Sorbonne University

Doctoral School of Informatics, Telecommunications and Electronics of Paris

EURECOM

Autonomous cars' coordination among legacy vehicles applied to safe braking

Presented by Raj Haresh PATEL

Dissertation for Doctor of Philosophy in Information and Communication Engineering

Directed by Jérôme Härri and Christian Bonnet

Publicly presented and defended on 3 December 2018

A Jury committee composed of:

Prof. Philippe Martinet	École Centrale de Nantes	President
Prof. Arnaud de la Fortelle	MINES ParisTech	Reviewer
Prof. Cristina Olaverri Monreal	Johannes Kepler Universität Linz	Reviewer
Prof. Henk Wymeersch	Chalmers University	Examiner
Prof. Prométhée Spathis	Sorbonne University	Examiner
Prof. Jérôme Härri	EURECOM	Examiner
Prof. Christian Bonnet	EURECOM	Examiner

Abstract

An autonomous vehicle operation can be impacted by various internal factors like onboard system failure, sensor failure, etc. or by external factors like risky maneuvers by immediate neighbors threatening a collision, sudden change in road conditions, etc. In such situations when conditions dynamically change and the nominal operational condition is violated by internal or external influences, an autonomous vehicle must have the capability to reach the minimal risk condition. Bringing the vehicle to a halt is one of the ways to achieve this.

At early market penetration, automated vehicles will share the road with legacy vehicles. When the road is occupied by only autonomous vehicles, coordinating autonomous vehicles is easy as each of them can be controlled. On the other hand, the presence of legacy manually driven vehicles among autonomous vehicles complicates coordinated maneuvers. For a safe transportation system, autonomous vehicle controllers, therefore, need to estimate the control behavior of the legacy vehicles and use the estimate to generate controls. However, mismatches between the estimated and real human behaviors can lead to inefficient control inputs and even collisions in the worst case. This coupled with uncertainties and errors in different modules on the autonomous vehicle like the perception and localization module, the communication module, and the control module makes coordination issue challenging.

This thesis proposes a safe stop algorithm which generates controls for autonomous vehicles considering the presence of other legacy vehicles on the road. A Model Predictive Control based algorithm is proposed which is robust to communication, localization, control errors and model mismatch. Control computations for autonomous vehicles take place on a centralized controller whereas legacy vehicles react to the action of the vehicle in the front.

The performance of the robust controller is evaluated under different errors implemented using different models and compared to the performance of the non-robust controller. Collisions avoided and discomfort faced by the driver are two evaluation parameters. Simulations show that the performance of the robust controller under the presence of communication and localization errors is similar to the performance of the non-robust controller in the absence of those errors.

Driver in the loop experiments were also performed to evaluate the performance of the proposed controller in the presence of human drivers. A driving simulator was interfaced with a Matlab based autonomous vehicle controller. Controls implemented by a human driver were extracted using a driving simulator and transmitted to the controller. While theoretical simulations did not result in any collisions, the experiment using a driving simulator resulted in collisions and a higher value of

discomfort. These issues support the need to validate the developed algorithms by experiments and not just by simulations.

Acknowledgements

This three-year journey has been an unforgettable experience for me. Going through ups and downs, with challenges every other day, I faced them with growing motivation and positivity. I am happy I had the opportunity to live this experience, and I would like to thank EDITE for the scholarship they offered.

I am indebted to my supervisors, Jerome Haerri and Christian Bonnet at EURECOM without whom I wouldn't have been able to be as competent as I am today to tackle issues, not just technical but also managerial. Thanks for their continuous guidance and support I have been able to complete the thesis. I would also like to thank the members of the CAVS team at EURECOM for technical discussions. I would like to thank EURECOM and its members for funding my trips to conferences where I met many and discussed with fellow researchers working in this domain and made friends around the world.

My special thanks go to Henk Wyemeersch who provided me the opportunity to spend a few months at Chalmers and learn from his team there. His consistency and availability played an essential role in me being able to complete this thesis. His ideas and to the point discussions were of critical importance. I have learnt from him to be available for my colleagues, friends, and family when they need me.

It is my great pleasure to take this opportunity to thank all the people who helped me and accompanied me during my Ph.D. study. Good friends from EURECOM, the coffee group, the beach volley gang, and people who have been there by my side when I needed them, thanks to all of them.

Finally, I wish to thank my family for regularly checking on me and being by my side. I would like to thank my mother who usually used to be critical of my performance, but the day I started my Ph.D. was always supportive and encouraging; my father who always had faith and my brother who always supported me.

Contents

List of Figures

List of Tables

Acronyms

ACC .... Automated Cruise Control

ADAS .... Advanced Driving Assistance Systems

ADS .... Automated Driving System

AWGN .... Additive White Gaussian Noise

BSM .... Basic Safety Message

CA .... collision avoidance

CACC .... Cooperative Adaptive Cruise Control

CAM .... Cooperative Awareness Message

CAV .... Connected Automated Vehicle

C-ITS .... Cooperative Intelligent Transportation Systems

C-V2X Cellular-V2X

DCF .... Distributed Coordination Function

DDT .... Dynamic Driving Task

DENM .... Decentralized Environmental Notification Message

DGPS .... Differential GPS

DSRC .... Dedicated short-range communications

EDCA .... Enhanced Distributed Channel Access

EGNOS ...... European Geostationary Navigation Overlay Service

EU .... European Union

GNSS .... Global Navigation Satellite System

GPS .... Global Positioning System

HDM .... Human driving model

IDM .... Intelligent Driver Model

IDM+ .... Intelligent Driver Model+

IID .... independent and identically distributed

IMU .... Inertial Measurement Unit

MaaS .... Mobility-as-a-Service

MAC .... Medium Access

MDV .... Manually Driven Vehicle

MPC .... Model Predictive Control

NHSTA ...... National Highway Traffic Safety Administration

OBU .... Onboard Unit

ODD .... Operational Design Domain

PC .... Probable Collision

PHY .... Physical layer

PLR .... Packet Loss Ratio

RMPC .... Robust MPC

PRT .... Perception Response Time

RSSI .... Received Signal Strength Indicator

RSU .... Road Side Unit

RTK .... Real-time kinematic

SAE .... Society of Automotive Engineers

Satre .... Safe Road Trains for the Environment

SINR .... signal to noise plus interference ratio

TaaS .... Transportation-as-a-Service

VANET ...... Vehicular Ad hoc Network

VLC .... Visible Light Communications

V2I .... Vehicle to Infrastructure

V2V .... Vehicle to Vehicle

V2X .... Vehicle to Everything

Chapter 1

Introduction

1.1 General Introduction

Autonomous vehicles were introduced because they are supposed to contribute to reducing problems faced today like collisions, travel time, pollution, etc. Every day lots of people travel from their home to office, school, gym, parks, etc. Some travel from one corner of the world to another, may it be from the house to the office or moving between offices situated in different countries across the globe. Today, the world is fast-paced, literally and figuratively. Recent innovations from hoverboards to supersonic planes enable this movement. Growing population has also increased demand for transportation. Higher per capita income usually means better lifestyle which can be related to a need for a better, more comfortable and safer travel. The increase in travel demand has encouraged new business models like Transportation-as-a-Service (TaaS) and Mobility-as-a-Service (MaaS) which focuses on private or enterprise owned vehicle sharing. Ride-sharing applications like Uber and country specific applications like Chauffeur Privé (in France), Ola (in India) and DiDi (in China) are a few examples of them. In the air market, low cost flights primarily introduced a decade ago are constantly attracting the majority of the customers.

Need for safe and sustainable transportation

Increase in transportation demand has also directly or indirectly resulted in increased congestion and saturation not only on the roads but also in the air and rail sectors. According to a study in Britain, the average travel time of a person is more than one and a half hours daily [1]. Entrepreneurs have grasped the opportunity of providing services like online gaming, video streaming, etc. to occupy passengers during long travels and make money from these services [2]. If road traffic is considered, commuters in Los Angeles, CA spend 102 hours yearly on an average in traffic [3] which costs the city of Los Angeles more than $19 billion USD annually [4].

Increase in travel time and congestion is related to an increase in emissions from motor vehicles (vehicles with engines which consume energy). Fossil fuels contribute to global warming due to the creation of carbon dioxide. Observations from the United States Environmental Protection Agency point that motor vehicles contribute a sizable percentage towards pollution of earth's atmosphere, especially, over 55% of the total NOx pollution [5]. Other poisonous gases like Methane (CH4), Nitrous oxide (N2O), and hydrofluorocarbons (HFCs) not only impact the climate but also human and animal health alike. Air pollution causes cardio-vascular diseases [6] whereas noise pollution arising from excessive honking is related to hypertension and stress [7]. Bigger the transportation demand, more the trips, more the pollution. Thus cleaner and healthier transportation systems are required.

Congestion on the roads also leads to people taking riskier maneuvers which invites collisions. Human factors like fatigue, prolonged reaction time and human errors are significant reasons behind road accidents. More than 90 % of the road accidents occur due to human errors [8]. Although motor vehicle deaths decreased by 1% in 2017 in the US, more than 40,000 people lost their lives and the estimated cost of motor-vehicle deaths, injuries, and property damage in 2017 was $ 413.8 billion [9]. Thus, a safer transport system is of critical importance.

The need of the hour is to have a transportation system which ensures reduced travel time, reduced pollution, a decrease in the number of collisions, etc. Different measures have been taken to improve safety, for example, the introduction of seatbelts and airbags in the 1950s; lane departure warning, forward collision warning around 2000s; Advanced Driver Assistance Systems (ADAS) which include Automatic Emergency Braking, Lane Centering Assist, Rearview Video Systems etc. gradually from 2010 to 2016; adaptive cruise control, self park, etc. after 2016 highlight some of the use of partial-automation and technical advancements to improve safety. These innovations solve some of the above mentioned issues [10].

From awareness to autonomy

As human factors like fatigue, prolonged reaction time are a major reason for collisions, the initial goal is to use technology to perceive and make the driver aware of the situation earlier. Day one applications intend to increase the awareness horizon for the driver by detecting risky situations like different types of hazards early, and the driver is responsible for taking appropriate action. Some of the Day 1 services, which need to be available sooner than others include hazardous location notification, emergency braking notification, etc. [11]. Such Day 1 applications should be able to reduce collisions. Day 1 applications still require human action. In order to reduce the dependency on humans, vehicle automation can take over driving tasks as well. Day 2 applications are applications which use technology not just for perception, but also for automated driving. Cooperative Adaptive Cruise Control (CACC) and platooning are examples of such Day 2 applications involving cooperation between multiple autonomous vehicles.

Stand-alone driver assistance systems can help drivers in routine and critical situations and thus have positive effects on safety and traffic management. However, if individual vehicles were able to continuously communicate with each other or with the road infrastructure benefits could be further magnified. When vehicles are connected, different information can be shared with the road users and with the road infrastructure. Vehicle to Vehicle (V2V) and Vehicle to Infrastructure communications (V2I), together known as Vehicle to Everything communications (V2X) play a big part in Cooperative Intelligent Transportation Systems (C-ITS). Different entities in the system can cooperate to achieve common goals. The communication between vehicles creates a vehicular ad hoc network (VANET). If vehicles act independently, they can optimize individual goals; whereas if vehicles coordinate, they can optimize goals globally.

For vehicles to be a part of the C-ITS system, specific information will be communicated between the ego vehicle and other entities like neighboring vehicles or roadside units. Vehicles will transmit state information like position, velocity, etc. and receive information like safety warnings and traffic information. This communication usually takes place over the air as maintaining a wired connection with the vehicles would be difficult. Researchers in the US and European Union (EU) have been working on protocols to support low latency communication for high speed nodes, to fulfill the needs of C-ITS systems [12].

The predominant access technology for vehicular communication is based on the IEEE 802.11 standard, called Dedicated Short Range Communications (DSRC) in the USA, which was previously known as the IEEE 802.11p. The European variant of the access technology is called ITS-G5 (G5 refers to the 5 GHz frequency band) and is derived from DSRC with modifications to adapt to European requirements. Both in the USA and EU, it shall operate in the 5.9 GHz unlicensed frequency band. Cellular-V2X (C-V2X) is another technology for vehicular communication.

More than one-fifth of accidents occur with the vehicle immediately behind or ahead in longitudinal direction [13], and approximately 40 percent of all accidents take place at intersections [14]. In France, the number of fatalities per billion vehicle-kilometers has reduced since 1990 by 73% and stands at 7.0 deaths per billion vehicle-kilometers in 2011 [15]. In developing countries, the road infrastructure is unable to keep pace with the sharp increase in the number of vehicles, and thus the percentage of collisions hasn't scaled down like in developed countries [16].

Often, multiple vehicles collide one after another leading to terrible losses and traffic jams [17, 18]. As most of the collisions occur due to human errors (90 % of total collisions) like a distracted driver, over speeding, drunken driving, error in judgment, not following the laid down rules, etc. it becomes necessary to automate driving to reduce these accidents. In general, any vehicle equipped with ADAS systems can better follow traffic rules compared to humans and thus reduce collisions. Automated Cruise Control (ACC) is one such application which enables vehicles to maintain a safe distance with the vehicle in front. Vehicles equipped with such ADAS systems can help reduce collisions.

Cruise control and adaptive cruise control are examples of ADAS systems. Cruise control enables vehicles to maintain a preset velocity without any considerations about the traffic state. ACC ensures that the vehicle maintains a particular speed if there is no vehicle in front (within the detection range of the radar or forward-looking sensors) or maintains a desired range with the preceding vehicle. ACC systems are capable of slowing the vehicle all the way to a standstill and accelerate back and are also known as stop-and-go cruise control systems. ACC enabled vehicle usually do not require communication capabilities and usually operate in a stand-alone mode, in the sense that they do not coordinate with other vehicles rather they act individually.

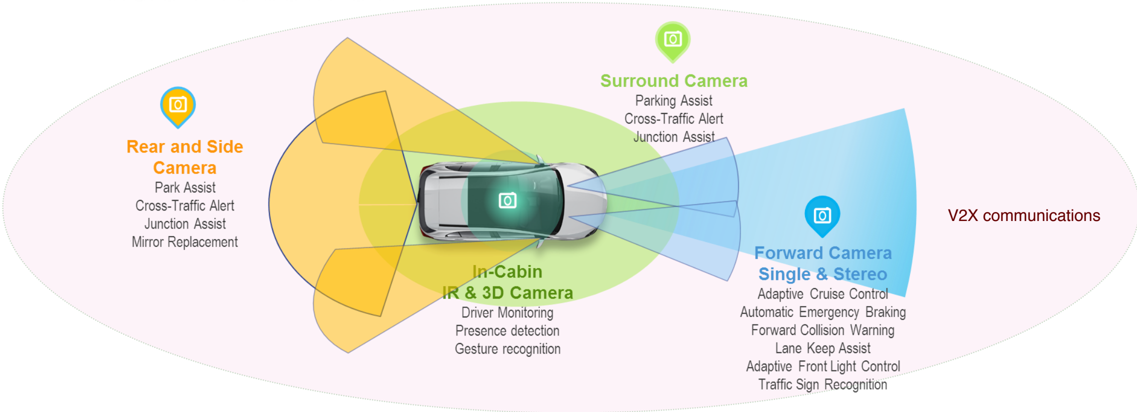


Figure 1.1: Range of sensors and communication modules

In a platoon, multiple such ACC enabled vehicles follow one another usually with short intervehicle distances. Platooning helps vehicles face lower air drag leading to lower fuel consumption [19]. Fuel efficiency can be increased by reducing the intervehicle distance to maximize the reduction in air drag. In a stream of ACC vehicles, any errors or disturbances should not amplify downstream, this condition is known as string stability. Communication delay can result in string instability [20]. At smaller time headways, ACC vehicles may be string unstable, and thus, the intervehicle distance can only be reduced to a certain extent without endangering a collision [21]. If a large time gap is chosen, at higher velocity the intervehicle distance would be larger between two ACC vehicles and manually driven vehicles would cut in and cause the platoon to break [22]. Thus depending on the ACC vehicle, a specific minimum time gap or distance between vehicles is necessary, which indirectly restricts the limit to which the technology can be advantageous.

ACC enabled vehicles usually use sensors and cameras for gathering data locally about neighboring vehicles. Usually, they can only detect the vehicle in the immediate vicinity. The use of communication enables the availability of vehicle information of vehicles even further away. The communication range is generally bigger than the sensing range as shown in Figure 1.1. Thus, sensor limitations can be overcomed by the use of V2V communications. Cooperative Adaptive Cruise Control (CACC) is a feature which can be enabled on vehicles with communication capability like V2X (V2V included) and vehicle control capability. CACC enabled vehicles thus have a broader perception of neighboring vehicles compared to ACC vehicles. The delay in information transfer from the leader to the vehicles downstream can be reduced by using V2X communication facility. Thus the simultaneous implementation of controls is possible due to low latency of V2V/V2X communications which results into quicker control implementations which enable maintaining small headway distances leading to a higher capacity. CACC enabled vehicles are hereon referred to as CACC vehicles. Such vehicles can communicate and cooperate with other vehicles on the road for increased safety and efficiency [23]. Such vehicles are also referred to as Connected automated vehicles (CAVs) in the literature. CACC vehicles form an integral part of the C-ITS system and can reap the bene-

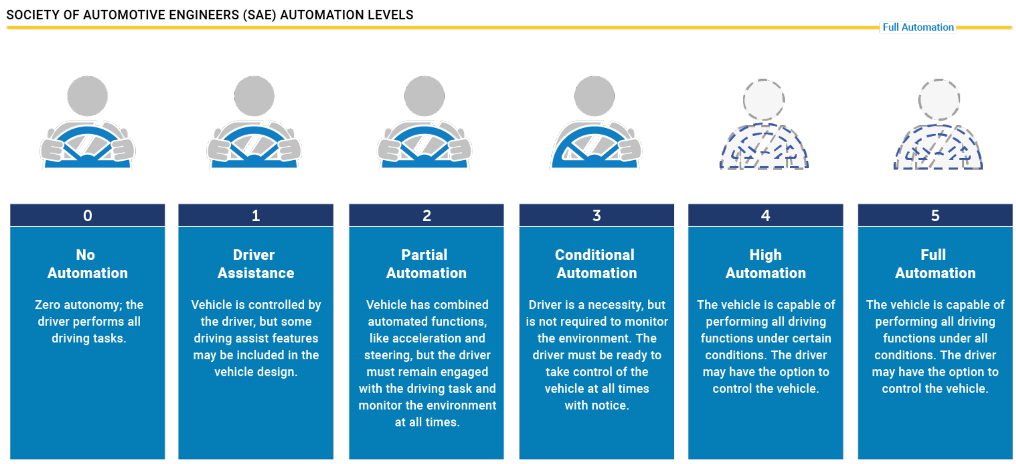


Figure 1.2: Levels of Automation - SAE; source: https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety

fits of various C-ITS applications like mobility, connectivity, soft and hard warning messages, etc.

Control computation of such CACC vehicles can either take place locally or remotely on another vehicle like a platoon leader, on a Road Side Unit (RSU), a cloud based server, etc. Moreover, controls can be optimized for the CACC vehicle (ego vehicle - specific vehicle optimization), or controls can be optimized for all vehicles (global optimization).

Each approach has its benefits and drawbacks. Regarding computational speed and computational efficiency, locally computed controls with limited neighbor information would be quickest and least intensive, but the controls may not be optimal. Centrally computed controls considering broader neighboring vehicle information is slower and computationally intensive, but controls are optimized for all vehicles. Decentralized control computation considering broader neighboring vehicle information might be relatively quick and less intensive, but controls are not always optimal for all vehicles. Near-optimal strategies have though been proposed recently in the literature.

Sharing the road under various levels of autonomy

The above introduced algorithms focus on homogeneous vehicle scenarios, meaning, all vehicles have the same level of automation. Having only autonomous vehicles on the roads is still a distant dream. The market penetration of autonomous vehicles will gradually increase over time. Vehicles with different levels of automation will share the same road. Different levels of automation characterize different levels of freedom a driver has in an autonomous vehicle. In May 2013, National Highway Traffic Safety Administration (NHTSA) released a policy on automated vehicles that defined automation Levels 0 to 4 [24]. The Society of Automotive Engineers (SAE) refined the standards released by NHSTA; they start at level 0 with no automation to level 5 full automation. SAE's levels of automation have been summarized in Figure 1.2 which explains the following:

- Level 0: Traditional manually driven vehicle: the human driver does everything
- Level 1: The automated system on the vehicle can sometimes assist the human driver with a single driver assistance feature such as cruise control (lateral or longitudinal control)
- Level 2: The automated system on the vehicle can assist the human driver with a more than one driver assistance feature such as lane keeping with cruise control (lateral and longitudinal control)
- Level 3: The automated system can both actually conduct some parts of the driving task and monitor the driving environment (like driving on-ramp to off-ramp on a freeway). The human driver must be ready to take back control when the automated system requests;
- Level 4: The automated system can conduct the driving task and monitor the driving environment, and the human does not need to take back control, but the automated system can operate only in certain environments and under certain conditions (like within a geo-fenced area and limited speed limits);
- Level 5: The automated system can perform all driving tasks, under all conditions that a human driver could perform

Heterogeneity of vehicles in terms of their automation introduces various challenges. In a mixed vehicle scenario, different vehicles will have different levels of automation and thus different control behaviors. Safe and efficient control of autonomous vehicles present among other legacy manually driven vehicles is a challenge because of the uncertainty and unpredictability of the behavior of manually driven vehicles.

Safe stop - what is it?

Despite all these positive developments, one of the biggest challenges still facing large-scale deployment of C-ITS services is the inability to detect and handle dangerous situations which could lead to collisions. A hazardous event is a relevant combination of a vehicle-level hazard and an operational situation of the vehicle with potential to lead to an accident if not controlled by timely action [25]. If it is an independent vehicle with stand-alone operation (ACC), internal sensor malfunction, system failure or breakdown would result into an emergency scenario. For CACC vehicles which are cooperating with other vehicles dangerous situations like detecting malicious intent of another vehicle, failure of cooperation between vehicles, external attack like a vehicle spoofing its location or velocity, etc. can derail the vehicle from its nominal operation and the vehicle is expected to be able to detect and resolve such situations.

In short, all vehicles must either handover the driving task to the human or be self-capable to reach a safe state. SAE guidelines mention that a minimal risk

condition is a condition to which either the driver or an Automated Driving System (ADS) may bring a vehicle to halt, after performing the dynamic driving task (DDT) in order to reduce the risk of a crash when a given trip cannot or should not be completed [26]. An excerpt from SAE guidelines mentions different 'minimal risk conditions' for different levels of automation:
"NOTE 1: At levels 1 and 2, the conventional driver is expected to achieve a minimal risk condition as needed.
NOTE 2: At level 3, given a DDT performance-relevant system failure in the ADS or vehicle, the DDT fallback-ready user is expected to achieve a minimal risk condition when s/he determines that it is necessary, or to otherwise perform the DDT if the vehicle is driveable.
NOTE 3: At levels 4 and 5, the ADS is capable of automatically achieving a minimal risk condition when necessary (i.e., due to Operational Design Domain (ODD) exit, if applicable, or due to a DDT performance-relevant system failure in the ADS or vehicle). The characteristics of automated achievement of a minimal risk condition at levels 4 and 5 will vary according to the type and extent of the system failure, the operational design domain (ODD), if any, for the ADS feature in question, and the particular operating conditions when the system failure or ODD exit occurs. It may entail automatically bringing the vehicle to a stop within its current travel path, or it may entail a more extensive maneuver designed to remove the vehicle from an active lane of traffic and/or to automatically return the vehicle to a dispatching facility."
To summarize, level 4 autonomous vehicles must be able to reach a minimal risk condition if nominal operation conditions are violated. This minimal risk condition may be to come to a safe stop.

