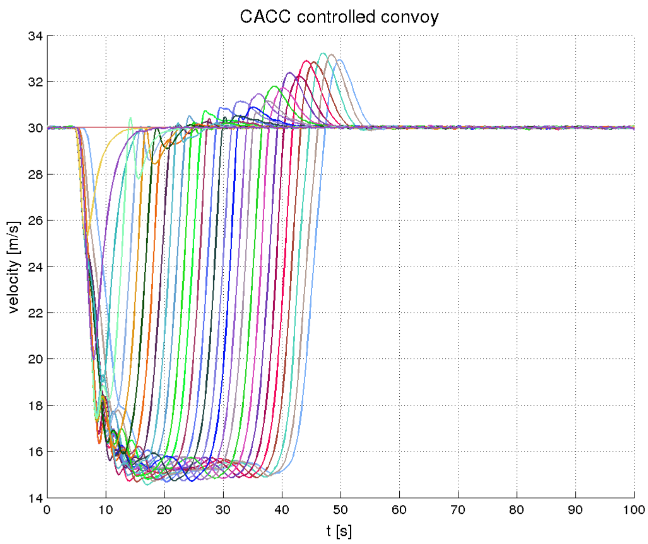


Figure 29: Exp. 5.3: Velocity, lat. = 1600 ms

During this experiment the string stability was violated at latency equal to 1400 ms. No accident occurred but some agents acted strangely. For example with latency 1000 ms one agent did not received enough space, so he had to let few vehicles go in front of him and joined to the end of convoy in ongoing lane.

4. 3. 5. 4. Conclusion of Experiment 5

An algorithm worked within our criterion range in all parts of experiment when latency was below 1000 ms. We have to consider a fact, that this experiment was performed with packet delivery ratio equal to 100 % so every message was delivered. In real situation with non-perfect PDR we should secure shorter latency e. g. 600 ms or less. Despite problems with some agents, group was still able to perform zipper maneuver below 200 meters from position of the first agent when sending request to switch lanes. Previously proposed minimal communication distance of 300 m seems reasonable (still considering message forwarding between agents).

4. 3. 6. Experiment 6: CACC with proposed parameters

In this experiment we have set parameters of simulation as we proposed in previous experiments. PDR was set to 80 %, TPR was set to 90 % and latency to 1000 ms.

4. 3. 6. 1. Sine wave

This experiment aims to prove that our control will perform well with previously individually tested parameters when all involved. We also used added sin wave to a reference for vehicles velocity. Simulation has following settings

- Reference velocity 20 m/s + *sin()* with 5 m/s amplitude, 31.4 s period
- Reference distance 27 m
- Number of agents 30
- Simulation time 120 s

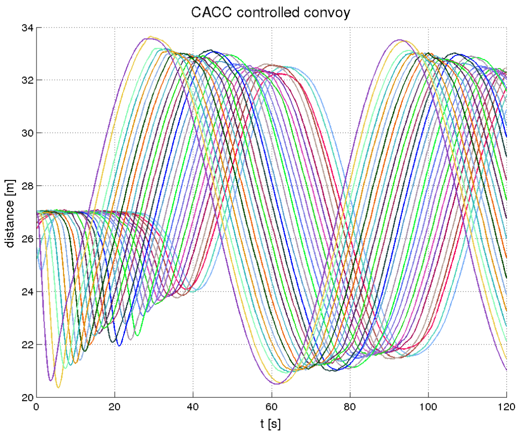


Figure 30: Exp. 6.1: Distance

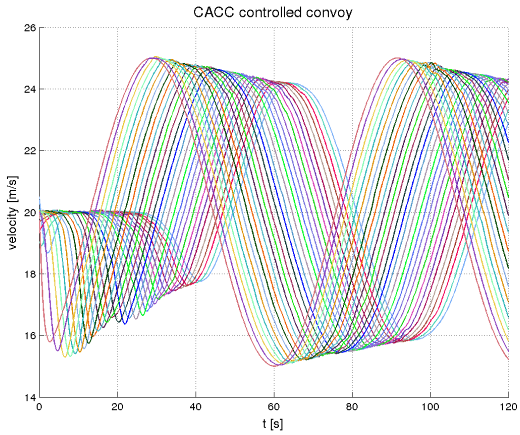


Figure 31: Exp. 6.1: Velocity

As we can see at Figures above, the distance is amplified a little bit when vehicle is not capable to communicate. Velocity seems to vary only a little in this experiment.