Putting this safe stop maneuver to perspective, if we consider multiple autonomous vehicles on the road, they might be required to come to a safe stop in the following cases:

- consider a scenario where two streams of autonomous vehicles are approaching an intersection. Under ideal circumstances, these vehicles should coordinate and cooperatively clear the intersection. However, due to any possible issue like communication failure or intersection clearance algorithm failure, if the coordination fails, it is imperative that atleast one stream of vehicles attain minimal risk condition (which can be to come to a total halt) and let other stream of vehicles clear the intersection.

- consider a scenario where a vehicle has broken down on the road, and potential longitudinal collision alert messages (or DENMs) are being transmitted. This may be seen as a hazardous event where vehicles approaching that broken vehicle need to come to a halt to avoid colliding into the broken vehicle or an obstacle.

In the above mentioned scenarios, ADS should resort to safe stop to reach a minimal collision risk state. This safe stop might require emergency braking of vehicles, individually or collectively depending on the types and operation modes of the vehicles.

Path planning (also referred to as motion planning) involves finding collision-free paths over the entire route from the start point (source) to the endpoint (destination). A path is made of multiple segments, and all these segments are linked and continuous. Planning on these segments while considering physical constraints and timing element is called trajectory planning. Thus path planning can be broken down into multiple trajectory planning. Different algorithms on motion planning can be found in the literature survey [27].

In this thesis, we focus on the ADS application and the trajectory planner. The supervisory layer or the application receives sensor inputs, ego vehicle localization information, neighboring vehicle state information which could be either sensed or received in the 'here I am' messages (CAM or BSM), among other data. As soon as a risk of collision is identified, the vehicle will not be in the nominal operation state, and a *safe stop maneuver* would be triggered, the behavior and goals of trajectory planner would be changed. Moreover, it has been proven that for the autonomous vehicle following exercises, longitudinal and lateral movement can be considered independently of one another which reduces complexity [28]. Thus in this work, we only focus on the longitudinal motion of vehicles.

In summary, in this thesis the focus is on *optimal safe brake strategy planning for CACC vehicles in mixed traffic consisting of vehicles with different levels of automation*. The objective is to maximize comfort and ensure safe stop of vehicles without any collisions. When the vehicle state information is received at the centralized controller, the controller computes controls and transmit them back to CACC vehicles, CACC vehicles implement these controls. Manually driven vehicles react to the vehicle in front based on a particular control profile. At the next time slot, updated state parameters are transmitted back to the centralized controller, controls are recomputed, and the loop continues until vehicles reach a safe stop. The system fails if there are any collisions.

There are various constraints and restrictions based on vehicle dynamics, passenger constraints, etc. The system must find an optimal braking strategy while obeying these constraints. The performance of the system is evaluated based on the set objectives: 1. whether collisions were avoided and all vehicles came to a halt or not. 2. the degree of discomfort faced by the occupants of the vehicle.

The above mentioned computation takes place at the centralized controller. Figure 1.3 places the controller in perspective of the data flow. The bock shaded in green is the focus of this study, which involves only processing received data and optimizing controls. The inflow of information and outflow of computed controls, which is desired acceleration takes place over V2V/V2X communications. Security issues impacting intra-vehicle and inter-vehicle communications is an active research topic in the community, but out of the scope of this work.

Coordination under uncertainties

Under ideal circumstances, there would be no errors or uncertainties anywhere in the system, and the performance of the system would be perfect. A centralized control system has a few main components. Localization module on the vehicle helps position the vehicle either locally or globally. The computed localization information is then transmitted to the centralized controller using the communication module. Due to communication failure, transmitted messages might not reach the

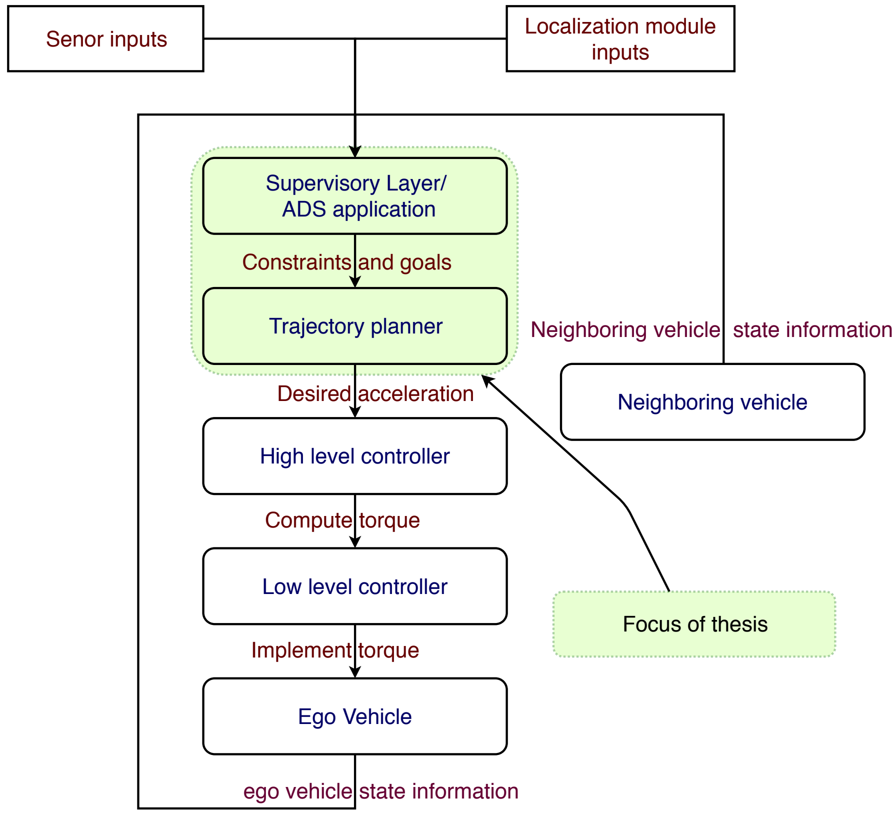


Figure 1.3: Control structure and flow of information

destination creating issues. Communication delay can also cause delayed reception of messages or out of order reception of packets. Computations take place on the centralized controller using the received state information on the uplink. On the downlink, the controller transmits computed controls, and the vehicle's control module implements the received controls. Thus, errors can manifest themselves in either of the following modules: communication or localization or control modules.

Moreover, the centralized controller requires future controls of manually driven vehicles to compute controls for CACC vehicles. As future controls of MDVs cannot be known, they can only be predicted. MDVs could behave differently from the prediction. Thus the presence of MDVs complicates the safe brake procedure. This difference in predicted and actual control behavior of MDVs causes the controls computed by the centralized controller to be optimal only for a short duration. This necessitates frequent recomputation of controls. Model Predictive Control (MPC) helps solve the above issue in a receding horizon fashion, and thus we solve the optimal control problem using MPC.

MPC is a closed loop optimal control and thus helps in effectively mitigating the impact of these errors. In the system being analyzed, there are lots of external errors and disturbances like localization errors, control errors, communication errors, model mismatch, etc. The feedback component helps update the centralized controller with the latest state parameters and mitigate the impact of these errors. Moreover, as the name suggests, it is a predictive control model which uses a prediction model. This model can be changed and thus provides flexibility.

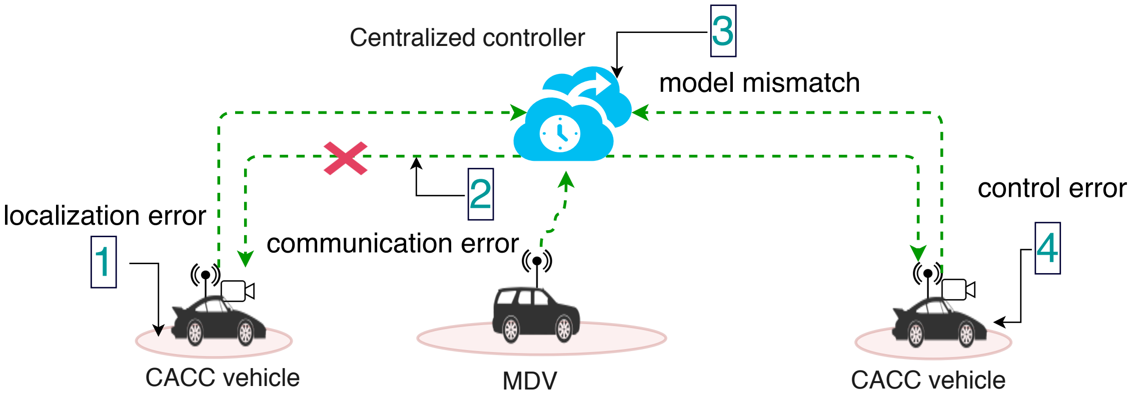


Figure 1.4: Different errors impacting a centralized controller operation

Real-time implementation of the MPC algorithm depends on prediction and control horizon, number of constraints, optimization variables, etc. The work focuses on the evaluation of the proposed algorithm using parameters like collision avoidance and discomfort. There could exist a better suited optimization method to improve efficiency and speed, and it is not evaluated in this work.

Research question

The problem formulation is as follows: 1. Is safe brake possible under uncertainties and imperfections? 2. What kind of performance can a control coordination algorithm provide under these imperfections information?

To summarize, an algorithm that enables safe braking for CACC vehicles in the presence of other manually driven vehicles is required. The presence of uncertainties in communication, control, and localization can lead to faulty controls resulting in collisions. These errors need to be modeled, and the impact of these errors on the controller needs to evaluated. The proposed algorithm needs to be robust to imperfections in communication, control and localization errors to counter and mitigate these errors.

1.2 Methodology

Motivated by the fact that autonomous vehicles will always need to be in a secure state, despite faults in any of the modules like perception, communication or control, in this work, we develop a safe brake algorithm for autonomous vehicles on the road among manually driven vehicles. The impact of different errors shown in Figure 1.4 on the developed algorithm is verified and the performance of the algorithm is evaluated.

Errors and uncertainties impacting different modules can be modeled differently. The thesis first introduces different ways to model communication errors, control errors and localization errors. Next, we implement errors introduced above and analyze the impact of these errors on the centralized controller one by one. We then study the performance of the robust controller and compare it with the non-robust centralized controller under multiple errors. The evaluations are carried out for an event triggered multi vehicle braking scenario considering both cooperative connected and automated vehicles and legacy manually driven vehicles.

The scenario being simulated is a multi vehicle braking scenario with CACC (automated and connected) vehicles and manually driven vehicles. Four vehicles are approaching an obstacle or an intersection and have to brake to come to a halt. The leading vehicle is notified about the obstacle and the centralized controller intervenes and coordinates the braking procedure. Buffer represents a data storage unit on the CACC vehicles which can store received control data from the centralized controller. Controls from the buffer are to be used in case on unavailability of fresh controls in case of communication failure (downlink packet is not received) or when computations are infeasible. Infeasibility can occur when the feasible set of the optimization problem is empty, which can occur, e.g., when the collision avoidance constraint is violated. The performance of the controller is evaluated based on the number of collisions avoided while braking and the discomfort.

We propose the use of Model Predictive Control (MPC) to solve the above introduced issue of safe braking with multiple vehicles on the same road with different levels of automation. The cost function is set to minimize the change in acceleration to minimize discomfort. Discrete time state equations based on kinematic equations are used. Vehicle and passenger constraints like the maximum and minimum acceleration possible, jerk sustainability are used to keep simulations realistic. Using conditions on terminal velocity and the maximum terminal location of the vehicle, it can be ensured that the vehicle comes to a halt before the obstacle (or the intersection). Keeping distance between vehicles always greater than zero enforces collision avoidance. The assumed MDV profile is also used to predict future states of the MDV and this information is used to avoid collisions with CACC vehicles. In short, the computation highlighted in the box below is solved at every time slot until the objective is achieved.

Minimize   (cost function)
subject to
    state equations
    vehicle and passenger constraints
    initial and desired final state parameters
    collision avoidance constraints
    braking condition
    assumed MDV braking profile

The difference in the predicted and the actual control of the MDVs gives rise to a model mismatch. The first set of simulations evaluates the impact of different models giving rise to different magnitudes of model mismatch, on the centralized controller. The permitted change in acceleration was limited to ensure realistic vehicle behavior. The assumed manually driven model which resulted in the least number of collisions and maximum comfort was chosen for further simulations.

The lower level controller may not always immediately achieve the desired acceleration. The performance of the controller was then evaluated under the influence of the lower level controller. Next, the performance of the controller under communication errors is analyzed. Bernoulli model is used to generate random packet loss, whereas to realize multiple consecutive packet losses Markov model is used. The performance of the controller under communication loss modeled using Bernoulli and Markov models is then evaluated.

In the presence of localization errors the performance of the centralized controller is evaluated using a non-robust controller which doesn't account for errors and under a robust controller which accounts for localization errors. A scenario consisting of vehicles with homogeneous localization system was considered, where all vehicles had the same standard deviation of localization. Alternately, scenario consisting of vehicles with heterogeneous localization system was considered, where localization errors were generated with zero mean and standard deviation of 0.25 m for CACC and 4 m for MDVs. Next, the impact of all errors: localization error, communication error, model mismatch, and control error was studied on the non-robust centralized controller and the robust centralized controller. The manuscript details more experiments with different parameters, analysis of simulation results and conclusions.

1.3 Contribution

The goal is to simulate various configurations of a safe brake scenario with multiple vehicles with different levels of automation. Errors and uncertainties impacting different modules can be modeled differently. This work first introduces different ways to model communication errors, control errors, and localization errors. This thesis contributes to the literature by evaluating the impact of one or multiple errors influencing the centralized control operation. Novel ideas to counter these errors, mitigate them and the uncertainties are proposed. These proposed ideas are implemented and evaluated as a controller robust to those errors. In this work, longitudinal control of vehicles is considered.

Key contributions are as follows:

- We devise a MPC based controller specific to the case of braking for autonomous vehicles
- We provide a framework to integrate different kinds of errors into the MPC based framework
- Different kinds of errors are modeled and integrated into the MPC framework to simulate the impact of these errors on a centralized controller, like in a real-world scenario
- We propose solutions to counter these errors and make the controller robust
- The performance of the robust controller is compared to the non-robust controller

- A driving simulator was interfaced with Matlab based centralized controller to simulate the safe brake scenario including human drivers and CACC vehicles. Results and experiment details can be found in the conference publication [29].

- A heuristic simple two-vehicle collision avoidance approach was proposed in [30]. Algorithms developed next could solve multi vehicle safe brake scenarios. Impact of localization errors on the controller was studied and evaluated in [31], [32], [33]. The impact of model mismatch was studied and evaluated in [29]. The impact of communication errors on the centralized controller was the focus of evaluation in [34]. Multiple errors like localization error and model mismatch have been studied, and their impact on the centralized controller has been evaluated in an IEEE transaction publication [35].

1.4 Structure of thesis

The content of the following chapters is summarized in this section for the ease of the reader.

The following chapter 2 introduces related work on the subject of centralized and decentralized control for autonomous vehicles and then lists different vehicle following models found in the literature. The literature on platooning as an application of the autonomous vehicle and methods to ensure collision avoidance in platooning is then introduced. Different types of errors which impact a centralized controller like communication error and localization error are reviewed. Different ways of realizing model mismatch and control errors are introduced. The problem statement is then clarified, and the proposed algorithm is put in place with state of the art.

Chapter 3 analyses the braking application requirements and constraints. A mathematical formulation of the centralized controller is presented. Based on various components of the centralized control system, different sources of error are analyzed and modeled.

Chapter 4 introduces simulation parameters and simulates the performance of the robust controller and compares it against the legacy (non-robust) controller. The implementation and functioning of the buffer are explained. The impact of different errors is studied one by one. First, the model mismatch has been implemented, and the impact of different models is analyzed. The model which is more realistic is chosen for further simulations. The experiment conducted by interfacing Matlab based centralized controller with the driving simulator is explained next. Lower level engine behavior is then implemented as a control error. Different models of communication error and the performance of the robust controller is evaluated using different fall back techniques when packets on the downlink are lost due to communication errors. Next, the performance of the robust controller is compared with the non-robust (legacy) controller for different values of localization error. The final part of this chapter focuses on the evaluation of the impact of all kinds of errors introduced above, on the robust controller. The need for the robust controller is validated by comparing it against the non-robust controller. The chapter is concluded by discussing the sensitivity of the controller to different parameter values.

Chapter 5 derives conclusions from simulations listed in Chapter 4 and provides ways to improve and further this work. Different interesting approaches to make this work adapt to real world applications are discussed. The last part of this chapter lists different publications during the thesis.

Chapter 2

State of the Art

This chapter highlights the state of the art of various algorithms and their shortcomings. The first part of this chapter introduces different platooning algorithms and platoon characteristics, and literature related to platoon performance under communication or localization uncertainties. Methods to ensure collision avoidance between vehicles in a platoon are introduced next. Individual vehicle following models used in the literature are considered next. We then introduce algorithms for collision avoidance between vehicles at different road junctions like intersections, roundabouts, ramp merging, etc. and we highlight the difference between a centralized and a decentralized controller. The second part of the chapter introduces related work on four major types or uncertainties and errors considered in this work: communication error, localization error, control error, and model mismatch. The concept of model mismatch and achievable braking capacity is also introduced, which influence the simulations. The final part of this chapter introduces in short the motivation behind the work and formulates the safe brake problem.

2.1 Collision avoidance between vehicles

2.1.1 Platooning

Platooning is a concept where multiple vehicles follow one another with short intervehicle distances to improve traffic efficiency and throughput, and reduce fuel consumption due to reduced aerodynamic resistance. Fuel consumption reduced by around 10% [19] because of constant speed. According to experiments, at 10 m spacing and a velocity of 80 km/h, the reduction in fuel consumption was about 21 percent [36]. Other authors also have similar results on the reduction of fuel consumption due to platooning [37, 38]. Closer the vehicles, better are the results of platooning, but higher are the risks of collision.

Japans semi-governmental New Energy and Industrial Technology Development Organization, developed a technology for large and small trucks to safely maintain a 4 m distance between vehicles in a single lane while driving 80 kmph [39]. Whereas in Germany, a project at RWTH Aachen University in Germany operated a platoon of four trucks spaced at 10m intervehicle distance [40]. A European project: Safe

Road Trains for the Environment (Sartre) based on platooning has explored using cars and lorries simultaneously at 85 kmph with a gap between each vehicle of 6m[41], whereas [19] mentions the velocity range for heavy vehicle platooning is between 37 to 50 kmph with 10 meters distance between them.

Most of the work has been done on collision avoidance in platoons with vehicles moving in a single lane. However, authors in [42] focus on collision avoidance in a scenario where alternate vehicles from the platoon steer towards right or left to stop somewhere along the diagonal with the vehicle in front. There is a big assumption that the roads have atleast three lanes with platoon moving on the lane in the middle, or in case of roads with just one lane, there is free open space on both sides of the roads. In this chapter, we focus on approaches to avoid collisions in platoons based on different scenarios. Some of the modern day vehicles already have Adaptive Cruise Control(ACC) systems which are usually radar-based systems that maintain a particular velocity in case there are no vehicles in front, or a safe distance with the vehicle in front, but the safety gap maintained is pretty big, and it serves neither purposes of platooning. The distance between vehicles can either be constant, which is known as constant spacing policy or velocity dependent, which is also known as constant time gap policy. In the first type of spacing policy, the distance between the vehicle remains fixed irrespective of the speed of the vehicle [43, 44]. [44] states the recommended headway in Germany is 1.8 sec for manual driving in a platoon scenario. Moreover, [43] considers not just the vehicle in front but also the vehicle behind for autonomous driving.

String stability is another critical measure of the safety of ACC vehicles. If a disturbance or an error in the platoon of vehicles magnifies down the platoon, the system of is said to be string unstable. If the error is absorbed, the system (or the control behavior) is termed string stable. Usually, the use of on board sensors in an ACC system restricts the vehicle only to have information of the preceding vehicle. Thus, the following vehicle only reacts based on the vehicle in front. This creates a delay in information transfer and thus a delay in reaction, which has a substantial impact on the string stability.

A system of three or more ACC vehicles is very likely to get string unstable according to Lu, Wang et al. [45] and this was proven in [46]. A minimum intervehicle distance of 1.2 m can be achieved for two identical vehicles without endangering a collision, assuming that there is no delay present in the feedback system. In case of a delay, intervehicle distance is deduced based upon the vehicles maximum deceleration ability [21].

Milanes et al. mention that ACC in a multi vehicle scenario is unstable because while designing ACC systems, researchers do not consider vehicle actuation and sensor data evaluation delay, whereas in practise it is non-negligible [47]. Moreover, with CACC, the last vehicle in the vehicle stream can have information of the actions of the first vehicle, whereas, in ACC, the last vehicle will only have information of the vehicle in front. Milanes and Shladover recommended time gap for ACC to be set to 1.1 sec whereas CACC is set to 0.6 s.

[48, 49] consider CACC vehicle longitudinal dynamics to be composed of two parts: the ACC part (feedback control) and CACC part (feedforward control). The former is computed using relative distance and velocity sensed by onboard

sensors whereas the latter is computed using the information received over V2V communications.

There may be different means of communication and information transfer in V2V, used for platooning like IEEE 802.11p, Visible Light Communications (VLC), cellular technologies, etc. Authors in [50] propose using VLC as a backup or as an offload measure to DSRC technology as they show that it can very well handle the demands of vehicular communication within a range of say 25 meters, except for the induced time delay leading to an increased safety distance between the vehicles.

2.1.2 Collision avoidance in platooning

One of the main purposes of the introduction of autonomous (ACC/CACC) vehicles is to reduce the number of collisions occurring due to human errors and human factors like fatigue, drowsiness, etc. Automating driving procedure using ACC and CACC control algorithms removes the human factor in driving and can thus be considered as algorithms that reduce collisions. String stability is usually verified to ensure ACC and CACC control algorithms are stable. String stability ensures collision avoidance in a platoon under nominal operational conditions, and other methods are required to ensure collision avoidance in a platoon when the system of vehicles is out of the operational domain.

[51] introduces an algorithm for safe collision free platoon braking in the event of a total communication failure. They assume it takes some time (perception response time t_{prt}) to detect a communication loss, and thus the response can begin only after t_{prt} s. Different situations like hard braking by the leader, hard braking by any vehicle in the middle of the platoon, etc. are simulated with no communication loss or total communication loss. String stability is proved mathematically using transfer functions. When a vehicle in the middle of the platoon starts braking, the original platoon is split into two. The first one with the original leader, whereas the vehicle in the middle which starts braking becomes the leader of the second platoon. As the authors consider all vehicles to have the same braking capability, the proposed system and mathematical model might not be valid if different vehicles have different braking capacities.

[52] focusses on longitudinal control for collision avoidance of ACC vehicle platoon in all conditions. Longitudinal collision avoidance is achieved by keeping a minimum safe distance between vehicles based on the deceleration capacity, and the velocity of these vehicles. On the same lines, [53] proposes vehicle following algorithm based on safe following distance dependent on inter vehicular distance and relative velocity. These approaches are conservative and usually require having a large intervehicle distance thus leading to a decrease in road capacity. [54] proposes laws using which two independent platoons can engage in activities like lane changing, merging of platoons, platoon splitting, etc.

Next, we look at approaches proposed for the ego vehicle collision avoidance with neighboring vehicles. These approaches can be extended to multiple vehicles in the platoon for collision avoidance in the platoon. Long ago, researchers from Ford [55] had a vision of inter vehicular communication for collision warning system. They had formulated a method to avoid collisions by taking into account various factors like time to a collision, time to collision avoidance, human reaction time, warning issuable time, etc.

[56, 57] focusses on rear-end collisions, either the following vehicle should be informed as to when by latest it should start braking[56] or alternately leading vehicle can accelerate at the last moment [57]. [58] focusses on longitudinal collision avoidance based on collision cone approach. The use of elastic band models to compute lateral control to ensure collision avoidance, where different vehicles in the vicinity asserting different forces on the ego-vehicle has also been proposed [59].

In [60], a fixed inter vehicular distance based approach is discussed arising from the idea of the subject vehicle being towed by the vehicle in front using V2V communications only. The goal of the subject vehicle is to align angles using a lateral controller, and use the longitudinal controller to hard maintain a fixed distance between vehicles. Achieving these goals would be difficult given communication delays, non-synchronisation of the information exchange of vehicles, calculation and implementation delays in real life situations with unpredictable behaviors of the leading vehicles. Shorter the turn radius, larger was the error observed.

2.1.3 Vehicle Following Models

Human drivers usually try to avoid collision with the vehicle in the front. Vehicle at the back is usually not considered. Although different humans drive differently, the driving behavior can be generalized. In literature, different researchers have tried to model this driving behavior into driving models. We shall list a few of the most commonly used vehicle following or driving models.

The microscopic model looks at car/vehicle individually; macroscopic model analyses the traffic flow as a whole focussing on characteristics like volume, density, and average speed; mesoscopic analyses is in between macroscopic and microscopic flow models. Mesoscopic models focus on characteristics like headways, spacings, speeds and speed differences. We shall mostly look at the microscopic traffic models which include vehicle following models. Most commonly used microscopic models are based on follow-the-leader concept where the subject vehicle ensures it doesn't crash into the vehicle in front.

Intelligent Driving Model

Intelligent Driver Model (IDM) was introduced in [61] and is one of the most widely used models. IDM or a modified version of IDM is assumed to be a sound basis for the implementation of Adaptive Cruise Control (ACC)/Cooperative ACC(CACC) [62]. Mathematical formulation of IDM can be given as in equation 2.1.

$$
\begin{aligned}
u_i(n) &= u_i^{\max} \left(1 - \left(\frac{v_i}{v_0}\right)^{\delta} - \left(\frac{s^*(v_i, \Delta v_i)}{s_i}\right)^2\right) \\
s^*(v_i, \Delta v_i) &= s_0 + v_i\, T + \frac{v_i\, \Delta v_i}{2\sqrt{u_i^{\max} \cdot b}} \\
s_i &= p_{i-1} - p_i - l_{i-1} \\
\Delta v_i &= v_i - v_{i-1}
\end{aligned} \tag{2.1}
$$

The main feature IDM is the non-linear response to speed difference with the vehicle in front, denoted by the symbol s^*, the desired headway being determined dynamically. i is the vehicle being considered, $i-1$ is the vehicle in front and so on; u_i, v_i, p_i, l_i is the acceleration, velocity, location and length of vehicle i; s_i is the actual distance between vehicles i and $i-1$; $u_i^{\max}$ is the maximum possible acceleration of the vehicle. b represents comfortable braking strength (a positive number in this expression). The desired velocity and minimum distance between vehicles is denoted by v_0 and s_0 respectively. T represents the time headway observed by the vehicle; Δv_i is the difference in the velocities of vehicle i and vehicle $i-1$ in front; δ corresponds to a factor which can be tuned to control the behavior of the vehicle. Bigger the value of δ, more aggressive the reaction of the vehicle in general.

IDM or a modified version of IDM has been used vastly in the literature [45, 47, 63–70]. Arne continues the improvement of IDM as Human driving model (HDM), which can account for driver's reaction times, finite estimation capacities, multi-vehicle anticipation, etc. [71].

IDM+

According to Schakel et al., IDM has realistic shockwave patterns, but the traffic capacity is just below 1900 vehicles/h, Intelligent Driver Model+ (IDM+), expressed in (2.2) was introduced as an improvement over IDM [72]. Rest of the parameters remain the same. Compared to IDM, IDM+ has been used less often in the literature. [73] uses IDM+ in their research.

$$
u_i(n) = u_i^{\max} \min\left(1 - \left(\frac{v_i}{v_0}\right)^{\delta}, \left(1 - \frac{s^*(v_i, \Delta v_i)}{s_i}\right)^2\right) \tag{2.2}
$$

Krauss

Krauss's model gives the safe speed a vehicle should maintain to ensure a collision free ACC operation [74].

$$
\begin{aligned}
v_{i,safe} &= v_{i-1}(n) + \eta \\
\eta &= \frac{s_i(n) - v_{i-1}(n) * t_{prt,i}}{t_{prt,i} + \frac{v_{i-1}(n) + v_i(n)}{2u^{\min}}}
\end{aligned} \tag{2.3}
$$

where $t_{prt,i}$ is the perception response time and $u^{\min}$ is the maximum deceleration (braking) of the vehicle. This $v_{i,safe}$ value must be bounded by the achievable acceleration in the next time slot and maximum speed restriction of the road.

Gipp's

Gipp's model is an accident-free model which generates acceleration requirements in a realistic range. Even if the vehicle in front decelerates suddenly to a complete stop, the distance gap to the leading vehicle should not reduce below the minimum gap s_0. The following Gipp's model has been adapted to the original Gipp's model by adding a constant reaction time Δt as in Chapter 11,[75].

$$
\begin{aligned}
v(t + t_{prt}) &= min(v + at_{prt}, v_0, v_{safe}(s, v_{i-1})) \\
v_{safe}(s, v_{i-1}) &= -bt_{prt} + \sqrt[2]{b^2 t_{prt}^2 + v_{i-1}^2 + 2b(s - s_0)} \\
s &\geq s_0 + vt_{prt} + \frac{v^2}{2b} - \frac{v_{i-1}^2}{2b} \\
\Delta x &= vt_{prt} + \frac{v^2}{2b} \\
\Delta x_{i-1} &= \frac{v_{i-1}^2}{2b}
\end{aligned}
\tag{2.4}
$$

Assumptions in Gipp's model:
1. Vehicles always brake with constant acceleration 'b' (although we can assume different vehicles to brake with different braking strength, the formula doesn't implicitly say so).
2. The model assumes a constant reaction time t_{prt}

More car following models can be found in the literature. Please refer to [75,76] for a detailed list.

2.1.4 Coordinated CAV applications

There are various applications based on V2V communications which will help reduce traffic and accidents in the future. Some of these applications which have interested researchers are platooning, coordinated intersection clearance, ramp merging and round about clearance. Platooning has already been detailed before, in this section, we look at the other applications.

When multiple vehicles are approaching an intersection, ideally, they should clear the intersection one after another without any vehicle requiring to come to a halt. Moreover, the intersection should be occupied by the next vehicle as soon as the vehicle before quits the intersection. These intersection clearance algorithms can either be centralized or decentralized. Rios-Torres et al. [77] produce a well-written survey of work on centralized and decentralized coordination for collision avoidance and intersection clearance at intersections. Intersection clearance algorithms are evaluated by comparing their performance with the results on traffic light based approach.

[78] Gives an overview of the coordination between cooperative autonomous vehicles for collision avoidance in intersections and other cases. Three major issues to solve coordination problems namely, sensing (and signal processing), communication, and control are discussed. Techniques used for vehicle coordination can be

broadly split into 1. Rule-based coordination, 2. Optimization based coordination. Different centralized and decentralized control algorithms are discussed. [79, 80] introduces the concept of decentralized coordination problem solving and centralized coordination method for intersection management. The coordination strategy decided by a centralized server is optimal as all vehicles implement controls generated by the centralized server. The decentralized system has two steps:
1.: Decision order: It decides the sequence in which the participating vehicles are offered to make their control choice. there are multiple options: FIFO, distance to the intersection, etc. to decide the ordering. 2.: Sequential control computation: Each independent vehicle makes a control decision such that it either enters the intersection after all vehicles (1... i-1) have left or enters the intersection before any of the vehicle (1...i-1) has entered.
They mention the complexity is lower in distributed approach and doesn't depend on the number of agents in case of a decentralized approach as collision avoidance is enforced by local state constraints at given time stamps. [80] mentions the requirement of having an emergency mode if the participating vehicles which cannot find any solution to the coordination problem.

[81] focuses on decentralized control approach for intersection crossing and proposes two algorithms. First, prioritizing access to the intersection based on their time of arrival and second, based on the inertia of the vehicle and time of arrival at the intersection. [82] focuses on intelligent intersection clearance and collision avoidance with the vehicle in front while approaching the intersection. They use a hybrid system between a centralized and distributed system for intersection clearance.

[83] focuses on a centralized intersection management issues where multiple vehicles need to coordinate to pass through an intersection. Vehicle dynamics and physical restrictions act as constraints to this issue and sum of local costs (at all vehicles) is optimized to get the best strategy to clear the intersection.

[84] Authors adopt a simple interesting approach to solve the issue of intersection collision avoidance, they use Pontryagin's minimum principle and Hamiltonian equations to get closed form solution. To compare results of their algorithm, they assume the base case where all vehicles coming from the main lane pass by freely whereas those coming from the side lane/merging lane need to come to a full halt and then enter. The flaw is that the decision making order is based solely on distance from the intersection (thus, when velocities of vehicles are not the same, their decision order fails).

[67] proposes a centralized control algorithm robust to localization errors for intersection management in mixed traffic scenario involving conventional, connected and automated vehicles. Optimal intersection clearance strategy is found using branch and bound algorithm used in a tree search problem. Their proposed algorithm is compared with the scenario where: traffic signal is controlled using traffic sensors placed on the roads to estimate the traffic statistics, etc.

[85] focuses on evaluating communication requirements in a centralized coordinated intersection management system for CACC vehicles; they provide results in terms of transmission power, the probability of reception of a packet in the uplink and in the downlink for designing centralized controllers. Whereas [86] evaluates the centralized control algorithm for autonomous vehicle intersection clearance under

unreliable uplink communication.

Schildbash et al. focus on a centralized Robust MPC (RMPC) based collision avoidance (intersection clearing) system for a mixed vehicle scenario [87]. The centralized controller finds safe gaps in the crossing traffic and optimizes the longitudinal motion to make the CACC vehicle cross the intersection (rest of the vehicles are MDVs). Authors counter uncertain MDV behavior by assuming the worst case (minimum and maximum) values of acceleration possible to create a robust MPC based technique for intersection clearance in a mixed vehicle scenario [87]. Robustness comes from the fact that despite the uncertainty of the future control of MDV, the maximum and minimum acceleration of MDV is considered to have a safe intersection clearance.

Controls of CACC vehicles are influenced to improve the macroscopic performance of the system like improving the optimal capacity of roads with traffic inflow from one end of the highway and on-ramp [66]. Lane mergings and overtaking scenarios are also handled by cooperative maneuver planning in [88]. Roundabout clearance in heterogeneous traffic has also been studied using predictive control strategies [89]. To summarize, platooning, lane mergings, ramp merging, etc. are a few of the CACC vehicle applications.

2.1.5 Centralized vs decentralized controller

In a centralized system, the coordination strategy is decided by a centralized server, which informs all participating vehicles about controls to ensure optimal performance. The complexity of a centralized controller depends on the number of vehicles to be coordinated and the number of constraints. Moreover, in a centralized CACC platoon, CACC vehicles do not (are not needed to) have a particular control profile as they implement controls received from the centralized controller.

A decentralized system can have different implementations. Each vehicle can have a control profile, and they compute controls based on this profile using state information of the preceding vehicle (decentralized ACC platoon) or multiple preceding vehicles (decentralized CACC platoon). Another type of decentralized CACC platoon can be one in which each vehicle sequentially computes controls [80]. Such kind of a decentralized operation usually requires two steps: 1. Model-based decision heuristic (deciding the order in which vehicles shall compute controls - prioritizing vehicles) 2. Sequential control computation (based on the priorities assigned, compute controls). The performance of one such decentralized controller is computed against a centralized controller in [80] in an intersection clearance scenario. [79] mentions the complexity is lower in distributed/decentralized approach and doesn't depend on the number of agents as collision avoidance is enforced by local state constraints at given time stamps. A decentralized system can have varying levels of performance based on the implementation of the control computation. If all vehicles take decisions locally, without any knowledge of what other participants are doing, it can result in a sub-optimal system.

2.2 Types of errors

2.2.1 Communication error

Communication delays impact a centralized and a decentralized controller differently. In a decentralized control system, every vehicle broadcasts their state parameters and control computations are done locally by each vehicle. A centralized controller requires transmission of vehicles' state parameters to the centralized server (computational unit) on the uplink and computed controls back on the downlink to the vehicles. Communication impairments like packet delays, packet losses and out-of-order delivery of packets can thus manifest themselves either on the uplink or on the downlink. Out-of-order delivery of packets can be addressed by discarding the newly received packet if the time stamp of a newly received packet is older than that of the last received packet [90]. The occurrence of out-of-order delivery of packets also indirectly signifies packet delays (and/or losses).

End to end delays arising in a centralized control system can be categorized into the following:
1. propagation delays: unequal propagation delays for different packets causes delayed, and disordered reception and packet losses (sometimes due to very long propagation delays) are inevitable in V2X communications [91].
2. Delay introduced at the PHY layer: Multiple nodes might need to compete for the right to access the network causing an additional time delay.
3. Delay introduced due to MAC layer algorithm: Different data types could have different priorities, (introduced in Enhanced Distributed Channel Access (EDCA) as an improvement over Distributed Coordination Function (DCF)), which can introduce delays of varying magnitude as well.

This paragraph looks at the modeling of communication errors in a centralized control scenario. Communication delay on the uplink is understood as feedback delay which results into a difference in the received state and the actual state information if V2V communication is used (or difference in measured state and actual state if onboard sensors are used). This issue arising due to communication delay is considered as model mismatch [73]. In the case of uplink packet loss, the predicted behavior of MDVs can be used to estimate MDVs' state parameters. Control values from the last transmission on the downlink can be used to predict future states for CACC vehicles. These estimated states can be used to compute controls [92]. A centralized intersection manager coping with communication errors on the uplink is introduced in [93]. Nazari et al. compute a closed form expression for centralized intersection coordination of automated vehicles valid under different communication conditions like no packet reception, all packet reception, and few packets reception on the uplink using Bernoulli's random variable; downlink communications are assumed to be perfect [86].

Continuous periodic uplink data transfer is assumed, and event-based downlink communication is proposed [94] to reduce the load on the network. Delay threshold of a downlink communication channel which allows collision free control of vehicles is analyzed. At the end of the threshold, as there would not be any control information at the vehicle, emergency braking must be activated. Communication requirements based on control algorithm to ensure collision avoidance is discussed [94]. The impact of communication disturbances on centralized and decentralized controllers impacting a centralized intersection clearance algorithms has also been surveyed [77]. Design guidelines for both uplink (whereby vehicles send intentions to the central controller) and downlink (where the controller prescribes vehicles of safe control actions) are suggested based on the communication system analysis for the centralized intersection crossing coordination [85].

Team AnnieWAY observed that vehicles not transmitting anything or some outdated data result into unavailability of communicated information, resulting in problems during cooperation during the Grand Cooperative Driving challenge [95]. Authors assume, in case a packet is not received, it will be retransmitted, thus effectively assuming, different communication delays for different vehicles and no packet losses [96]. End-to-end communication delays of the IEEE 802.11p communication system is assumed to be 0.09 sec by authors in [97]. Naus et al. implemented 802.11g for V2V communication and found the communication delay of 10 ms during a CACC vehicle platoon experiment [98], whereas Ploeg et al. found a delay of 150 ms in a V2V communication based on 801.11a [99]. In [100] Ploeg assumes a nominal value of 0.02 s as the CACC communication delay time. Xu et al. model communication delays in 3 types: 1. fixed delay 2. distance dependant delay 3. random delay. Distance dependent delay is assumed to be squarely proportional to the distance between vehicles [101]. [102] assumes a communication delay of 22 ms, and they design a min-max MPC controller robust to communication delays. Dey et al. [103] review the impact of communication delays on different control algorithms operating ACC and CACC vehicles in a decentralized control scenario.

Relatively less work exists on the performance of the controller under communication failure. Saxena et al. choose the channel such that it allows 80 % message reception between 2 consecutive vehicles, 75 % between alternate vehicles and 70 % if 2 vehicles are in between transmitting and receiving vehicle [104]. On the other hand, 1% packet loss has been assumed by Massera et al. to simulate a realistic communication environment [102].

Performance of a platoon of CACC vehicles under both centralized control system and a decentralized control system in the presence of communication delays has been studied by various researchers and we summarize them next. A prediction model is used to predict the position of vehicles in case of delayed communications [96] to counter delays. Li et al. propose a low-latency driving command dissemination (LCD) algorithm to reduce communication delay in a vehicle platoon [105].

Ploeg et al. propose adjusting the time headway of the decentralized CACC vehicle platoon to ensure the platoon is string stable; they derive a relation between V2V communication delay and the time gap that ensures string stability [99]. String stability of a decentralized control system in case of packet delay or loss has been studied previously [101]. Alternately, Ploeg in [106] suggests a fallback to ACC algorithm, which would lead to an increase in inter-vehicle distances, necessary to maintain string stability in case of a communication error in CACC platoon. They also propose the implementation of estimated acceleration of the preceding vehicle computed using a Kalman filter to counter communication delays. In some other work [100], Ploeg suggests, to switch from 1-vehicle to 2-vehicle look ahead while maintaining the same headway when if the time delay is high. A relation (graph) between headway distance for vehicles in a platoon and the communication delay for the platoon to be string stable is also provided. However, if the time delay is too high (degraded conditions), it is best to avoid the use of wireless communication [100].

If no packets are received, Saxena et al. make rule-based predictions (rules like the vehicle in front maintains same velocity or same acceleration) to compute where the vehicle in front would be and used this prediction to compute controls [104] in a decentralized control scenario. String stability of a CACC platoon where control strategy of the vehicle depends on the data reception status from different vehicles changes at each instant is computed as well. Whereas [94] introduces the concept of event-based transmission in a centralized controls scenario to reduce the load on the communication channel to reduce failures.

To summarize, some work in the literature tries to reduce communication errors (like communication failures) whereas other literature focuses on reducing the impact of communication errors on the control system.

2.2.2 Localization error

Any autonomous vehicle needs to be aware of its localization, destination and neighboring vehicles' state information for navigation. Collisions can be avoided if vehicles have a local dynamic map of their neighbors, for which localization is necessary. For a safe trip, an autonomous car would like to have the following information:

1. Where is the car
2. What is around the car
3. What should be car's next decision (change in direction, velocity, etc.)

One of the critical issues faced by the participants of the Grand Cooperative Driving challenge was noisy position estimates (sometimes) or total blackouts under a highway bridge [95] leading to poor localization estimate or no localization estimate at all. Our focus would thus be on vehicular localization techniques, their accuracies, and the impact of uncertainty in localization on safety applications.

The use of erroneous localization information can lead to erroneous controls which could potentially result to collisions. Emerging C-ITS applications demand either road-level, lane-level and where-in-lane-level localization accuracy depending on their applications [107]. 1 Hz frequency would be enough for applications working on road-level positioning whereas 10 Hz updates are needed for lane-level and where-in-lane-level positioning [108]. Requirements of 0.5 m and 10 Hz as positioning accuracy and the position update rate for C-ITS applications respectively have been expected [109]. Until now, most of the research work is based on achieving localization in the horizontal plane latitude (lat), and the longitude(lon) coordinates with the desired level of accuracy. Although there is a field dedicated to 'elevation' reading according to SAE's DSRC implementation guide, it is not mandatory [110].

Different localization techniques including satellite based localization, stereo camera based localization, etc. have different accuracy. Most widely adopted global localization system is Global Navigation Satellite System (GNSS), usually referred to as Global Positioning System (GPS). Performance of GPS based localization system depends on two attributes. First, the availability of GPS services. There are serious limitations on successful 'locking' of GPS in urban cities with high rise buildings, low chances of having a clear view of the sky, multipath, etc. Second, the accuracy of GPS positioning solution when available. If Global Navigation Satellite System based localization technique is used, localization can be achieved with a std of error of 4 m [111]. Well constructed GPS receivers achieve a horizontal accuracy of 3 meters or better and vertical accuracy of 5 meters or better 95% of the time [112]. In Europe, European Geostationary Navigation Overlay Service (EGNOS) supports and improves GPS based localization by providing correction data and integrity information based on a network of ground stations and three geostationary satellites to users in Europe and achieves an accuracy of 3 m horizontally and 4 m vertically with 95% confidence bound [113]. Real-time kinematic (RTK) and Differential GPS (DGPS) are two techniques which enhance the performance of localization using GNSS to centimeter level and decimeter level accuracy respectively [114–116]. The accuracy of a couple of meters was achieved for an autonomous-off road vehicle [117] by fusing DGPS and inertial navigation systems.

European project 'GLOVE' [118] was sponsored to improve OBU services for positioning and other vehicular applications by focusing on cross-domain integration between Galileo/EGNOS GNSS (Global Navigation Satellite System) and Vehicular Ad-Hoc Networks (VANETs). HIGHTS is an example of a European project with a goal to achieve a high precision positioning system with the accuracy of 25cm for autonomous vehicles as a part of the ITS [119]. Authors in [115] propose to fuse GPS and INS readings and evaluate the performance in the case where GPS is unavailable and in the case where GPS is degraded. They conclude that relying on INS as a standalone measure gives more errors compared to GPS and INS both when available. [117] proposes to merge readings from DGPS and INS using Kalman filtering technique for better localization using different commercially available receivers for land rovers.

An indirect V2V localization scheme where each vehicle localizes itself locally based on laser scanner data generating a real-time local map, and merging such data from various vehicles to obtain a broader perception of the system has been studied [120]. This SLAM based system helps not only in navigation locally but also helps the global system for cooperative localization. The issue is that merging of maps may need transmission of a lot of data and inclusion of new headers/information in packets. Although the results show interesting improvements of cooperation based localization over individual localization, there is a lot of uncertainty over the communication module for V2V to transfer the required data. Localization based on lidars or laser scanners faces issues and fails when curbs are broken or damaged or when there are multiple lanes or when 2 wheelers are obstructing the view of curbs, etc. [121].

Received Signal Strength Indicator (RSSI) of received CAMs have been used for localization as well [122]. Levinson and others propose a map-matching based technique, where live data is compared with the already downloaded environment map which provides decimeter level accuracy [123]. They integrate data from GPS, inertial measurement unit (IMU), wheel odometer and LIDAR to achieve this. They mention two benefits of the proposed system over GPS in terms of availability and accuracy. The disadvantage of the technique is that it relies heavily on maps. The lateral accuracy achieved was around 5 cm. This work was further improved in [124] by Levinson and Turn (one of the founders of Google car project). They consider maps as probability distributions over environment properties which helps them represent the world/environment more accurately and localize with fewer errors. They achieve 10 cm level overall localization accuracy (lateral and longitudinal) in real life experiments, finding it more than sufficient for autonomous driving. Map matching techniques have been implemented on autonomous vehicles which have covered hundreds of miles with std of accuracy better than 10 cm [125]. [126] focuses on using a single monocular camera for localization. Their algorithm requires 3D ground map generated by survey vehicles using 3D lidar scanners. Although the results do not have the same level of accuracy as [123, 124], they manage to achieve an accuracy in the longitudinal direction between 20 to 45 cm. Cooperative localization techniques in VANETs by fusing GNSS and Infrared range measurements managed to achieve an accuracy between 20 to 40 cms [127].