4. 3. 6. 2. Acceleration and deceleration

This experiment aims to prove that our control will perform well with previously individually tested parameters when all involved. We used acceleration to 20 m/s and deceleration to 5 m/s at once. Simulation has following settings

• Reference velocity	20 m/s	5 m/s
• Reference distance	27 m	12 m
• Number of agents	30	
• Simulation time	140 s	

Results are illustrated at Figures below. We can see that string stability was violated because of two agents in a row driving only in ACC mode. No accident occurred. We can conclude this experiment such that we should toughen the minimal values of the parameters or incorporate better ACC which doesn't amplify previous car disturbances that much. It also shows, that technology penetration rate can cause bigger problems than other variables.

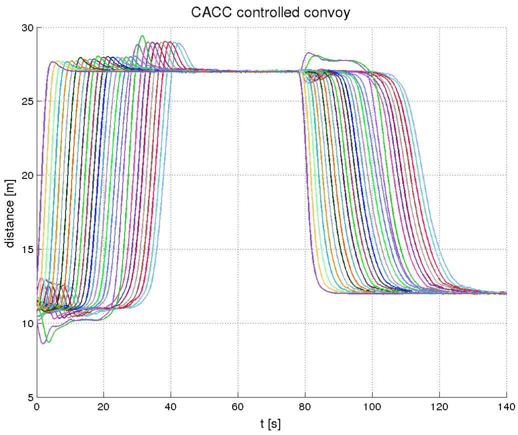


Figure 32: Exp. 6.2: Distance

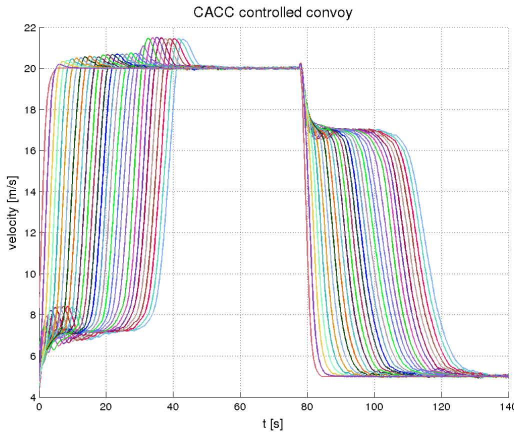


Figure 33: Exp. 6.2: Velocity

Chapter 5

Conclusion

In this chapter we summarize the content of thesis. We conclude the results of our experiments and provide proposals for future work.

In Chapter 2 we explored current wireless technologies which might be used for V2V communication. We broadly discussed European standard for vehicular ad hoc network based on IEEE 802.11p. We have shown its structure, properties and introduced its main components for message management, such as “Cooperative Awareness Message”, “Decentralized environmental Notification Message” and “Local Dynamic Map”. One section was dedicated to efficiency and reliability aspects of this standard and also its comparison with another common communication standard LTE. Some current research projects were mentioned in this chapter.

We explored simulation tools which might be suitable for our thesis. They are listed in Chapter 2 with focus on current projects, where outdated projects were omitted. We have chosen “AgentDrive” framework with “Simulator-Lite“ tool which were both designed within Czech Technical University in Prague. The main reason was possibility of immediate support from colleagues with AgentDrive while other simulators had lack of documentation.

We designed a controller for autonomous velocity control based on “Cooperative Adaptive Cruise Control” (CACC) with special emphasis to string stability keeping. We also implemented message management according to European standard (with CAM and DNM messages and LDM mentioned in Chapter 2). Special functionality for autonomous zipper maneuver was also implemented including its own messages and their organization. Implementation was build such that it enables to change various parameters to affect communication process. All implementation details can be found in Chapter 3.