Different authors in literature have assumed different localization accuracy. [128, 129] refers the inter vehicle distance error is in the range of 0.3 m. In their experiments, they use localization accuracy of GPS based systems (on CACC vehicles) to be between 2 and 6 m. In [102] disturbances of 3 m in the distance between vehicles were added at random instances to check if the controller is robust to localization errors. Yang et al. in [67] implement localization errors with std of 15 m. Brown noise (Wiener process) and Gaussian noise (AWGN) are combined to generate localization errors as well [94], but the motivation behind it is not understood. Whereas 10 cm accuracy has already been achieved and autonomous vehicle with this accuracy is already out on the roads for test drives [123]. In short, a wide range of magnitudes of localization error is used, but not a lot of literature can be found on testing the performance of the control system in the presence of localization error.

2.2.3 Control error

Any vehicle cannot instantaneously achieve the desired acceleration, thus there exists a difference between the desired control and the actual control (acceleration). This difference in control is termed as control error in this work. The source of control error is the engine of the vehicle and is thus an internal source of error. Ideally, every vehicle will exhibit different engine behavior based on the engine tuning and engine properties. Usually, this engine behavior can be represented as a low pass filter, meaning, at low frequency, the gain is one, and the output is similar to the input. However, at high frequency, the gain is less than one, and the output can not keep with the input.

Usually, the ADAS application or the upper level controller computes the desired acceleration. Desired acceleration is then converted to desired torque, which is converted to required throttle. The engine that implements the computed throttle angle to provide the torque required to generate the desired acceleration. It is the lower level controller which computes the required throttle value and implements it. The lower level controller is associated with a finite bandwidth because of which the desired acceleration and the actual (observed) acceleration are different. A first order lag is thus assumed in the implementation of the lower level controller [130].

To summarize, actuator delay is the delay after which the vehicle begins to implement the torque whereas actuator time constant is the behavior of the controller, which is also known as the engine time constant. It is the engine time constant which gives the controller properties of a low pass filter.

In Plexe [131] Michele Segata and others have modeled the engine behavior introduced by power-train dynamics as a first order low-pass filter and implemented it as follows:

$$u^*(n) = \beta u(n) + (1 - \beta)u^*(n - 1); \tag{2.5}$$

where

$$\beta = \frac{\Delta t}{\tau + \Delta t} \tag{2.6}$$

$u^*(n)$ represents the applied (effective) acceleration whereas $u(n)$ represents the desired acceleration. τ is the time constant of the engine, Δt is the sampling frequency (update step) in seconds.

Assuming τ to be constant, β is small if Δt is small, and at low values of β, $u^*(n) \approx u^*(n - 1)$. Small τ represents high sampling frequency. Where as β is large if Δt is large, and at large values of β, $u^*(n) = u(n)$. Large Δt signifies a low sampling frequency. It is because of this reason that the engine can be said to have low pass filter characteristics.

Using the nomenclature introduced earlier, control error arising from the lower level controller can also be presented as follows [130]:

$$u^* = \frac{1}{\tau s + 1} u \tag{2.7}$$

[132] models control error as a fixed delay in implementing the desired acceleration. i.e.: The value of realized acceleration is the same as the desired acceleration after a certain delay due to actuator lag.

If both, actuator delay and engine constant is considered [130], the control behavior can be represented as:

$$u^* = \frac{1}{\tau s + 1} u^{\mathrm{T}s} \tag{2.8}$$

where T is the actuator delay.

[133] adds communication delay to the actuator delay to compute the effective time after which the controller implements the torque. The realized acceleration is thus a factor of the desired acceleration generated from the MPC controller, the and the previous acceleration.

Alternately, control errors arising from the lower level controller can be implemented by assuming the realized (implemented) acceleration to be a fraction of the desired acceleration, with an assumed delay of 0.5 sec with minimum and maximum acceleration being $-0.5\,\mathrm{g}$ and $0.25\,\mathrm{g}$ [130], [(chap. 6, pg 145)]. Sensor delays ranging from 0.1 s to 0.3 s and actuator delays ranging from 0.1 to 0.2 s are considered, and Meng et al. [92] evaluate their impact on a stream of ACC vehicles. Issues arising because of actuator lag and sensor delays are:

- when actuator lag is ignored, actually implemented acceleration is different from the desired acceleration (the difference between open loop and closed loop- because in a closed loop, actuator delay can be known and can be used)
- the predicted and the actual velocity of neighboring vehicles are different (it is assumed that neighboring vehicles move at the same speed or same acceleration based on the setting, which need not be the case actually)
- sensor delay(or communication delay): due to sensor delay, the current state needs to be predicted based on the last state of the ego vehicle

To compensate these delays, Meng proposes:

- compensate actuator lag by using the third order system dynamics model for prediction instead of the second order model
- compensate sensor delay: model-based estimation of the current system state based on the previous system state; use this predicted system state to compute controls.

For an Electric vehicle, $\tau = 0.015$ is implemented [134]. Whereas a MPC based system for an ACC vehicle was simulated with the engine constant $\tau = 0.45$ [135]. Massera, Wolf et al. use the worst case engine constant of 0.1 s with unbounded jerks in their min-max control formation [102]. Actuator delay and engine behavior are both considered by van Nunen et al. while designing and testing a robust MPC controller [133]. Sensor delay is considered to be feedback delay which is accounted for in the Robust MPC formation with a value of 0.2 s along with actuator delay of 0.2 s while designing a robust MM-MPC by Chen et al [73]. Ge et al. consider sensing delay for ACC vehicles to be between 0.05 and 0.2 s while evaluating the dynamics of a connected vehicle system with feedbacks [136]. [137] considers actuator lag of 0.2 s.

To summarize, some of the literature implements control errors by using a first order low pass filter with an engine time constant and actuator delay, whereas a significant portion of them only use a first order low pass filter with an engine time constant.

2.2.4 Model mismatch

Mixed vehicle scenarios have not received enough attention in the literature. Even among the work that focuses on mixed vehicle scenario, usually, a fixed profile of MDVs is assumed, and then the autonomous vehicle adjusts or derives the control (centralized or decentralized control) based on the MDV behavior. Very few authors have considered model mismatch as an error impacting the performance of the system of vehicles.

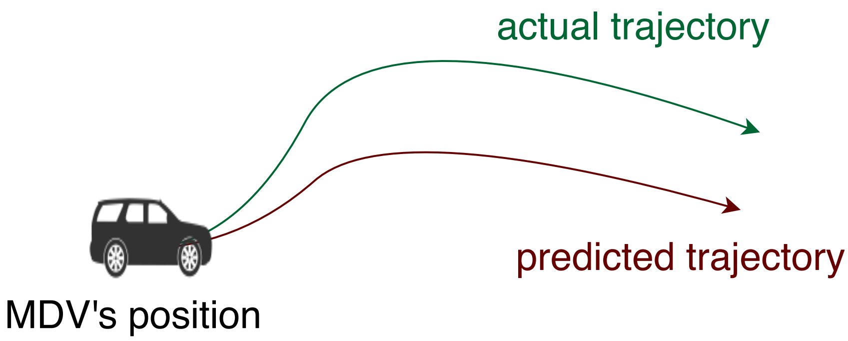


Figure 2.1: Different trajectories generated due to model mismatch

Model mismatch arises due to the difference in the predicted and the actual control of the MDVs resulting in different actual and predicted trajectories as shown in Figure 2.1. Usually, longer the prediction horizon, more can the magnitude of error be. Controls generated using predicted trajectory (or predicted control behavior) of MDVs different from the actual behavior and can result in collisions, and thus model mismatch can be considered as a source of error. The prediction model is at the centralized controller, and thus model mismatch can be classified as an internal source of error.

Methods in which model mismatch has been implemented in the literature are highlighted next. [73] model mismatch is produced because of difference in parameters received and the actual parameters of that vehicle because of the feedback delay caused due to sensors (sensor delay) or communication (communication delay). In heterogeneous platoons (mixed traffic) they assume model mismatch to arise from uncertainties/unknown parameters of α and b of IDM+ and actuator delays (mixed MDV and CACC scenario). In the case of homogeneous platoons, the model mismatch is only because of feedback delay. [138] implements model mismatch by assuming different parameter values of vehicle engine model. They believe these parameters depend on the operating conditions and external disturbances like wind, road slope, etc.

Xu et al. consider model mismatch as external uncertainty arising due to the difference between the prediction (computed using assuming all vehicles other than ego vehicle continues to move at the same acceleration) and the actual actions of road participants [139].

Next, we list different methods used to counter model mismatch in the literature. Chen et al. counter model mismatch by using a Min-Max based robust longitudinal controller for a platoon of vehicles [73]. One of the other methods to improve model mismatch is an online adaptation of the model for improving prediction accuracy which is also known as online calibration [70] or online parameter estimation for modeling driving behavior of MDVs [140]. Parameters like 1. reaction time (perception response time) and 2. driving behavior like observed headway and velocity can be updated over time (at each computation) to get a better MDV prediction model. This online computation requires advanced machine learning capabilities and high computational power.

Our model is a simpler version of the same, where the predictions generated by the model adapts to the observation. The proposed model is introduced later in the next chapter in section 3.2.1.

2.2.5 Perception response time

Two major human factors affecting MDVs are the reaction time and the limited visibility. This *perception response time* (PRT) is added to imitate response delay of a human driver [32,141]. We assume MDV's visibility to be limited to the vehicle in immediate front only. We define $t_{i,i-1}$ and $t_{i,1}$ as the pair of PRT of a MDV i compared to the vehicle in front and the delay vehicle i shows before it starts to brake with respect to the first vehicle respectively. $t_{i,1}$ is also thus called the effective perception response time. Moreover, we assume MDV i would react $t_{i,i-1}$ seconds after vehicle $i-1$ and $t_{i,1}$ seconds after vehicle 1. And $t_{i,1}$= $t_{i,i-1}$+$t_{i-1,i-2}$+...+ $t_{2,1}$ if all $2, 3..i$ front vehicles are MDVs (refer to Fig. 2.2).

CACC vehicles (labeled CACC in the diagram) are assumed to be able to implement received controls synchronously. In heterogeneous traffic where there is a mix of CACC and MDVs, the t_{prt} of the MDV would depend on the number of vehicles between itself and the immediate CACC vehicle in the front. From the figure, we can see that for vehicle 3 (i=3), $t_{3,1} = t_{3,2} + t_{2,1}$. Although the perception response time of vehicle 3 is $t_{3,2}$ as it reacts $t_{3,2}$ s after vehicle 2, the effective perception response time is $t_{3,1}$. The fourth vehicle is CACC and thus can begin reacting at approximately the same time as the first CACC vehicle. Whereas for i =5, $t_{5,1} = t_{5,4}$ as the vehicle in front is a CACC vehicle.

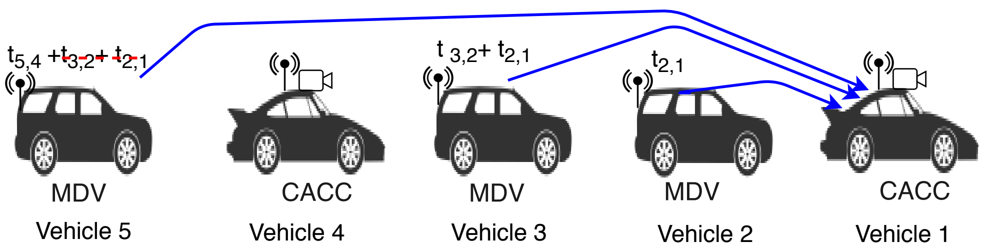


Figure 2.2: Delays in reaction due to finite reaction time of MDVs.

In this work, we use the effective perception response time (now on, referred to as the perception response time of the vehicle). Perception response time values are drawn from a normal distribution $\mathcal{N}(1.33, (0.27)^2)$ and are capped between 0.8 s and 1.8 s [141].

2.2.6 Achievable braking capacity

Braking capacity of each vehicle can be different, and it describes the maximum deceleration strength of the vehicle. More the value, higher the deceleration. In literature different range and distribution characteristics of the braking capacity has been used.

Braking distribution with a mean of 0.6g, and standard deviation(std) of 0.1g, within a minimum of 0.3g and a maximum of 0.8g has been used by Brunson et

Table 2.1: Braking capacity under different environmental conditions

	Maximum Deceleration [m/s^2]	Visibility [m]
Optimal weather conditions	8	200
Wet road	6	200
Dirt on the road	4	200
Fog	8	50

al. [142]. Another interesting article [143] provides different deceleration strengths under different weather conditions and visibility range. The values have been summarized in the table below:

Moreover, different drivers of different age groups based on their physical state, may or may not be able to achieve the maximum braking capacity that the vehicle can offer. Unless mentioned, in most of the work we have assumed the maximum achievable braking strength to be 0.6 g. The impact of vehicles with different achievable braking strength has been studied previously [30].

2.3 Problem formulation

This section introduces the problem and summarizes in short the motivation behind choosing this particular subject for this thesis.

An autonomous vehicle operation can be impacted by various internal factors like onboard system failure, sensor failure, etc. or by external factors like risky maneuvers by immediate neighbors threatening a collision, sudden change in road conditions, etc. Moreover, vehicle coordination algorithms may fail, in this case, autonomous vehicles will need to make decisions for themselves [80, 144]. In such situations when conditions dynamically change and the nominal operational condition is violated, an autonomous vehicle must have the capability to reach the minimal risk condition. Bringing the vehicle to a safe stop without any collisions is one of the ways to achieve this.

A few researchers have already proposed solutions to the safe stop problem. A centralized longitudinal collision avoidance based on MPC for coordinated braking by minimizing the kinetic energy between consecutive vehicle pairs is introduced by researchers from PATH, University of California, Berkeley [45, 64]. But they consider perfect communication and sensing and they do not consider mixed vehicle scenario. Heterogeneity is introduced by considering vehicles with different braking capacity. Alternately, Svensson and Masson suggest storing a pre-generated set of trajectories in an offline database [144]. When the safe stop maneuver needs to be executed, the vehicle chooses the trajectory with least cost based on the vehicle dynamics, other vehicles on the road which act as obstacles, etc. This approach is decentralized and involves no coordination with other vehicles.

Robust MPC techniques are implemented to make MPC-based algorithms perform well even under adversaries like communication errors, localization uncertainties, model mismatch, and inherent lower level control behavior. A min-max

MPC approach has been used to make a centralized controller robust to uncertainties in MDV behavior [87] whereas Chen et al. use min-max MPC to create a centralized controller robust to communication and actuator delays [73]. Authors Gao et al. in [138] propose the use of a robust controller to counter both parameter uncertainties and uniform communication delay. The use of a buffer to mitigate the impact of communication failures has been recently proposed for a decentralized controller [133]. In short, papers [73, 87, 133, 138] propose robust algorithms to counter path planning uncertainties, communication uncertainties, actuator delays and vehicle model uncertainties like the uncertain range of frequency responses from torque command respectively.

In this work, we focus on a mixed vehicle coordinated braking scenario in the presence of different kinds of errors. The centralized control problem can be represented as follows:

Objective: **Safe stop of vehicles**:
Given:

- a road network structure including number of lanes
- the number of vehicles and their states
- classification of vehicles and the level of automation (MDV or CACC)
- vehicle characteristics like maximum braking capacity, velocity, position of the vehicles, etc.
- location of the obstacle or intersection

To find: Control values for autonomous vehicles that will bring all vehicles to a halt without any collisions and optimize the objective function at the same time

Desired controller properties:

- Should be able to handle multiple vehicle data
- Should be able to predict controls of MDVs
- Should be able to optimize controls

To summarize, this chapter introduced related work in the domain of collision avoidance between vehicles primarily on a single lane but also covered different road junctions. Different kinds of errors impacting a centralized control system were introduced, and the manner in which fellow researchers have implemented them has been highlighted. The chapter closes with a short problem statement and the concept of resolving the safe stop algorithm.

Chapter 3

Analysis of Centralized Controller Operation

This chapter introduces different entities and their responsibilities in the centralized controller operation. Centralized controller constraints and objectives are derived based on the safe braking operation requirements introduced in Section 2.3. The mathematical representation of the centralized controller is then represented. The last part of the chapter introduces different errors impacting the centralized control operation, and the modeling of these errors.

3.1 Centralized Controller Operation

This section looks at ways of obtaining state parameters of different types of vehicles and obstacle detection. Based on the state parameters, constraints and objectives, the mathematical expression of the centralized controller is derived.

3.1.1 Obstacle detection and gathering vehicle information

There are multiple ways of gathering state information of MDVs. First: MDVs have onboard units (OBUs) which can be plugged into On-board diagnostics (OBD) port or the CAN bus to retrieve state parameters. Second: A CACC vehicle can use its sensors like radars and lidars to detect where the neighboring vehicles are. If those vehicles are not communicating, they can be assumed to be MDVs else, CACC enabled. In this way, state information of all vehicles can be obtained and transmitted to the centralized controller.

Next, we introduce two ways of detecting the scenario with a broken vehicle (obstacle) or an intersection. Case A - the leading vehicle is a CACC vehicle: (1) There is a broken vehicle on the road which is transmitting DENMs or the RSU at the intersection is transmitting messages to the CACC vehicle approaching the intersection. (2) the CACC vehicle uses its sensors or cameras. In this case, the probability of awareness depends on the probability of receiving a message or the probability of sensing, at a particular distance. This probability of awareness thus changes with distance and depends on factors like communication channel occupancy, atmospheric conditions, etc. When two vehicles are approaching one another, or a vehicle is approaching an obstacle, the distance reduces gradually over time. The probability that the communication succeeds when they are far is small. Case B - the leading vehicle is a MDV: There is a broken vehicle or an intersection which needs to be detected by a human driver. The probability of awareness, in this case, depends on the attentiveness and visibility of the driver to perceive and understand the scenario.

This work does not focus on how the information is extracted, it focuses on processing and using the information. We consider notification distance as the distance at which the first message is received (or the object is sensed) by the CACC vehicle, or the vehicle is detected by the MDV. A range of notification distances is used to cover different values of awareness at different distances to the obstacle or the intersection.

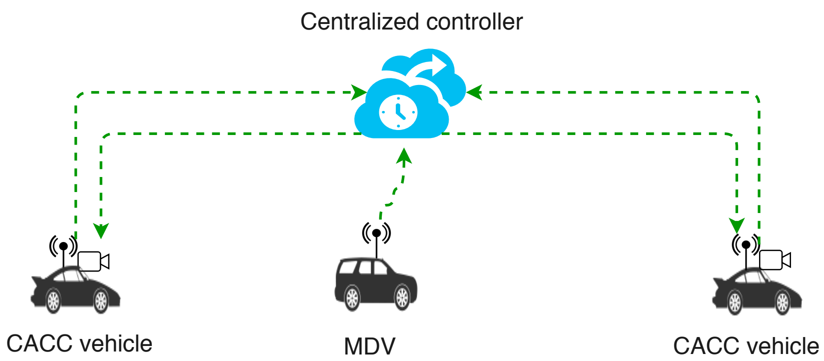


Figure 3.1: Centralized control operation

We consider a system of multiple vehicles traveling in the same direction on a single lane, as illustrated in Figure 3.1. On detecting or sensing an obstacle ahead if the first vehicle has to brake, the following vehicles would need to brake as well in a coordinated way to avoid collisions. There exists a centralized controller located on either one of the vehicles or a Road side unit or there exists a Cloud or an Edge Service, which helps coordinate vehicles to a safe stop. All vehicles (CACC and MDVs) are connected and transmit their noisy position, speed and acceleration estimates to the centralized controller.

The centralized controller receives the state information of the vehicles, processes it and generates controls for CACC vehicles. These controls are transmitted back to the CACC vehicles on the downlink. CACC vehicles are assumed to start implementing control action simultaneously on the reception of controls inputs from the centralized controller. On the other hand, we assume MDVs are human driven, they are without any control capabilities and react to the behavior of the vehicle in front.

3.1.2 Model predictive control basic principle

The general concept of MPC is explained in this subsection. The goal is to make sure the plant output follows the desired reference. The control input to reach the desired reference is computed by predicting the future of the plant based on the plant model over the prediction horizon. The MPC controller (optimizer) chooses that sequence of control inputs (over the control horizon) which minimizes the difference between the predicted behavior and the reference behavior. MPC has the ability to consider different constraints while finding the optimal solution. MPC is also robust to uncertainties because of the concept of recomputation of controls at every time slot.

Similarly, in the scenario we focus on, the MPC controller needs to compute control inputs such that the vehicles come to a halt before reaching an intersection. The above introduced problem formulation requires control computation for CACC vehicles while satisfying various constraints. Moreover, control predictions of other (MDV) vehicles generated using models, need to be considered while computing controls of CACC vehicles. This means controls are computed at the current time slot while keeping future controls into account. The first value of control from the computed controls is applied to the system. At the next time slot, fresh data is received, and optimization problem is resolved in a receding horizon fashion and controls (acceleration) are implemented on the CACC vehicles. Due to the receding horizon methodology, the MPC controller can be seen as a closed-loop technology as it computes controls using real-time feedback. The MPC controller thus has the ability to deal with the uncertainty of the real world, caused by unpredictable disturbances, and mismatch errors of the prediction model.

Real-time computation and implementation is a challenge to centralized MPC implementation as the optimization speed depends on the number of variables, the number of vehicles, the horizons, etc. More the number of vehicles which need to be coordinated, lower is the optimization speed. Decentralized control structures can be used to overcome the issue of real-time feasibility of the MPC controller.

3.1.3 Mathematical formulation of the MPC Controller

Horizons

The centralized controller uses different horizons, namely: prediction horizon, control horizon, and simulation horizon.

- Prediction horizon (N_p): This defines the horizon over which predictions are generated about controls of other entities by the controller. In our scenario, MDVs' controls are predicted for a certain number of time slots. This duration over which the predictions are generated is known as prediction horizon.
- Control horizon (N_c): This defines the horizon over which the ego-vehicle can be controlled, beyond which, the control implemented is usually assumed to be constant. The length of the control horizon can be equal to or lower than the prediction horizon. In our scenario, the centralized controller uses the predictions of controls of MDVs to generate controls for CACC vehicles. As the length of the control of MDVs is equal to the prediction horizon, the length of the control horizon is set to the prediction horizon as well. Let $N_c = N_p = N$. Note that the control horizon defines the number of values which need to be determined, longer the control horizon, more are the unknowns and longer the simulation takes.

- Simulation horizon (N_s): This defines the duration over which the simulation should continue. A finite simulation horizon would signify that the simulation must end within the defined duration. Infinite simulation horizon means the simulation would end when a certain value is reached, or a condition is triggered. In this work, the simulation horizon is kept infinite, but the MPC computation stops as soon as all CACC vehicles come to a halt.

It is important to ensure that appropriate values of horizons are chosen to ensure the feasibility of the optimization problem [145]. If it is too large, it increases computational time and memory requirements, if it is too small, it might render the optimization problem infeasible. The relation between computational time and time horizon has been studied in [146].

Vehicle Dynamics

This work only focuses on longitudinal control of vehicle and discrete time representation is used. The state vector of vehicle i ($i \in 1...n_v$) is defined as the position p_i and speed v_i tuple (3.1).

$$x_i = [p_i \; v_i]^T \tag{3.1}$$

Acceleration u is the input to the vehicle, u is a second derivative of position of the vehicle and thus the model we use is called a double integrator model [95]. The general kinematic relation between position, velocity, acceleration and jerk is given by (3.2). We assume acceleration to remain constant between two time slots.

$$\begin{aligned} u_i(n+1) &= \Delta u_i(n+1) + u_i(n) \\ v_i(n+1) &= v_i(n) + u_i(n)\Delta t \\ p_i(n+1) &= p_i(n) + v_i(n)\Delta t + 0.5 u_i(n)(\Delta t)^2 \end{aligned} \tag{3.2}$$

The above equations can be represented as a discrete time linear control system represented by (3.3) where values for constants are given by (3.4):

$$x_i(n+1) = Ax_i(n) + Bu_i(n) \tag{3.3}$$

$$A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} (\Delta t)^2/2 \\ \Delta t \end{bmatrix} \tag{3.4}$$

where Δt is the time between two consecutive time slots n and $n+1$.

Vehicle constraints

Vehicle and road constraints in terms of minimum and maximum values of position, velocity, acceleration are accounted for in (3.5a), and (3.5b),

$$\begin{bmatrix} p_i^{\min} \\ v_i^{\min} \end{bmatrix} \le x_i(n) \le \begin{bmatrix} p_i^{\max} \\ v_i^{\max} \end{bmatrix} \tag{3.5a}$$

$$u_i^{\min} \le u_i(n) \le u_i^{\max} \tag{3.5b}$$

where $(\cdot)_i^{\min}$, $(\cdot)_i^{\max}$ corresponds to minimum and maximum value of that parameter for vehicle i. $u_i^{\min}$ and $u_i^{\max}$ stand for maximum braking and maximum acceleration capabilities of the vehicle. Assigning appropriate values of $p_i^{\min}$ and $p_i^{\max}$ could define the maneuverable region for the vehicle.

Jerk constraint

Restricting jerks Δu within certain acceptable bounds ensures smooth braking for CACC vehicles and is implemented using (3.6).

$$\Delta u_i^{\min} \leq \Delta u_i(n) \leq \Delta u_i^{\max} \tag{3.6}$$

Collision Avoidance constraint

Collision avoidance for CACC vehicles is achieved by ensuring the distance between vehicles is always positive:

- Front-end collision avoidance:

$$d_{i,k}(n) > 0 \qquad i \in Z,\ \ k = i - 1 \tag{3.7}$$

- Front and rear-end collision avoidance

$$\begin{aligned} d_{i,k}(n) > 0 \qquad & i \in Z,\ \ i > 1,\ \ k = i - 1 \\ d_{i,k}(n) > 0 \qquad & i \in Z,\ \ i < n_v,\ \ k = i + 1 \end{aligned} \tag{3.8}$$

Where Z is the set of all CACC vehicles amongst n_v vehicles, $0 \leq size\{Z\} \leq n_v$; Z^c is complement set of Z which is the set of all MDV and ACC vehicles. Z^c is a null set (Z^c=$\varnothing$) if there are only CACC vehicles.

Terminal constraints

If the control profile of MDV models the relation between a MDV and a CACC vehicle, the centralized controller could be able to avoid accidents on both, MDV and CACC vehicles, but this is out of the scope of this paper. Starting position and velocity, and terminal position and velocity can be represented as constants $p_i(0)$ and $v_i(0)$ and $p_i(N)$ and $v_i(N)$ indirectly defines the range of the vehicle and the path it needs to follow in a 1D scenario. (3.9) finally ensures the terminal velocity of all vehicles reach zero and this signifies a braking scenario.

$$v_i(N) = 0 \tag{3.9}$$

Cost function

Cost functions are objectives which need to be optimized. Different articles in the literature have used different cost functions based on their requirements. Cost function used by Wang et al. has four components, 1. safety related, 2. travel efficiency related 3. comfort related 4. fuel economy related to ensure a safe, comfortable ride [146]. Liu et al. use a cost function that minimizes the longitudinal and lateral distance to the destination of the vehicle while ensuring it the actual speed is close to the desired speed, and they penalize large control input to maximize the comfort of the passengers [147]. Similarly, a cost function that penalizes: deviation from desired velocity, deviation from the desired inter-vehicle distance and any control input has also been used [95]. Liu also mentions to ensure MPC is always solved, the objective function is set to a constant value; once it is solved, it can be solved further to get optimal control based on original objective function [148]. If MPC computation fails, collision avoidance constraint is removed [148].

In our work, the objective is to minimize discomfort. Discomfort is related to the change in acceleration. Taking ∞-norm gives huge importance to big values whereas taking one norm gives equal importance to all values. Taking a two-norm ensures large values are minimized but at the same time gives importance to other values. It is for this reason, we set the cost function to minimize the two-norm of change in acceleration. To present the cost function in a quadratic form, we use the square of the two-norm of change in control inputs between two time slots, which can be given as :

$$\sum_{i=1}^{n_v} \sum_{\eta=1}^{N_p} (u_i(\eta) - u_i(\eta - 1))^2 \tag{3.10}$$

Optimization problem

The safe brake algorithm which integrates the above introduced constraints along with the optimization (cost) function can be put together as follows:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n_v} \sum_{\eta=1}^{N_p} (u_i(\eta) - u_i(\eta - 1))^2 \\ & \text{subject to} \\ \text{Assumed MDV model, } (3.1),\ (3.3),\ (3.4),& \\ (3.5\text{a}),\ (3.5\text{b}),\ (3.6),\ (3.8),\ (3.9)& \end{aligned} \tag{3.11}$$

Type of optimization problem

Based on the requirements of the system, different types of optimization problems can be formed. They are categorized as follows:

1. An optimization problem with an objective function without any constraints is called an unconstrained optimization problem.
2. An optimization problem with constraints and without any objective function is called a feasibility problem.
3. An optimization problem with an objective function and a set of constraints are called constrained optimization problem.

In the above equation (3.11), the decision variable is u which represents the acceleration of the vehicle which is the control input which needs to be optimized. Equation (3.2) shows that p, v, and δu are linearly dependent on the decision variable u. The constraints listed in equations (3.5a) and (3.5b) both are linear constraints. The cost function, on the other hand, is a quadratic function of the decision variable. If the MPC formulation uses slack variables, the cost function is different from equation (3.10), but could still be a quadratic function. Thus this system of equations is a quadratic optimization with linear constraints (a constrained optimization problem). It is convex and can be solved with various types of solvers, a few of them are CVX [149, 150], CVXGEN [151] or QUADPROG in Matlab. There are computational issues with CVXGEN if the number of decision variables is beyond a 'reasonable' number. Although formatting the optimization problem and constraints in CVX and CVXGEN is easier compared to QUADPROG, QUADPROG was found to be the quickest.

3.2 Modeling Errors influencing the centralized controller

3.2.1 Model Mismatch

The centralized controller operation involves using the current and previous available state parameters of vehicles to predict future control behaviour of MDV. The predicted trajectory of MDVs is used to compute collision free trajectory of CACC vehicles. As the actual trajectory (actual behavior model) of the MDV can not be known, it can only be predicted. *Model mismatch* arises when the actual model is different from the predicted model (or the assumed model) which leads to two separate trajectories as shown in Figure 2.1.

Model mismatches can only be eliminated if the predicted and the actual model are the same. In real life, different drivers have different driving characteristics like reactiveness and alertness (considered in perception response time), magnitude of braking they can achieve, etc. It is for this reason that the predicted and actual driving models can never be exactly same in real life.

If there is no model mismatch (and no source of error) then the control inputs computed once should be valid and ensure a collision free maneuver. In a scenario where model mismatch exists, due to model mismatch induced error (refer to Figure 3.2), CACC controls are not valid for long. Implementation of controls derived using the predicted trajectory of MDVs on CACC vehicles can result into collisions. It is for this reason that model mismatch is characterized as a source of error.

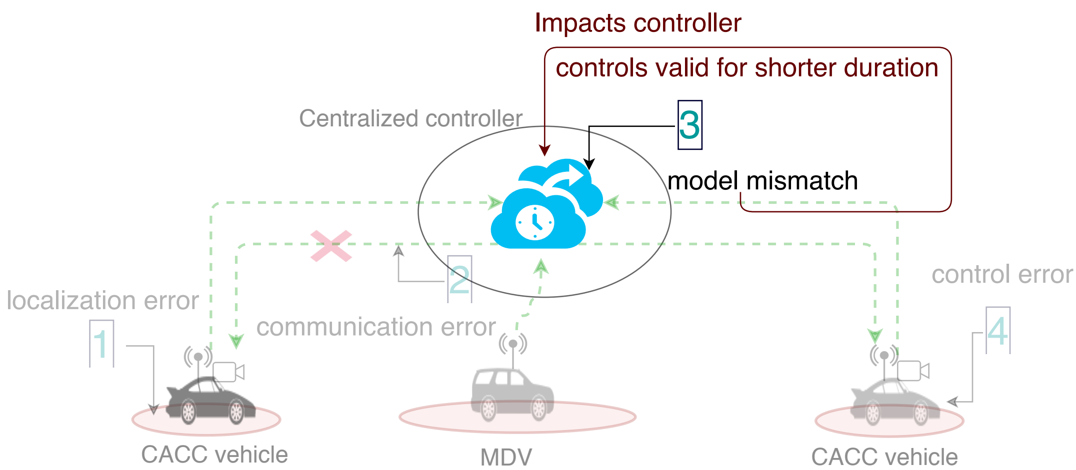


Figure 3.2: Centralized controller operation impacted by model mismatch

We implement model mismatch by using two totally different models for prediction of controls of MDVs and for actual control of MDVs.

Actual model of MDVs:

The actual behavior of MDV is based on 'follow-the-leader' strategy. In order to imitate actual human driving, MDVs are assumed to respond and react to the vehicle in front after a certain time delay represented by the perception response time. If there are multiple MDVs following one another in a stream of vehicles, the effective perception reaction time of a MDV would be proportional to the number of MDVs immediately ahead. This effective perception response time (now on referred to as perception response time) for vehicle i is denoted by $t_{i,1}$ (refer to section 2.2.5 for details). The behavior of MDVs after perception response time is actually governed by intelligent driver model (IDM) [62], i.e., a simplified version of Human Driving Model [71] where we assume the visibility to be limited to just the vehicle in front. We consider time slots of duration 100 ms, and introduce $c = 10$ slots/second. Control input is assumed to be the acceleration of the vehicle $u_i(n)$. The profile implemented by MDVs can be represented mathematically as follows:

$$u_i(n) = \begin{cases} 0 & n \leq c \cdot t_{i,1} \\ acc_i & n > c \cdot t_{i,1} \end{cases} \tag{3.12}$$

$$\begin{aligned} acc_i(n) &= u_i^{\max} \left(1 - \left(\frac{v_i}{v_0} \right)^{\delta} - \left(\frac{s^*(v_i, \Delta v_i)}{s_i} \right)^2 \right) \\ s^*(v_i, \Delta v_i) &= s_0 + v_i\, T + \frac{v_i\, \Delta v_i}{2\sqrt{u_i^{\max} \cdot b}} \\ s_i &= p_{i-1} - p_i - l_{i-1} \\ \Delta v_i &= v_i - v_{i-1} \end{aligned}$$

where i is the vehicle being considered, $i-1$ is the vehicle in front and so on; u_i, v_i, p_i, l_i is the acceleration, velocity, location and length of vehicle i; s_i is the actual distance between vehicles i and $i-1$; $u_i^{\max}$ is the maximum possible acceleration of the vehicle. b represents comfortable braking strength. The desired velocity and minimum distance between vehicles is denoted by v_0 and s_0 respectively. T represents the time headway observed by the vehicle; Δv_i is the difference in the velocities of vehicle i and vehicle $i-1$ in front; δ corresponds to a factor which can be tuned to control the behavior of the vehicle. Bigger the value of δ, more aggressive the reaction of the vehicle in general [33, 35].

Prediction model for MDVs

We introduce two models used to predict future controls of MDVs.

Model 1

This is the simplest control model where the MDVs are assumed to begin braking after an assumed perception response time $t_{i,1}$ after the vehicle in front starts to brake, and they continue to brake with a fixed magnitude $u_i^{\min}$ until halt (zero velocity)[1]. Thus, at any moment during the simulation, the future predicted controls of MDVs ($\forall\ i \in Z^c$) [2] can be derived using the following equation:

$$u_{i,\eta} = \begin{cases} 0 & \eta + n \leq ct_{i,1} \\ u_i^{\min} & v_{i,\eta+n} \geq 0, \eta + n > ct_{i,1} \\ 0 & \text{otherwise} \end{cases} \tag{3.13}$$

for all MDVs where $u_i^{\min}$ is the maximum braking capacity.

Model 2

Model 2 is more realistic than Model 1 and can be described as follows. As before, $t_{i,1}$ is the perception response time; $u_i(n)$ is the actual value of the applied acceleration; the jerk $\Delta u_i(n)$ is given by $\Delta u_i(n) = u_i(n) - u_i(n-1)$; $\Delta u_i^{\min}$ is the maximum permitted decrease in acceleration between two time slots.

Until perception response time $n \leq ct_{i,1}$, the controller assumes that the driver will start braking after a certain perception response time $t_{i,1}$ and increase the braking strength gradually until it reaches a maximum. At maximum braking strength, the vehicle continues to brake until halt. Hence, when $n \leq ct_{i,1}$, we set:

$$u_{i,\eta} = \begin{cases} 0 & \eta + n \leq ct_{i,1} \\ \max(\eta \Delta u_i^{\min}, u_i^{\min}) & v_{i,\eta+n} \geq 0, \eta + n > ct_{i,1} \\ 0 & \text{otherwise} \end{cases} \tag{3.14}$$

After perception response time, $n > ct_{i,1}$, we discern between three cases, based on the braking magnitude:

1. *If the braking magnitude is zero (i.e., $u_i(n) = 0$)*, the controller assumes that the driver will start braking at this time slot and continue to increase its braking strength gradually until the vehicle attains maximum braking strength. At maximum braking strength, the vehicle will continue to brake until halt:

$$u_{i,\eta} = \begin{cases} \max((\eta - ct_{i,1})\Delta u_i^{\min}, u_i^{\min}) & v_{i,\eta+n} \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.15}$$

2. *If the braking magnitude is increasing (i.e., $\Delta u_i(n) < 0$)*, the controller assumes that the vehicle will continue to increase its braking strength until the vehicle attains maximum braking strength. At maximum braking strength, the vehicle will continue to brake until halt:

$$u_{i,\eta} = \begin{cases} \max(u_i(n-1) + \eta c \Delta u_i(n), u_i^{\min}) & v_{i,\eta+n} \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.16}$$

[1]Please note that the value of perception reponse time used is the delay in reacting with respect to the first vehicle, also termed as effective perception response time in Section 2.2.5.

[2]Z is the set of all CACC vehicles out of n_v vehicles.

3. *If the braking magnitude is decreasing or constant (i.e.,* $\Delta u_i(n) \geq 0$*),* the controller assumes, the vehicle will continue to brake at the previous braking magnitude until halt:

$$u_{i,\eta} = \begin{cases} u_i(n-1) & v_{i,\eta+n} \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.17}$$

To summarize, we have introduced two simple braking control modes which the centralized controller uses to derive controls for CACC vehicles. The second model is more realistic compared to the first.

3.2.2 Control Error

This work considers two restrictions arising from the lower level controller or engine properties. First, the restrictions in the change in acceleration arising from either the engine capabilities or from the jerk tolerance of humans. Second, related to controller or engine properties like engine constant which would result into the generation of acceleration different from the desired acceleration value. Although these two are similar and related, they can be implemented in different manner.

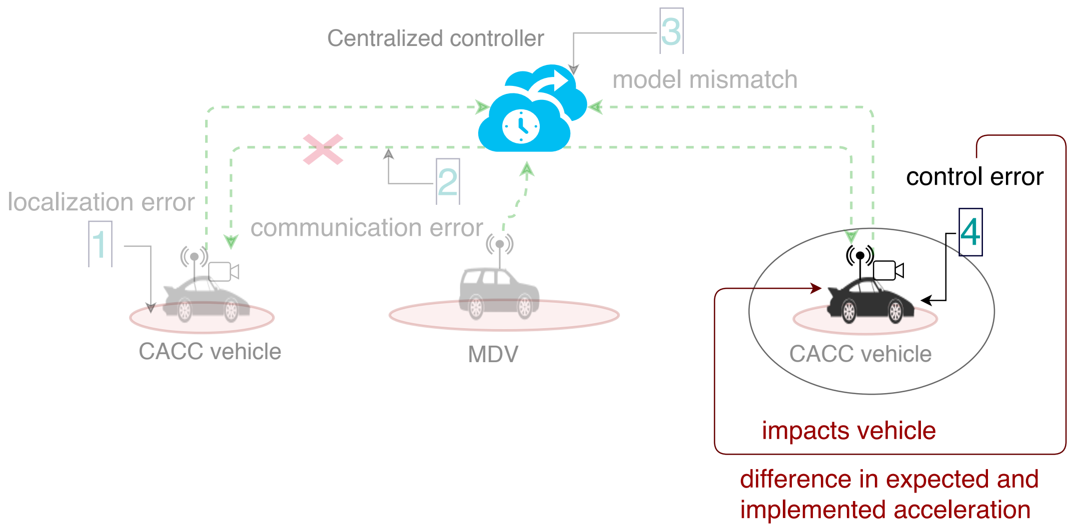


Figure 3.3: Centralized controller operation impacted by control error

Engine behavior and restrictions

To regularize the change in acceleration, the maximum and minimum jerk values can be bounded. As discomfort is directly related to the change in acceleration, bounding jerks ensures discomfort can be within reasonable limits. This can be implemented mathematically with the use of a jerk filter, introduced in (3.6).

When lower level controller profile is not considered, this jerk filter is important to restrict the change in acceleration. On the other hand, when lower level control profile is considered (as control error), the engine properties usually restrict the implemented acceleration to a value inferior to the desired value.

Implementation of Control errors

Control errors arise because of the inherent engine properties which act like a low pass filter which result into a difference between the desired and actual acceleration. This results into different actual and desired (or assumed) state parameters like position, velocity and acceleration of the vehicle (Refer Figure 3.3). But due to the availability of onboard sensors, state parameters can be measured and this information can be used in the feedback while generating controls at the next time slot.

In this work, where explicitly mentioned, we use the modeling introduced by Michele Segata in PLEXE [131]. The actuation lag introduced by power-train dynamics is modeled as a first order low-pass filter and implemented it as follows:

$$u^*(n) = \beta u(n) + (1-\beta)u^*(n-1); \tag{3.18}$$

where

$$\beta = \frac{\Delta t}{\tau + \Delta t} \tag{3.19}$$

$u^*(n)$ represents the applied (effective) acceleration whereas $u(n)$ represents the desired acceleration. τ is the time constant of the engine, Δt is the sampling frequency (update step) in seconds.

3.2.3 Localization Error

Localization error is one of the different errors impacting a centralized controller operation. As localization error would depend on the technology used, the magnitude of localization error can be vastly different. Different magnitudes of localization error will have different level of impact on the centralized controller. Controls computed using perceived location, which is different from the actual localization of the vehicle results into computation of faulty controls, which can result into collisions (refer Figure 3.4). The goal is to ensure vehicles are coordinated such that collision avoidance is ensured despite the presence of localization errors.

Modeling Localization Error

We differentiate true position p_i and the perceived position (erroneous localization) p_i^* of a vehicle i (refer to Figure 3.5), where the former is the actual position of the vehicle where as the latter is the calculated position of the vehicle. Neither the transmitting vehicle nor the receiver would know the true positions. Perceived position is different from the true position when there is an error in localization e_i. As localization error is generally available in 2D, perceived position can be represented as:

$$p_{i,x}^* = p_{i,x} + e_{i,x} \qquad p_{i,y}^* = p_{i,y} + e_{i,y} \qquad p_i = [p_{i,x}, p_{i,y}] \qquad p_i^* = [p_{i,x}^*, p_{i,y}^*] \tag{3.20}$$

where suffix i, x and i, y correspond to the parameter value in longitudinal and lateral direction of the vehicle i. Error e_i has been split into longitudinal error $e_{i,x}$

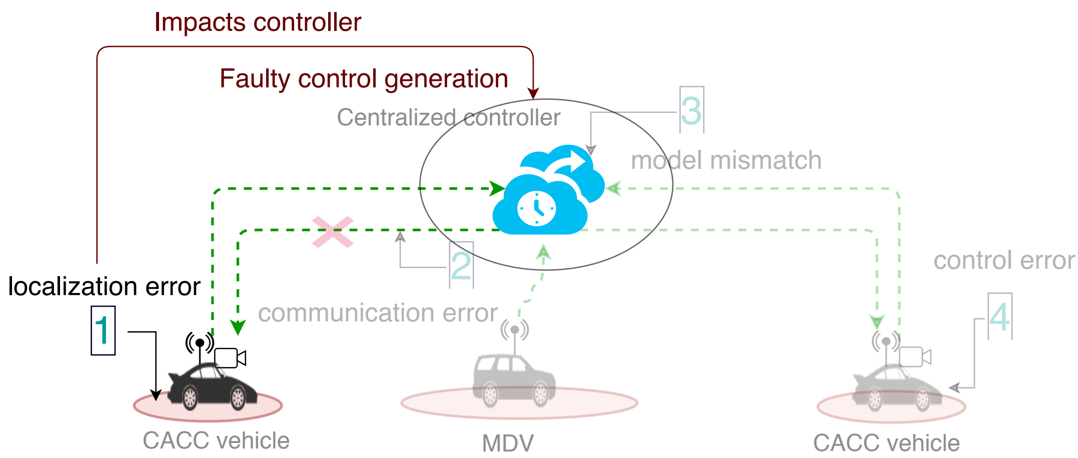


Figure 3.4: Centralized controller operation impacted by localization error

and lateral error $e_{i,y}$, which are related as:

$$e_i = \sqrt{e_{i,x}^2 + e_{i,y}^2} \tag{3.21}$$

Consider Figure 3.5, where the object in green is ego vehicle's true location. The perceived location of the vehicle is shown in blue, which is at a distance equal to the localization error magnitude. The above introduced formula is used to generate perceived localization values to be used in simulations. The proposed algorithm to make the centralized controller robust to localization errors is introduced in section 4.6.

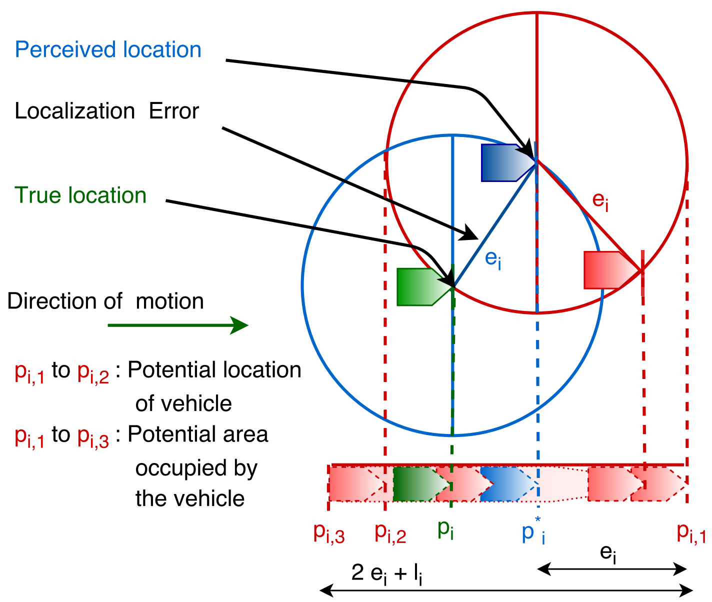


Figure 3.5: Modeling localization errors

3.2.4 Communication Error

Modeling communication error

As introduced in Chapter 2, in a futuristic C-ITS scenario, vehicles will be constantly communicating not only with one another but also with other entities like RSU or cloud servers. The concept of receding horizon MPC involves recomputation of controls at every time slot. Communication of state parameters to the centralized controller on the uplink and the communication of computed controls back to the vehicles on the downlink form an integral part of the proposed control system.