In Chapter 4 our controller was tested in three scenarios which contain acceleration maneuver, deceleration maneuver and zipper maneuver. We examine behavior of convoy with 30 vehicles because such quantity shows observed properties well. The first experiment was performed with adaptive cruise controller with PI control to show problems with string instability of such solution. Implementation of such controller was not the main purpose of this work, it served as illustration of string instable behavior. The second experiment shows full potential of our CACC control and is held within perfect communication conditions. The third experiment evaluates performance of our controller in different values of "Packet Delivery Ratio" (PDR). Our conclusion about minimal PDR for proper functionality is equal to 80 % if some improvements were made in zipper control. We also compared our results with other work and concluded that channel bandwidth dor such PDR should be 20 MHz which is twice larger than originally considered by ETSI. The fourth experiment shows dependency on "Technology Penetration Rate" (TPR) which represents percentage of cars in convoy equipped with V2V communication technology. During this experiment we discovered that performance of our controller is not conclusively dependent on TPR itself, but also on how many consecutive vehicles in a row were not capable of communication. We suggested minimal TPR to be at least 90 %. The fifth experiment has dealt with performance at different levels of communication channel latency. We could have affected latency only with step of 200 ms which was caused by fixed communication period of "AgentDrive". We discovered that our CACC was the most vulnerable to growing latency during acceleration experiment. As result we stated that latency should be lower than 1000 ms to operate properly. We also tested those settings in the sixth experiment. It shows that latency and PDR would still keep string stability when there were 100 % penetration rate. This marks TPR as main issue when driving in convoy. If we want to drive large convoy safely, all vehicles must be equipped with CACC controller to handle worse channel conditions. During our experiments a minimal communication distance of transmitter was determined to be at least 300 m with proper forwarding of DNM messages.

For future work we would suggest unification of our version of LDM, which contains only CAM messages, with other used messages. Especially messages for zipper maneuver control and their management might require more attention to improve reliability of autonomous zipper control. Other part which could be improved is velocity control, but it would require deeper analysis of agent motion logic based on "RouteAgent" or maybe make whole new one designed specially for our purpose. The last but not least some state estimator could be included into algorithm which could be used to improve behavior of CACC in lower PDR or even to improve ACC controller to reduce velocity overshoots.

Bibliography

[1] Olfati-Saber, R.; Fax, J.A.; Murray, R.M., Consensus and Cooperation in Networked Multi-Agent Systems, Proceedings of the IEEE , vol.95, no.1, pp.215,233, Jan. 2007

[2] Russell, Stuart, Peter Norvig, and Artificial Intelligence. A modern approach. Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs 25 (1995)

[3] Stanica, Razvan; Chaput, Emmanuel; Beylot, André-Luc: Simulation of vehicular ad-hoc networks: Challenges, review of tools and recommendations

[4] Christian Weiß: V2X communication in Europe – From research projects towards standardization and field testing of vehicle communication technology. Elsevier B.V. 2011

[5] Gérard Ségarra, Road Co-operative Systems – Societal and Business Values, IEEE

[6] Iulia Ivan, Philippe Besnier, Matthieu Crussière, M'hamed Drissi, Lois Le Danvic, Mickael Huard, and Eric Lardjane; Physical Layer Performance Analysis of V2V Communications in High Velocity Context, IEEE

[7] Böhm Annette, Jonsson Magnus; Position-Based Data Traffic Prioritization in Safety-Critical, Real-Time Vehicle-to-Infrastructure Communication, IEEE

[8] ETSI Standard for ITS, link

[9] Lei Shi, Ki Won Sung; Spectrum Requirement for Vehicle-to-Vehicle Communication for Traffic Safety, IEEE, 2014

[10] Zeeshan Hameed Mir, Fethi Filali; On the Performance Comparison between IEEE 802.11p and LTE-based Vehicular Networks, IEEE, 2014

[11] Steven Ashley, BMW demonstrates left-turn driver's aid, Automotive engineering magazine, SAE 2011, available online here

[12] Zeeshan Hameed Mir and Fethi Filali; On the Performance Comparison between IEEE 802.11p and LTE-based Vehicular Networks, IEEE, 2014

[13] Andreas Möller, Jörg Nuckelt, Dennis M. Rose, Thomas Kürner; Physical Layer Performance Comparison of LTE and IEEE 802.11p for Vehicular Communication in an Urban NLOS Scenario, IEEE, 2014

[14] Chenxi Lei, Emiel Martijn van Eenennaam, Wouter Klein Wolterink, Jeroen Ploeg, Georgios Karagiannis and Geert Heijenk; Evaluation of CACC string stability using SUMO, Simulink, and OMNeT++; EURASIP Journal on Wireless Communications and Networking 2012, 2012:116

[15] SUMO – Simulation of Urban MObility; Official website of Institute of Transportation Technology in Berlin, link

[16] VEINS – Vehicles in Network Simulation; Official website link.

[17] VsimRTI, Daimler Center for Automotive Information Technology Innovations; Official website link.

[18] Alite, AgentDrive; Agent technology center, CTU; Official website link.

Appendix 1

Contents of attached CD

svandmi2_thesis.pdf	This thesis in .pdf file format
figures	Figures used in this thesis in separate files
sources	Source codes with implementation of algorithm