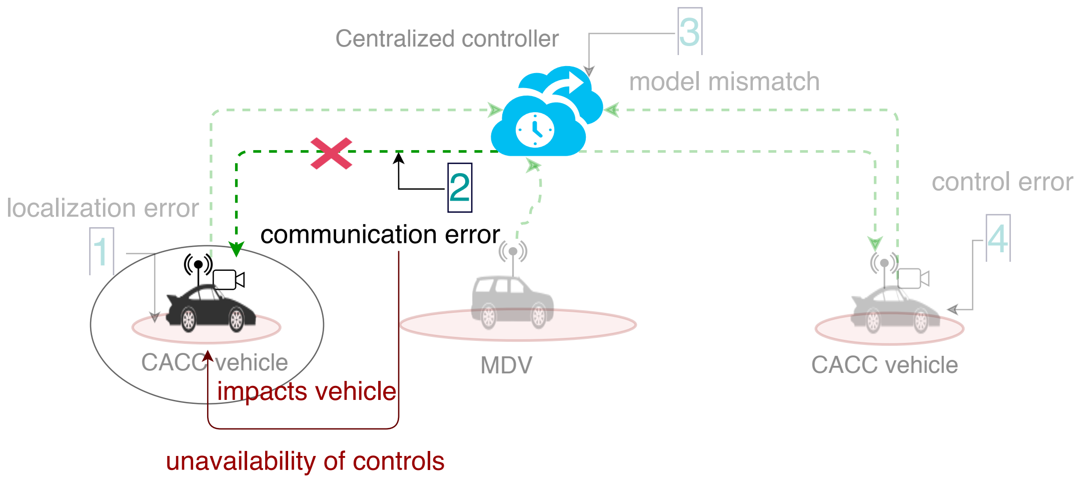


Figure 3.6: Centralized controller operation impacted by communication error

Communication in real life scenarios are subject to impairments leading to packet losses and errors can manifest either on the downlink or on the uplink. These communication errors can result into delayed reception, out of order reception of packets, packet losses, etc. Uplink transmissions are assumed to be perfect. Communication failure on the downlink can cause packet losses and nonavailability of controls at the CACC vehicle. The impact of packet losses on the downlink is evaluated and addressed in this work (refer Figure 3.6). These packet losses can be modeled as either independent occurrences (Bernoulli model) or as dependent occurrences (Markov model). We explain in brief each of them below.

Bernoulli model

Successful packet reception and decoding depends on the received signal to noise plus interference ratio (SINR), which is influenced by fading. Considering independent and identically distributed (IID) packet errors and white Gaussian fading, successive Bernoulli trials can be performed to obtain an analytical packet reception model.

Bernoulli distribution is discrete time probability distribution of a random variable. There are only two possible outcomes either $r = 0$ represents that a packet is not received with the probability p, or, $r = 1$ which represents the packet is received with the probability 1-p. Mathematically, it can be expressed as:

$$P(r) = \begin{cases} p, & \text{if } r = 0 \\ 1 - p, & \text{if } r = 1 \end{cases} \tag{3.22}$$

Bernoulli's distribution introduced in (3.22) has been often used in the literature with with $p = 1 - p = 1/2$. [86] uses Bernoulli's random variable is used to depict whether a packet has been received or not on the uplink by the centralized controller. In a decentralized control scenario, [152, 153] uses Bernoulli model to generate random packet reception status. Packet loss assuming independent packet reception probability has been used in the literature to compute the probability of receiving atleast one packet by the receiver by the time distance reduces to the minimum distance to avoid a collision [154].

Markov model

In practice, as illustrated by Gudmunson [155], fading is strongly correlated, leading to successive packet losses/reception. This results in a burst error channel that cannot be captured by a Bernoulli model due to non-IID losses. A two-state Markov model has thus been proposed to capture such 'shot-noise' model. This approach has notably been described in [156, 157] to model the packet reception probability during periodic transmissions influenced by burst errors.

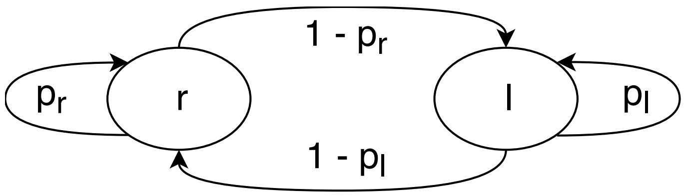


Figure 3.7: Markov chain used for modelling burst errors

We illustrate in Figure 3.7 such a discrete time stochastic first order two state Markov process. The probability of successfully receiving the next packet depends on the current link state. It is p_r, with $0 < p_r < 1$, when the link is in the reception state ($s = [1\ 0]^T$), and is $1 - p_l$, with $0 < p_l < p_r$, when the link is in the loss state ($s = [0\ 1]^T$) [156], where s is the communication state vector which is either in the reception state r or the loss state l. T represents the transition matrix between two consecutive states. We assume the transition matrix remains constant over the duration of the simulation (few tens of seconds). As long as the vehicles are within the communication range, the state vector at $(n + k)^{th}$ time slot can be given as:

$$s_{n+k} = s_n \cdot T^k \tag{3.23}$$

where

$$s = \begin{bmatrix} r \\ l \end{bmatrix} \quad T = \begin{bmatrix} p_r & 1 - p_r \\ 1 - p_l & p_l \end{bmatrix} \tag{3.24}$$

To summarize, the first part of this chapter introduced and explained the centralized controller operation. The reasoning and implementation of the MPC controller is explained and the mathematical formulation is introduced next. Different errors are modeled mathematically and the impact of these errors on the centralized controller is highlighted. These mathematical models shall be used in the next chapter to evaluate the performance of the MPC controller under different errors.

Chapter 4

Robust Centralized Controller

This chapter highlights various techniques introduced to ensure the controller is robust to different kind of errors. Different controller parameters result into different results. Depending on the scenario to be simulated, parameter values need to be chosen. Simulation parameters and controller configuration are introduced in this chapter. The implementation of a buffer to counter uncertainties is then explained. Next, simulation results corresponding to model mismatches and different ways of modeling control errors are evaluated. The experiment performed with the driving simulator to extract controls implemented by a human driver in a mixed vehicle scenario is explained and results are discussed. The chapter then introduces different fall back techniques to counter communication errors. The impact of localization error on the controller and the proposed method to counter localization error is later explained. Finally, a controller robust to all kinds of errors is proposed and compared to a non-robust controller. The final section of this chapter discusses parameter sensitivity.

4.1 Centralized controller configuration

While configuring the centralized controller, different things need to be considered, like: is it essential for the system to reach a specific state or value? Does the time taken to reach this value matter? Does the energy required to reach this state matter? Such questions help design the cost function. And the next question is about the horizon also known as the time span of the system operation during which one is concerned about such defined performance measures. If the end state must be reached within a finite time, the optimization problem is called a finite time horizon problem, but if time is not a constraint, it is called an infinite horizon problem. We answer each of those questions one by one to identify and develop the optimization problem.

For the problem the thesis targets, the terminal state is important, which is (near) zero velocity[1]. The time taken to reach this state does not matter. The energy required does matter, in the sense that, fluctuations in control inputs causes

[1]If the terminal velocity is put as a slack variable, the velocity of the vehicle can be slightly greater than zero.

discomfort. This means it is an infinite horizon constrained optimization problem.

Other system parameters which influence simulations are introduced next:

- notification distance: the distance at which the leading vehicle detects an obstacle in the front, or is notified about a potential obstacle via V2V/V2X communications (refer to section 3.1.1 for details)
- initial vehicle state parameters: like position, velocity, acceleration of different vehicles
- different MDV models: used for generating control predictions and for actual control of MDVs introduced in section 3.2.1

Key configurable options

- two end collision avoidance: CACC vehicles try to avoid collisions with vehicles on both ends as opposed to only avoiding collisions with the vehicle in front
- finite horizon control: a finite horizon is chosen within which vehicles must come to a halt
- infinite horizon control: vehicles have an infinite time horizon to come to a halt
- slack on terminal velocity: the terminal velocity of the vehicle can be allowed to be $0 \pm \epsilon$ ($\epsilon \geq 0$). Any value of ϵ is heavily penalized. This helps avoid infeasibility issues due to the granularity of the system and the fact that velocity while braking can not go below zero. The use of slack changes the cost function (3.10) to:

$$\sum_{i=1}^{n_v} \sum_{\eta=1}^{N_p} (u_i(\eta) - u_i(\eta - 1))^2 + \rho\, \epsilon \tag{4.1}$$

  where ρ is the heavy penalty factor on any positive ϵ value. Moreover, the maximum value of terminal velocity can also be bounded by assigning a non-negative value ζ:

$$v_i(N_p) = \epsilon; \quad 0 \leq \epsilon \leq \zeta \tag{4.2}$$

- terminal acceleration condition: terminal acceleration of the CACC vehicle must reach zero when the velocity of the vehicle reaches zero to ensure a smooth brake.[2]
- different assumed and effective perception response time: different values of assumed and actual perception response time are used to simulate the difference in the presumed and actual driving behaviors

[2]On the contrary, if the acceleration of the vehicle the instant before the vehicle reaches zero velocity is more than the jerk value, it would mean, the discomfort faced would be high when the velocity reaches zero (and the acceleration would directly go to zero).

- the ability to differentiate ACC and MDVs: In case ACC vehicles are present as well, the developed model has the option where the centralized controller can either have the knowledge that MDVs and ACC vehicles are different, or ACC vehicles can be treated as MDVs.

- communication error: the controller either faces communication issues or has perfect communication

- localization error: the controller either faces localization issues or has perfect localization

- control error: the controller assumes the implemented control and the applied control to be the same (no control error) or different (accounts for the lower level controller behavior)

4.1.1 Simulation parameters

We assume at the start of the simulation, four vehicles are approaching an intersection/obstacle. The first vehicle is notified of a potential obstacle or an intersection at the *notification distance* from the obstacle. Notification distance is also the distance at which the first vehicle senses an obstacle or is notified about an obstacle. A range of *notification distances* from 90 m to 150 m are used for simulations. Once the first vehicle is notified, a centralized controller intervenes and assists a collision-free braking procedure. MDV and CACC vehicles assume corresponding braking models: CACC vehicles implement controls received from the controller whereas MDVs react to the vehicle in front. All vehicles are assumed to have a zero initial acceleration.

An equal number of CACC and MDV vehicles (2 each) are used. There are six possible arrangements of these four vehicles, and 20 samples of each arrangement are taken to create a database of 120 simulation samples. We define a simulation *sample* as the set of vehicle state parameters. Each sample has different initial velocities (90 kmph $\pm$ 5 %) and initial time headway of 1.2 s and 3 m of safe distance (distance with the vehicle in front at zero velocity). We assume vehicles can only have non-negative velocity signifying, they can not move in reverse. The obstacle or the intersection is assumed to be at the origin.

The perception response time of a MDV is drawn from a normal distribution $\mathcal{N}(1.33, (0.27)^2)$ [141] and is capped between 0.8 s and 1.8 s. Vehicles can not reach an infinite braking capacity, the maximum braking capacity $u^{\min}$ in the simulation is set to $-5.928\,\mathrm{m/s^2}$ ($-0.6\,\mathrm{g}$) [142] which is the mean value of maximum braking capacity of cars. Other key simulation parameters are mentioned in the Table 4.1. The following values are used for the IDM (introduced in Eq (2.1)), which are used to generate the actual values of control for MDV. $v_0 = 25$ m/s ; $s_0 = 3$; $T = 1.2$; $a = 1$; $\delta = 4$.

Table 4.1: General simulation parameters

Parameter	Value	Parameter	Value
g	$9.88\,\mathrm{m/s^2}$	$(\Delta u^{\min}, \Delta u^{\max})$	$(-0.25, +0.25)$
l_i	$4\,\mathrm{m}$	δ	4
Δt	$0.1\,\mathrm{s}$	b	$-2\,\mathrm{m/s^2}$
$u^{\min}$	$-5.928\,\mathrm{m/s^2}$	N_p, N_c	100
$u^{\max}$	$1\,\mathrm{m/s^2}$	$(v^{\min}, v^{\max})$	$(0, \infty)$
$(p^{\min}, p^{\max})$	$(0, \infty)$		

A simulation is counted to be successful when the terminal velocity of all vehicles reach (almost) zero and is considered a failure if there are any collisions. Slack on terminal velocity is used. ζ is the maximum permitted velocity at the end of the simulation, it is set to $0.01\,\mathrm{m/s}$ and the penalty factor ρ is set to 10^6. Simulations are evaluated using two metrics: (i) *collision avoidance (CA)* during braking. (ii) *discomfort* (two-norm of change in acceleration per time slot of the CACC vehicles.)[3] Where ever applicable and necessary, the average *Packet Loss Ratio (PLR)* over all simulation runs are also mentioned.

4.2 Buffer

The idea of using controls generated using the predictive control formulation if a certain number of packets are lost consecutively after a successful reception has been proposed in [91]. It mentions that when predictive control is used, in case of multiple packet loss, the implementation of control inputs from the last received packet is better for control system compared to not changing the control input. It was then implemented to counter communication errors by [133]. [94] did not use a buffer but implemented emergency controls in case of lack of control information due to delay or communication loss. In this work, we use the concept of a buffer as well.

The output of the optimization problem is a vector of control actions for each CACC vehicle i, over a *prediction horizon* N_p:

$$u_i(n) = [u_i(n|n)\ u_i(n+1|n)\ u_i(n+2|n)...\ u_i(n+N_p-1|n)] \tag{4.3}$$

where n is the current time slot. $u_i(n)$ represents the controls generated by the centralized controller for vehicle i which has N_p values. $u_i(n|n)$ is the control value for time slot n, computed at n. $u_i(n+1|n)$ is the control value for time slot $n+1$, computed at n and so on. Control vectors corresponding to each CACC vehicle is sent in downlink; i.e.: $u_{i-1}(n)$ is communicated to vehicle $i-1$ and so on. As soon as control inputs are received, the control values denoted in (4.3) are stored in a buffer. The buffer is updated at each reception of a new packet.

[3]Computed only if the simulation terminates without any collisions.

At specific time slots either due to the uncertainty in localization or model mismatch or the robust-modeling of localization errors or violation of the collision avoidance constraint, the feasible set of the optimization problem could be empty and the control problem might be infeasible. When a computation is infeasible, no control data is transmitted in the downlink. Alternately, when there is either a communication delay or a packet loss, no packet is received at the destination vehicle. In such a scenario, control data corresponding to that time slot from the previous computation stored in the buffer is used.

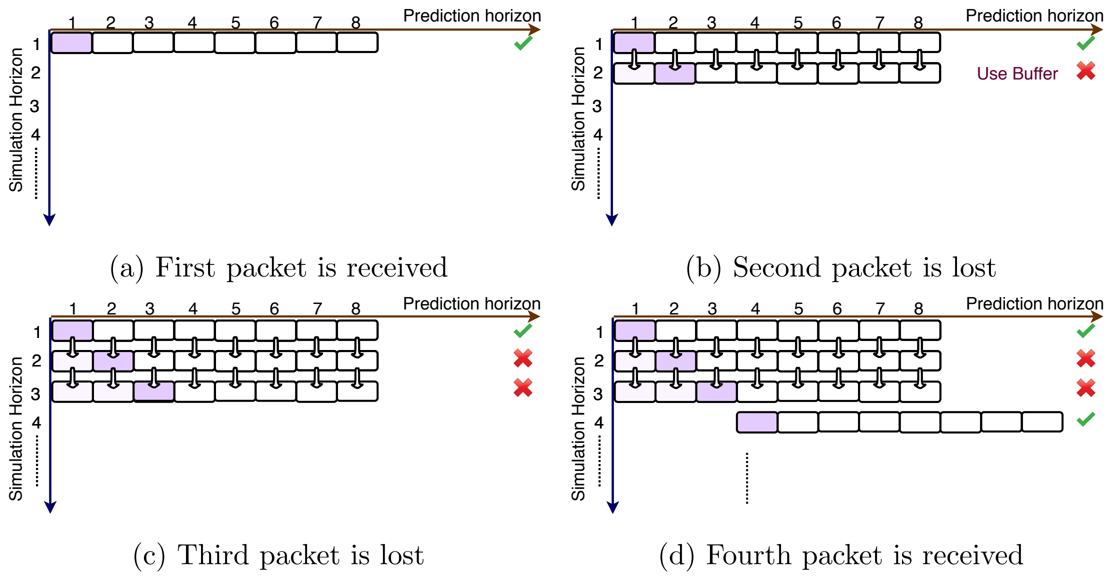


Figure 4.1: Explanation of implementation of the buffer

The implementation of the buffer is illustrated in Figure 4.1. Assume during the first time slot, controls $u_i(k)$ with N_p values are successfully computed at the controller and received by the vehicle. The received control information corresponding to that vehicle is loaded onto the buffer, and the first control value $(u_i(k|k))$ is implemented (shaded purple) as shown in Figure 4.1a. At the next simulation slot, assume new controls are not received; the buffer content is retained and the next control value from the buffer $u_i(k+1|k)$ is used by the vehicle as shown in Figure 4.1b. At the third time slot, assume new controls are not received, the buffer is retained, the next control value $u_i(k+2|k)$ is applied as illustrated in Figure 4.1c. At the fourth time slot, assume new data is received, the freshly received control data $u_i(k+3)$ is loaded in the buffer and the first value $u_i(k+3|k+3)$ is applied as shown in Figure 4.1d.

If the optimization problem is infeasible at the beginning, the algorithm ignores the jerk filter to allow any limit of jerk for the first time slot (within maximum braking capacity). If even then, the optimization problem is infeasible, CACC vehicles brake at the maximum braking capacity permitted by the jerk filter (e.g.: If permitted jerk is $2.5\,\mathrm{m/s^3}$ and the acceleration in the previous time slot was zero, the permitted braking strength for the current time slot would be $-0.25\,\mathrm{m/s^2}$). The optimization problem is retried at the next time slot with updated parameter values.

4.3 Robustness to Model Mismatch

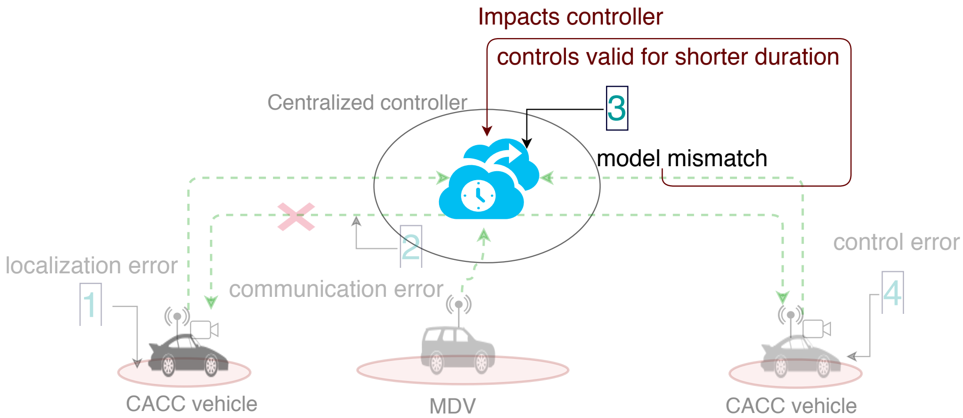


Figure 4.2: Model mismatch impacting centralized controller

In this particular study, the goal is to highlight the impact of model mismatch on a centralized controller in the absence of localization and communication and control errors. Different assumed (prediction) models give rise to different model mismatches. Two kinds of model mismatches are generated: first by using the MDV profile 1 represented by equation (3.13) in Section 3.2.1 as the assumed MDV model whereas the profile represented by equation (3.12) as the actual MDV model. This set of simulation is represented by (4.4). The second kind of model mismatch is generated by using the MDV profile 2 introduced using equations (3.14) (3.15) (3.16) (3.17) as the assumed MDV model whereas the profile represented by equation (3.12) in Section 3.2.1 as the actual MDV model. This set of simulation is represented by (4.5). We propose the use of a control *buffer* introduced in Section 4.2.

At each time slot, the updated state parameters (actual trajectory) need to be used to regenerate predictions to recompute controls to mitigate model mismatch. In other words, the error introduced by *model mismatch* would keep accumulating and compounding until it is recomputed. Thus if predicted control values computed at any particular time slot is used in time slots other than the one in which it was computed it might result in collisions. Thus re-computations of controls is necessary which is the basis of a MPC system robust to model mismatch.

Non-robust controller is the one where controls are computed only once (open-loop optimization). As control computations are only computed once, the computed controls are saved (the use of a buffer) and these controls are used until the vehicles come to a halt. Compared to the non-robust controller, controls are recomputed at every time slot for a controller **robust** to model mismatch.

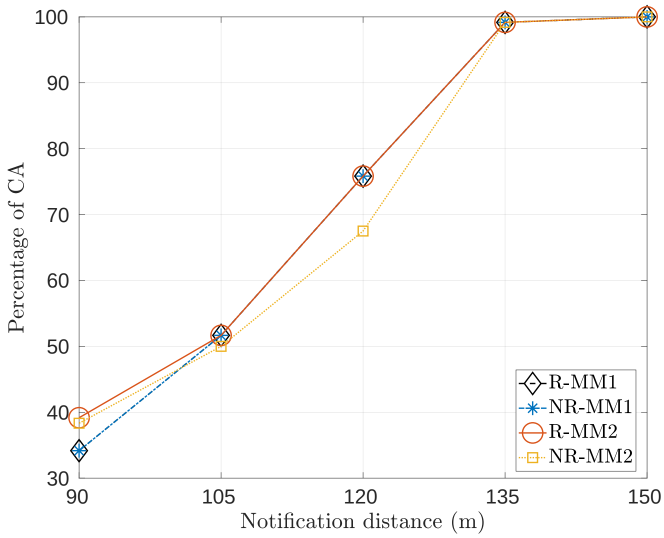


Figure 4.3: Collision avoidance under different types of model mismatch

$$\text{minimize} \quad \sum_{i=1}^{n_v} \sum_{\eta=1}^{N_p} (u_i(\eta) - u_i(\eta-1))^2 + \rho\,\epsilon \tag{4.4}$$

$$\text{subject to}$$
$$(3.1),\ (3.3),\ (3.4),\ (3.5a),\ (3.5b),\ (3.6),\ (3.8),\ (4.2),$$
$$\text{Assumed MDV model: } (3.13)$$

For both, robust and non-robust controller, two types of model mismatch are generated. We perform simulations for different values of notification distance and evaluate the impact of model mismatch on the system. Simulation results are plotted in the Figure 4.3.

$$\text{minimize} \quad \sum_{i=1}^{n_v} \sum_{\eta=1}^{N_p} (u_i(\eta) - u_i(\eta-1))^2 + \rho\,\epsilon \tag{4.5}$$

$$\text{subject to}$$
$$(3.1),\ (3.3),\ (3.4),\ (3.5a),\ (3.5b),\ (3.6),\ (3.8),\ (4.2),$$
$$\text{Assumed MDV model: } (3.14)\ (3.15)\ (3.16)\ (3.17)$$

Simulation results

The robust controller is represented using ‘R’, Non-robust controller is represented using ‘NR’ and Model mismatch type 1 is represented using ‘MM1’ and Model mismatch type 2 is represented using ‘MM2’. As there is no source of error other than the model mismatch, we can see that the performance of the *robust* controller is always better than the *non-robust* controller, although only marginally, in terms of collision avoidance as seen in Figure 4.3. It also shows that the percentage of collisions avoided under different model mismatch scenarios could be different. The difference is because different assumed MDV profiles (different types of model mismatch) lead to different constraints which lead to the computation of different values of controls and this could lead to a different number of total collisions avoided as seen for notification distance of 90 m in Figure 4.3.

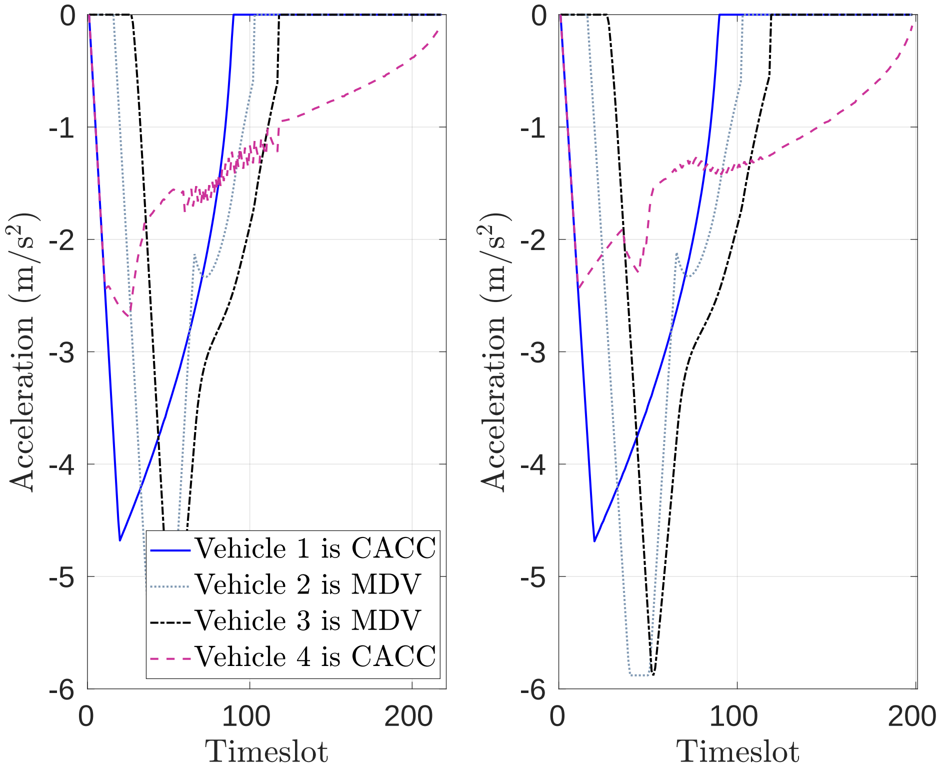


Figure 4.4: Controller performance in under different types of model mismatch

Figure 4.4 shows the difference in controls with the use of different prediction models. The subfigure on the left corresponds to the acceleration control using model mismatch type 1 whereas the one on the right is the acceleration control using the model mismatch type 2. The control of the first vehicle (CACC) is constrained by the motion of the vehicle behind and the obstacle in front. Vehicle 2 and 3 are MDVs which have the same control profile in both cases as the controls of the first vehicle is the same. But the last vehicle (vehicle 4, CACC vehicle) only needs to avoid collision with the vehicle in the front. The control prediction dictates the controls of this vehicle. It is clear that the acceleration plots of vehicle 4 are different, which is because different models used to realize a model mismatch. Moreover, this also results in different discomfort which is evaluated using Figure 4.5 later.

As the number of collisions avoided by the robust and the non-robust controller is the same, we look at the discomfort measure during the braking procedure. Discomfort is computed as the two-norm of the change in acceleration per time slot. In general, the performance in terms of average discomfort of the robust controller

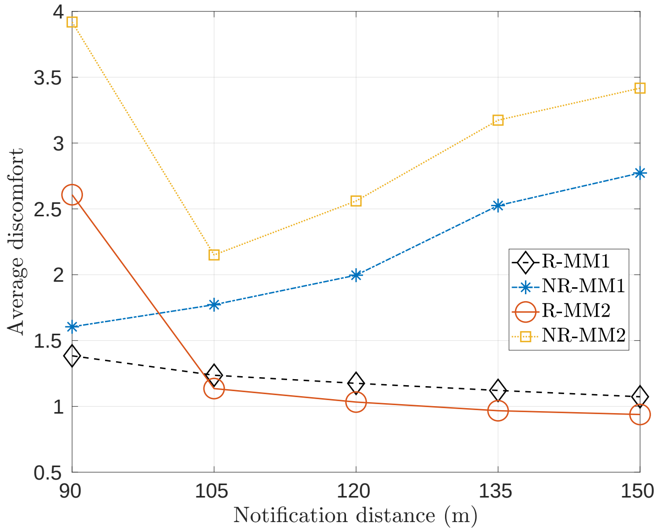


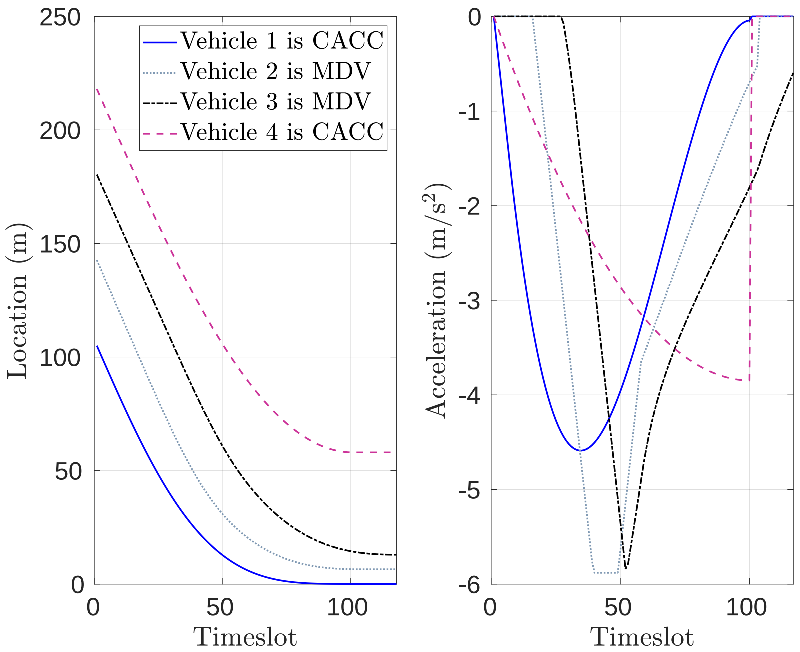
Figure 4.5: Discomfort evaluation for different types of model mismatch

is definitely better than that of the non-robust controller as seen in Figure 4.5. The discomfort of the robust controller with model mismatch type 2 is lower than the discomfort of the robust controller with model mismatch type 1 in general because the controls generated for CACC vehicles depend on the control profile of MDVs. If the MDV braking profile used for generating predictions is harsh (comprises of strong braking), other vehicles' computed controls would also usually have harsh braking.

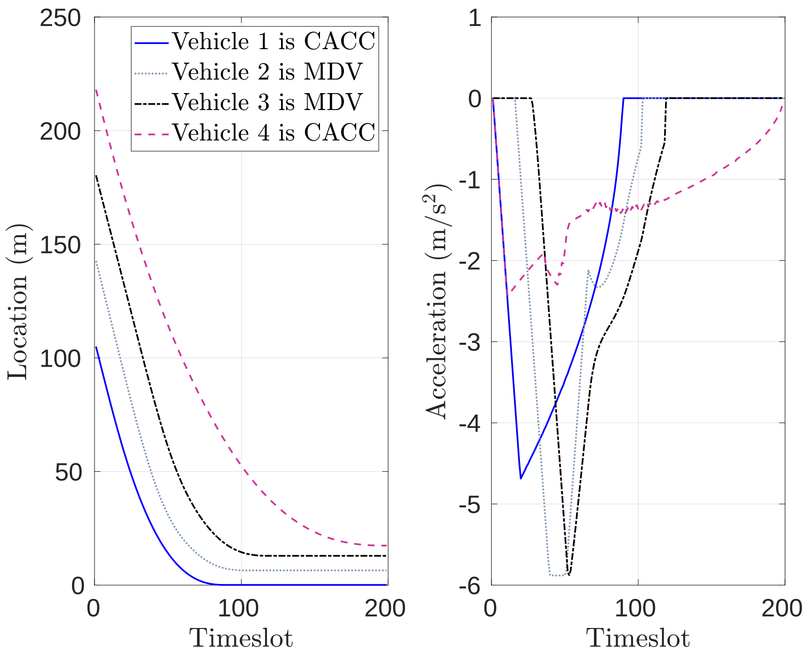
From figures 4.3 and 4.5 we can conclude that the robust controller with model mismatch type 2 can avoid more collisions at the notification distance is 90 m, at the cost of additional discomfort.

Figure 4.6a plots the localization and acceleration of all vehicles over the simulation duration, where controls for CACC vehicles is computed only once for a non-robust open-loop optimization. It is interesting to observe that the last CACC vehicle (vehicle number 4), stops before the 3rd vehicle which is MDV, and the distance between these vehicles when both of them come to a halt is approximately 40 meters, which is unusual driving behavior. This is because, CACC vehicle at position 4 has its controls computed once, and it implements those controls, whereas MDV at position 3, reacts to the control of vehicle 2. So, vehicle 2 comes to a halt after vehicle 1, vehicle 3 after vehicle 2 but vehicle 1 and vehicle 4 come to a halt simultaneously by the end of their prediction horizon. Controls for CACC vehicles are not recomputed here.

On the other hand, in Figure 4.6b plots the localization and acceleration of all vehicles of the same simulation assuming the implementation of MPC, where controls are computed at every time slot. When all vehicles come to a halt, vehicles are closely placed without collisions. This is because CACC vehicle controls are recomputed at every time slot based on the updated state parameters of other vehicles, which enables better control both in terms of comfort and collision avoidance. Moreover, unlike in Figure 4.6a, vehicles do not exhibit unusual behavior.



(a) Use of a non-robust controller



(b) Use of a robust controller

Figure 4.6: Robust and non-robust controller evaluation under model mismatch

If the results of the two types of model mismatch are considered with robust controller, model mismatch type 2 managed to avoid more collisions at smaller notification distances and overall had lower discomfort compared to model mismatch type 1. Now on, unless explicitly mentioned, model mismatch type 2 is implemented in all the simulations, i.e., (3.14)- (3.17) are used for generating control predictions for the MDV. Please note, for each notification distance, 120 simulations with four vehicles, 2 CACC, and 2 MDV are performed. Average jerk is computed per CACC vehicle only for simulations which end without any collisions.

4.3.1 Interface with Driving simulator

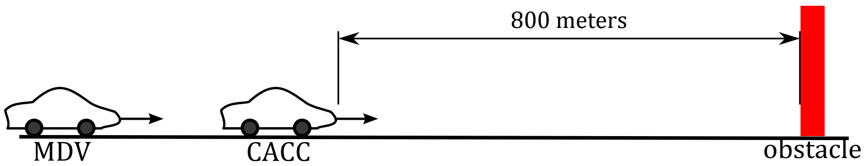
As previously summarized, model mismatch arises when MDV control predictions generated using the MDV prediction model are different from the actual MDV control. Matlab based simulations used different prediction models but kept the actual MDV model constant. Simulations are more realistic if a wider range of model mismatches are generated by using a variety of actual MDV control models.

The behavior of every human is different, the controls they would implement would thus be different as well. To have a wide range of actual MDV controls, humans can be asked to participate in experiments. Using a driving simulator, controls exerted by human drivers can be extracted and used as actual MDV controls in the experiments. Human drivers controlled a MDV based on the vehicle in front displayed on the screen of the driving simulator interface. Please refer to Figure 4.7. The controls which the human driver exerts were fetched from the driving simulator and were passed on to the centralized controller. The centralized controller used this as the actual control of the MDV rather than using the mathematical expression to generate the applied value of controls. The performed experiment is next explained.

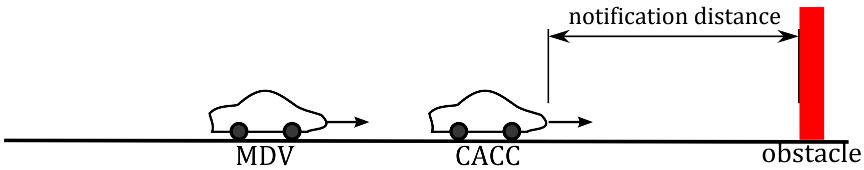
Figure 4.7: Model mismatch evaluation using a Driving Simulator

Although a generalized multi-vehicle braking system is formulated, in this experiment we consider a two vehicle braking scenario where a MDV is following a CACC vehicle. We assume the vehicles are on a single lane road, lane change is prohibited. Following are the parameters used with IDM (parameters follow standard notation as in [61]): v_0, s_0, T, a, δ, b are 25 m/s, 3 m, 1 s, 1 m/s^2, 4 and -2 m/s^2 respectively. Length of all vehicles l is set to 4 m. Initial distance between vehicles is 3 m. The frequency of control computation is defined by the controller's update frequency (set to 10 Hz; $\Delta t = 0.1s$). $u^{min} = 5.88$ m/s^2, Δu^{min}, Δu^{max} are set to -0.25 and 0.25 respectively. Control and prediction horizon is equal to simulation horizon N. $N = 100$.

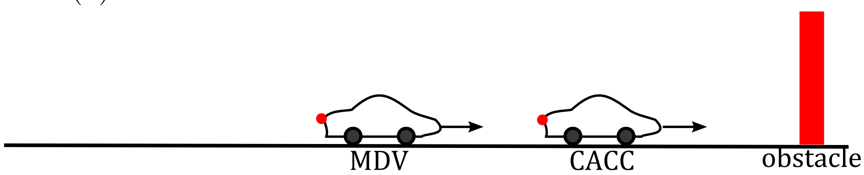
Both MDV and CACC vehicles are at halt initially. The first vehicle starts to accelerate with a fixed acceleration of 1 m/s^2, from 800 meters away from the obstacle (refer to Figure 4.8a). Once the desired speed of 90 km/h (25 m/s) is reached, it cruises until the "notification distance"(refer to Figure 4.8b). The notification distance is defined as a distance from the obstacle, where the CACC vehicle detects the obstacle using its sensors, or is notified about the obstacle over V2V/V2I communications. Overview of the scenario is illustrated in Figure 4.8. Three different notification distances are used in this study: *i*) 95.9 meters[4]; *ii*) 120 meters; and *iii*) 150 meters. We assume the proposed algorithm intervenes and helps coordinate



(a) The CACC vehicle starts 800 meters away from the obstacle at zero speed



(b) The CACC vehicle has reached notification distance



(c) After the notification distance, the CACC vehicle starts braking to stop before the obstacle

Figure 4.8: Overview of the simulated scenario

a braking procedure as soon as the CACC vehicle is 'notified' about a potential obstacle (refer to Figure 4.8c).

To involve human drivers in this work, VTI's driving simulation software is used in combination with MATLAB, as illustrated in Figure 4.9. All three software blocks in Figure 4.9 are running on the same desktop computer. The first MATLAB block, *Centralized Controller*, is executing the control computations explained by (4.5). The middle block, *Synchronization*, is for synchronizing data exchanges

[4]This is the distance at which at least one DSRC/ITS-G5 safety message would be received with 99.5% probability [158]

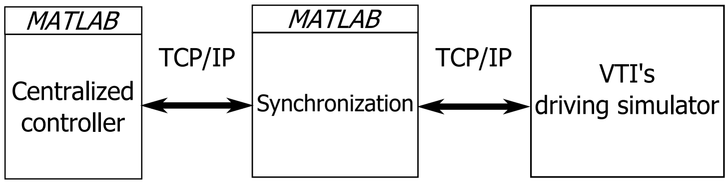


Figure 4.9: Connections between MATLAB scripts and the driving simulator

between the driving simulator and the centralized controller over a TCP connection. The data exchange happens every 0.1 second, which is also the simulation time step in MATLAB. The last block, *VTI's driving simulator*, handles the task of fetching inputs from the human driver and displaying positions of the CACC vehicle controlled by MATLAB. The human driver uses a gaming steering wheel and pedals to control the MDV. In the scenario where an actual human controlled the MDV, we have six participants (2 women and 4 men) involved in the experiments. For each notification distance, each driver is driving the scenario at least two times, in no particular order. The driver was instructed to follow the CACC vehicle in front without making any lane changes.

These computations ideally, should take place with a particular frequency known as the controller update rate. However, computations might take longer and controls may not be generated at the desired update rate. Alternately, computations might have terminated but not successful (the optimization problem was infeasible). In such scenario where control inputs for a particular time slot are not computed (due to processing time or infeasibility), we assume CACC vehicles continue to apply next control input from the previously received control values stored in the buffer. If the control computation is infeasible and the control buffer is empty, CACC vehicles apply brakes with maximum jerk added to previous acceleration value.

The mixed vehicle scenario consists of a MDV following a CACC vehicle, as shown in Figure 4.8. We analyze the following *model mismatch* in the braking scenario under two cases:

- Case A: the assumed driving model is the model 2 introduced by equations (3.14)- (3.17) in section 3.2.1 and the actual driving is based on a modified version of IDM introduced in equation (3.12)

- Case B: the assumed driving model is the model 2 introduced equations (3.14)- (3.17) in section 3.2.1 and the actual driving inputs are obtained from the driving simulator

Results of the above simulation scenarios are plotted in Figure 4.10. We observe in Case A, acceleration and velocity profiles are notably smoother compared to the scenario where the real human is driving the MDV. As one of the evaluation metric is the discomfort while braking, we compute discomfort values after vehicles are notified at the notification distance. The average discomfort value of Case B is higher than that of Case A. Moreover, collisions take place when an actual human is driving compared to no collisions when a mathematical model is used. These results are summarized in Table 4.2.

Table 4.2: Mixed traffic simulation results

	Case A	Case B
Discomfort	6.66	12.76
Collisions avoided (%)	100	53.57

As expected, an increase in notification distance results in a higher percentage of collision avoidance. For Case B, the percentage of collisions avoided at 95.9 m, 120 m, and 150 m is 35.72%, 57.89%, and 60.87% respectively. Overall, 53.57% of collisions were avoided. Whereas in case A, MDVs implement IDM, 100% collisions were avoided.

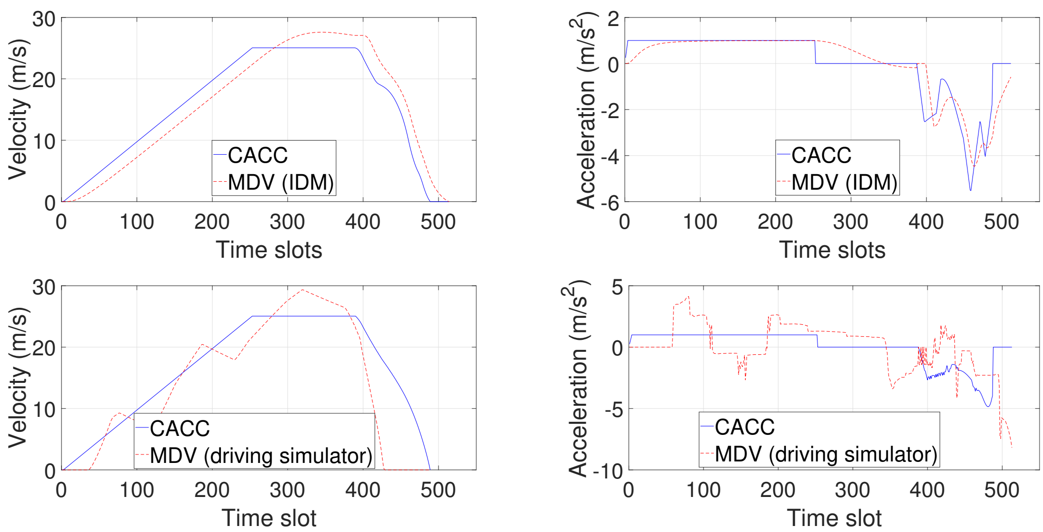


Figure 4.10: Mixed vehicle scenario results: plots on the top show results from case A; bottom plots show results from case B

There are several reasons for collisions, due to the differences between the assumed value in the centralized controller and the actual value in the driving simulator, e.g., the difference in perception response time, the difference in braking capacity and the maximum value of jerk sustainability. In such cases, control optimization computations are infeasible for the assumed values of jerk and the braking capacity. But it may actually be feasible for MDV to brake using the driving simulator.

Furthermore, we observe that the optimization computations take more time than expected and thus control optimization is not computed at the controller frequency. Computations ideally should take less than 0.1 seconds to ensure real-time performance. However, in our experiments, it takes a maximum of up to 2 seconds (20 times more) to compute. On an average, we have an optimization computation completed every 0.3 seconds. The previously computed control inputs in the buffer are old and may be from a few seconds ago. These control inputs from the buffer are sometimes not able to avoid collisions.

This experiment not only helped verify the performance of the proposed controller's performance but also simulate the coexistence of MDVs and CACC vehicles.

4.4 Impact of Control errors

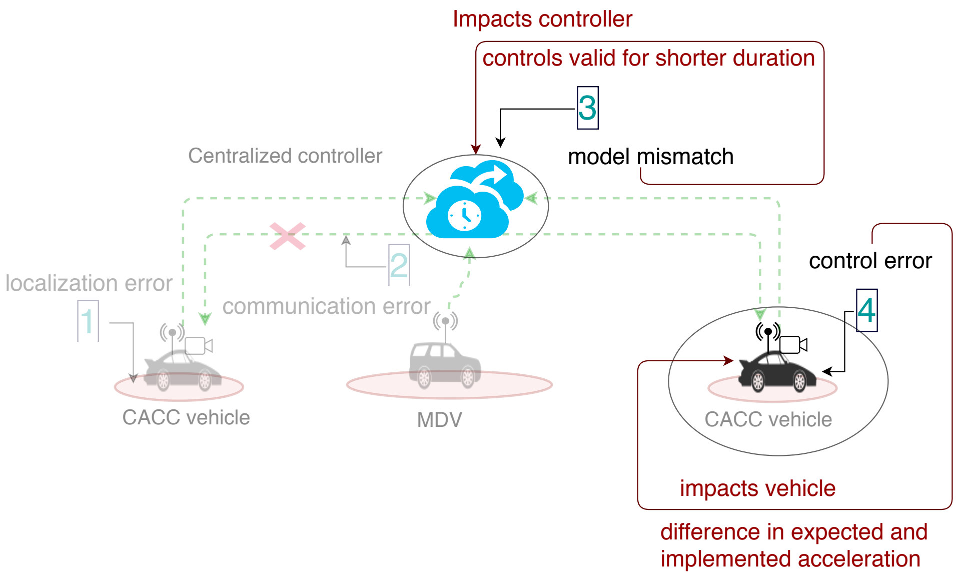


Figure 4.11: Model mismatch and control errors impacting centralized controller

In this set of simulations, the goal is to highlight the impact of control errors in the presence of model mismatch as shown in Figure 4.11. Control errors are modeled as the difference in the desired and the actually implemented controls. One set of simulations (Type 1), we assume, the desired and the actual control are the same. The maximum and minimum allowed change in acceleration implemented by the controller is restricted to 0.25 per time slot using Eq. (3.6). Jerk value around $2\,\mathrm{m/s^3}$ is usually considered comfortable in the literature [159]. In this work, increase and decrease in braking capacity is restricted to $2.5\,\mathrm{m/s^3}$ and $-2.5\,\mathrm{m/s^3}$ respectively. As there are 10 time slots per second, $\Delta u^{\min}, \Delta u^{\max}$ values are $(-0.25, +0.25)$ respectively. The mathematical system of equations representing simulation Type 1 is represented by (4.6). Note that this is the same as simulation Type 2 from the previous subsection, refer to equation (4.5).

$$\text{minimize} \quad \sum_{i=1}^{n_v} \sum_{\eta=1}^{N_p} (u_i(\eta) - u_i(\eta-1))^2 + \rho\, \epsilon \tag{4.6}$$

$$\text{subject to}$$
$$(3.1),\ (3.3),\ (3.4),\ (3.5a),\ (3.5b),\ (3.6),\ (3.8),\ (4.2),$$
$$\text{Assumed MDV model: } (3.14)\ (3.15)\ (3.16)\ (3.17)$$

In the other set of simulations (Type 2), we assume, the lower level controller is not perfect and can not generate the desired control immediately leading to different desired and actual controls. This has been implemented in our algorithm, like in PLEXE, using (3.18),(3.19). As CACC vehicles do not have fixed control profile, rather implement controls derived from centralized controller, the centralized controller must have jerk limitations. If increase and decrease in braking capacity is restricted to $2.5\,\mathrm{m/s^3}$ and $-2.5\,\mathrm{m/s^3}$ respectively, the *desired acceleration* at slot $n+1$ will only be different by a maximum of 0.25 from the *actual acceleration* at slot n. And the difference in the actuated control value at $n+1$ and the actual acceleration at n would differ by less than 0.25 due to the engine behavior. In order to accommodate this, engine constant τ value of $0.2\,\mathrm{s}$ is used in Eq. (3.18), (3.19). Δt value is fixed, $0.1\,\mathrm{s}$ as we assume the system operates at 10 Hz. The maximum and minimum allowed change in acceleration during computations is set to 1 per time slot in (3.6). The mathematical system of equations representing simulation Type 2 is represented by (4.7).

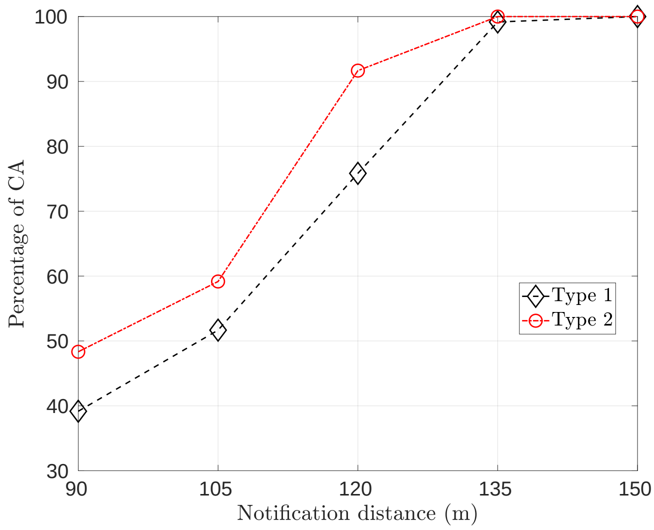


Figure 4.12: Collision avoidance evaluation for different ways of implementing control error

$$\text{minimize} \quad \sum_{i=1}^{n_v} \sum_{\eta=1}^{N_p} (u_i(\eta) - u_i(\eta-1))^2 + \rho\,\epsilon \tag{4.7}$$

$$\text{subject to}$$
$$(3.1),\ (3.3),\ (3.4),\ (3.5a),\ (3.5b),\ (3.6),\ (3.8),\ (4.2),$$
$$\text{Assumed MDV model: } (3.14)\ (3.15)\ (3.16)\ (3.17)$$
$$(3.18),\ (3.19)$$

Although control errors could be countered by implementing controls such that the effective (applied) control is equal to the desired control, this would require accurate information of the engine profile and the relation between the desired and the effective acceleration. However, this relation changes with engine age, road and weather conditions, tyre age, friction in vehicle components, etc. and thus has been avoided. On the other hand, recomputation of controls, which is the basis of MPC, ensures that control errors described above can be mitigated effectively. Thus, in this work, rely on MPC to mitigate control errors.

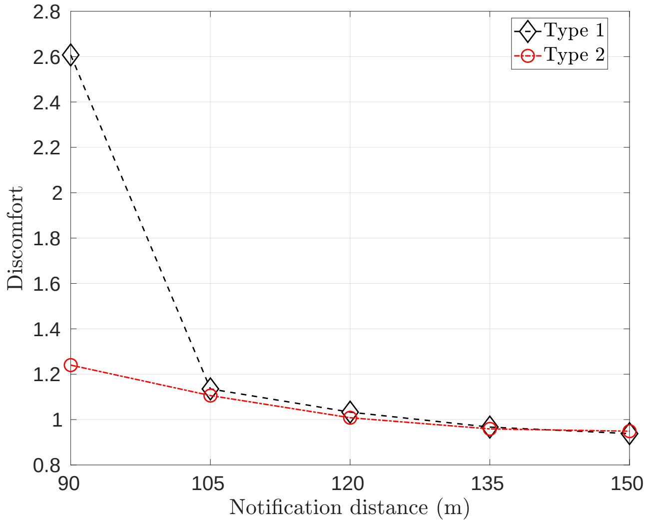


Figure 4.13: Discomfort evaluation for different ways of implementing control error

For both sets of simulations (Type 1 and Type 2), MPC was implemented with buffer. Simulation results are summarized in Figure 4.12 We observe that the number of collisions avoided is either equal or more when lower engine vehicle profile is correctly implemented, and the centralized controller is under the influence of model mismatch. The discomfort is either lower or similar as seen in Figure 4.13. Moreover, lower level vehicle controller behavior is an ingrained feature of the vehicle controller and thus it has been used in all simulations hereon.

4.5 Robustness to Communication error

The previously introduced controller is now evaluated under the influence of communication error. The goal of these simulations is to evaluate the performance of the centralized controller under multiple consecutive packet losses and compare it to the performance when there are random packet losses, and the controller is under the influence of model mismatch and the lower level controller performance as shown in Figure 4.14. In this work, communication delays of less than a time slot are ignored. If the packet being transmitted is delayed for more than a time slot, the packet is assumed to be lost. One of the fall back techniques (introduced later in this section) is to be used in this case.

Communication impairments have been implemented as introduced in Section 3.2.4. Markov model and Bernoulli model are used to simulate communication losses. When the packet loss is modeled using a Markov model, without loss of generality, two sets of values which correspond to poor communication denoted by *MarkovBad* using $[p_r = 0.8, p_l = 0.75]$ and good communication conditions denoted using *MarkovGood* using $[p_r = 0.998, p_l = 0.30]$ are simulated. The mathematical system of equations representing simulations using Markov model for communication errors is represented by (4.8).

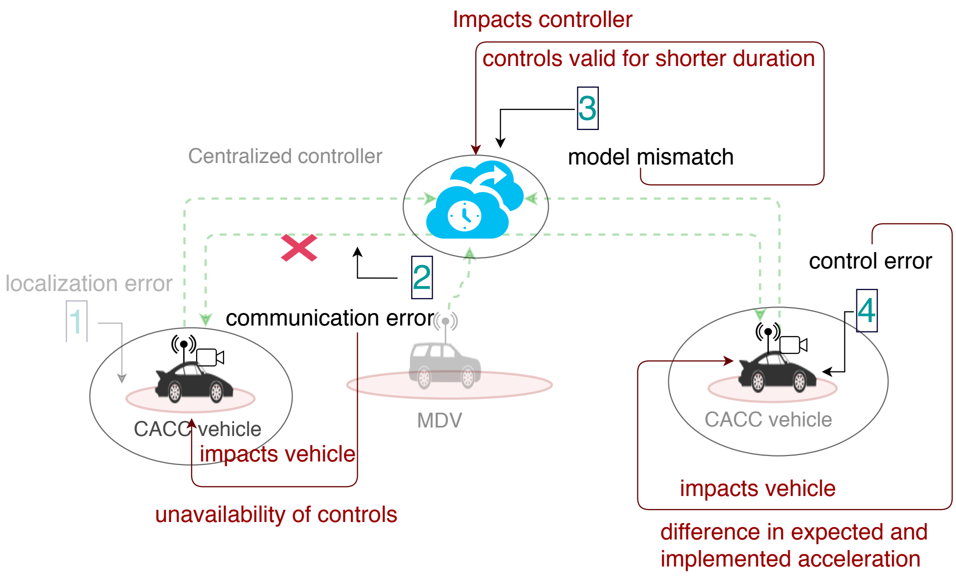


Figure 4.14: Model mismatch, control errors and communication errors impacting the centralized controller

Alternately, packet losses have also been modeled as random occurrences using *Bernoulli* model. The mathematical system of equations representing simulations using a Bernoulli model for communication errors is represented by (4.9).

More realistic values for Markov model parameters can be obtained from the communication channel in real life scenarios, and this is left for future work. Hereon, terms MarkovBad, MarkovGood and Bernoulli represent the modeling of the communication channel using the above introduced parameters.

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n_v} \sum_{\eta=1}^{N_p} (u_i(\eta) - u_i(\eta-1))^2 + \rho\ \epsilon \\ & \text{subject to} \\ & (3.1),\ (3.3),\ (3.4),\ (3.5a),\ (3.5b),\ (3.6),\ (3.8),\ (4.2), \\ & \text{Assumed MDV model: } (3.14)\ (3.15)\ (3.16)\ (3.17) \\ & (3.18),\ (3.19),\ (3.23),\ (3.24) \end{aligned} \tag{4.8}$$

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n_v} \sum_{\eta=1}^{N_p} (u_i(\eta) - u_i(\eta-1))^2 + \rho\ \epsilon \\ & \text{subject to} \\ & (3.1),\ (3.3),\ (3.4),\ (3.5a),\ (3.5b),\ (3.6),\ (3.8),\ (4.2), \\ & \text{Assumed MDV model: } (3.14)\ (3.15)\ (3.16)\ (3.17) \\ & (3.18),\ (3.19),\ (3.22) \end{aligned} \tag{4.9}$$

Countering communication errors:

Under nonideal circumstances when new control inputs are not received on the downlink, the vehicle switches to one of the three fallback strategies (previous, ACC or buffer) [35]:

1. the vehicle retains its **previous** applied acceleration (no buffer). i.e.: $u_i(n) = u_i(n-1)$.

2. the vehicle switches back to **ACC** mode (no buffer). i.e., the acceleration value obtained from ACC based on IDM (2.1) is applied.

3. the vehicle fetches acceleration value from the **buffer** as explained in the Section 4.2.

During the switch to and from the fallback strategy, (3.6) is implemented as a filter to ensure jerks remain within limits. The first two strategies do not require a buffer, whereas the last strategy can only be applied if a buffer is present.

We compare the results of simulations of the centralized braking procedure with different fall back techniques under communication error.Note that in all these simulations, lower level engine behavior is considered and model mismatch is present.

4.5.1 Evaluation set 1

In this subsection, the performance of the centralized controller for each fall back strategy at a time under the influence of communication errors.

Fallback: Retain previous acceleration

The performance of the centralized controller under packet losses modeled by three different types is analyzed next.

MarkovBad will result in multiple consecutive packet losses in the form of bursts of packet losses and in this case, the previously applied acceleration will be retained longer (without any control update). i.e., constant erroneous control will be applied which leads to collisions. When packet losses are not in bursts, frequently updated controls are retained for shorter duration. The latter is better compared to the previous scenario for collision avoidance. Thus the performance is better under Bernoulli than under MarkovBad as seen in Figure 4.15. This is even though the communication loss under Bernoulli (50.92%) is more than the communication loss under MarkovBad (42.68%). Thus we can conclude that burst errors result in accidents if the fallback strategy is to ‘retain previous acceleration’. At low packet loss modeled using MarkovGood (1.422%), the performance of the controller is better than other two.

The above discussion means, the controller is better situated to handle a low percentage of packet losses or random packet losses compared to burst errors when the fallback strategy is to retain previous acceleration.

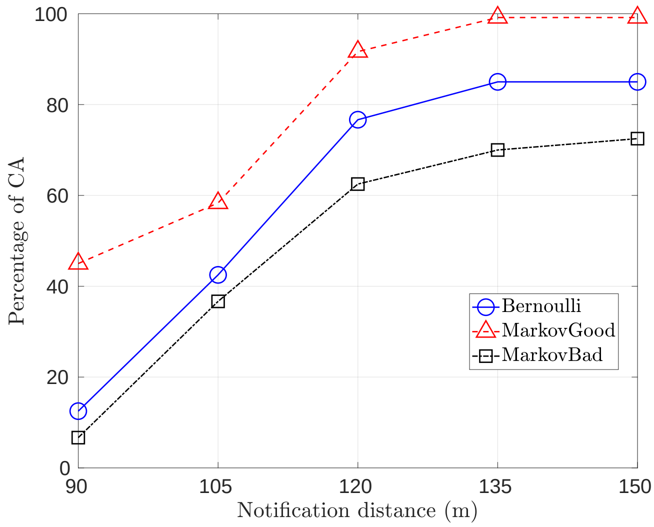
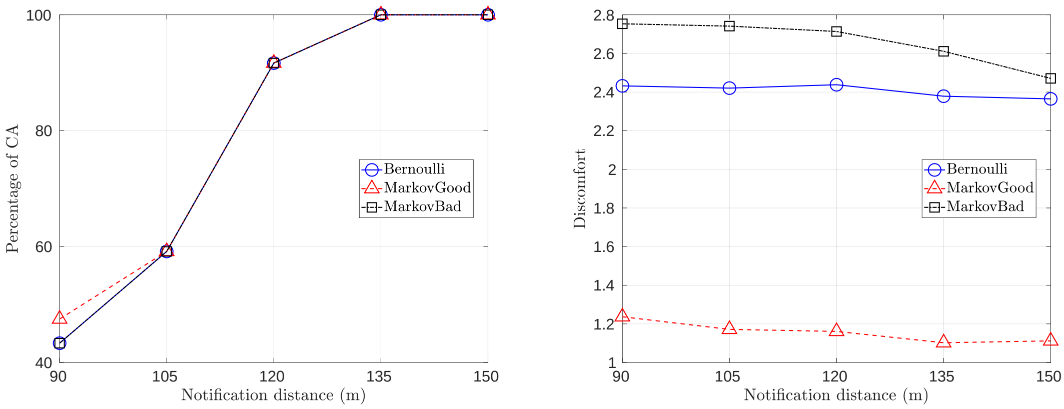


Figure 4.15: Collision avoidance evaluation under different communication error models for 'retain' previous acceleration fallback strategy

Fallback: ACC

Next, we summarize the results when the CACC vehicle falls back to ACC (implementing IDM using equations (2.1)) in case of packet loss.



(a) Collision avoidance statistics

(b) Discomfort statistics

Figure 4.16: Simulation results when fall back strategy is to switch to ACC

As evident from the Figure 4.16a the performance of the fall back to ACC strategy, with packet losses modeled as Bernoulli occurrences is similar with packet losses modeled as MarkovBad, in terms of collision avoidance but the average jerk is worse for the latter as seen in Figure 4.16b.

Fallback: Buffer

Next, we summarize the results when the CACC vehicle uses control values stored in the buffer in case of packet loss.

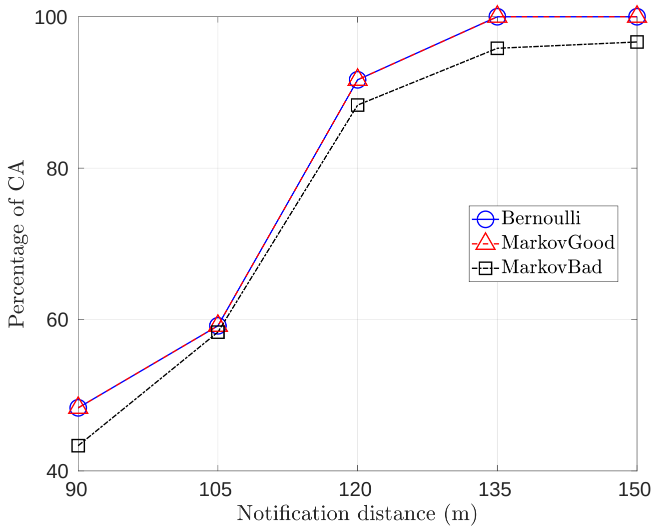


Figure 4.17: Collision avoidance evaluation under different communication error models for buffer based fallback strategy

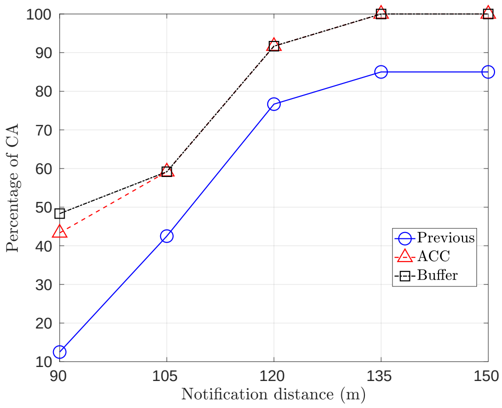
Despite a lower percentage of packet loss with Markov Bad model compared to the case when the Bernoulli model is used, the number of CA is slightly higher for Bernoulli as seen in Figure 4.17. This again proves that multiple consecutive packet losses is more dangerous compared to random packet losses and can lead to not only an increase in discomfort but also collisions. The performance of the buffer backed centralized controller is similar under Bernoulli distribution and Markov Good model (refer to Figure 4.17) despite communication loss being approximately 51.35% for Bernoulli, 42.13 % for MarkovBad and 1.40% for MarkovGood. We can conclude that the controller is better situated to handle a low percentage of packet losses or random packet losses compared to burst errors when the fallback strategy is to use controls from the buffer.

4.5.2 Evaluation set 2

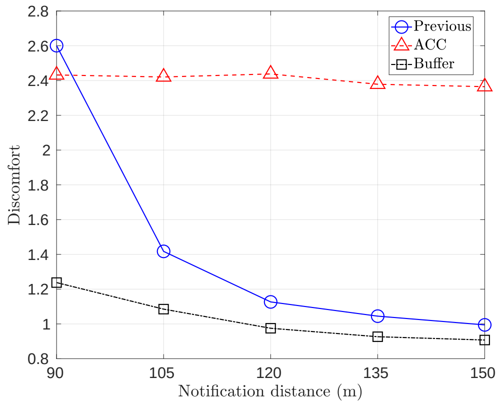
The goal of this type of simulation is to find the best type of fall back strategy for different communication error models.

The results show that the number of collisions avoided with fall back strategy of retaining previous acceleration is the least when communication errors are modeled using the Bernoulli model; The use of the buffer has the best performance in terms of both, the number of collisions avoided and discomfort as seen in Figure 4.18a and 4.18b.

When the vehicle falls back to ACC, it involves continuous computation of controls to be implemented; whereas the use of buffer ensures availability of controls computed a while ago. It is for this reason that the fall back to ACC strategy is slightly better (approximately 3 %) compared to fall back strategy using the buffer when multiple consecutive packet losses are modeled using MarkovBad model (refer to Figure 4.19a). Note that the fall back to ACC usually leads to higher discomfort whereas buffer based fall back is more comfortable as seen in Figure 4.19b.

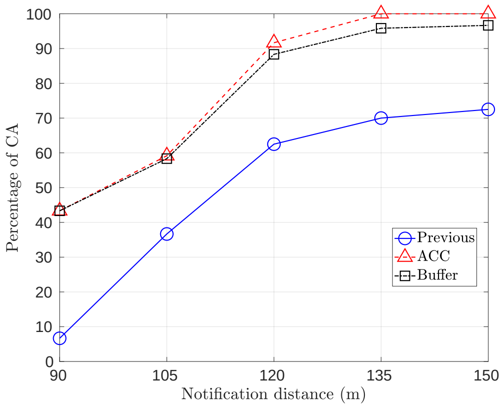


(a) Collision avoidance statistics

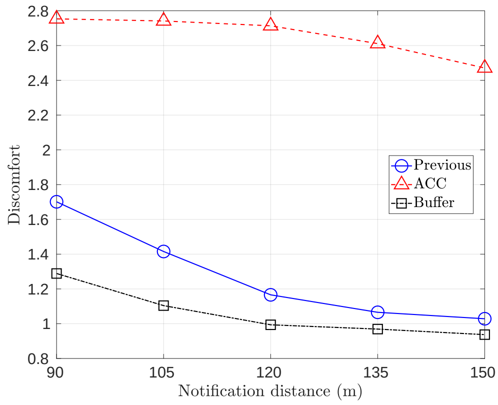


(b) Discomfort statistics

Figure 4.18: Summary or results for communication model: Bernoulli



(a) Collision avoidance statistics



(b) Discomfort statistics

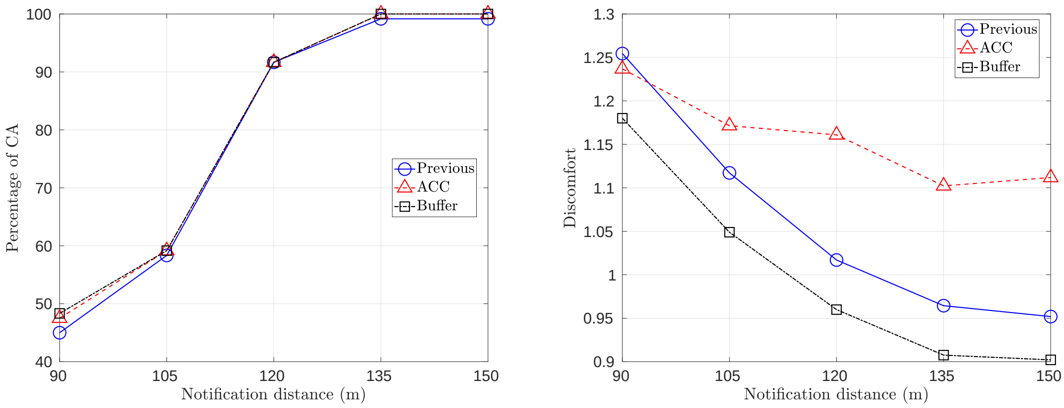
Figure 4.19: Summary or results for communication model: MarkovBad

The performance of the buffer based fall back is better in terms of both, discomfort and collision avoidance when the communication channel is modeled as MarkovGood as seen in Figure 4.20a and Figure 4.20b.

To summarize, fallback to buffer has better collision avoidance when the communication channel is modeled using MarkovGood and Bernoulli models; fallback to ACC has better collision avoidance when the communication channel is modeled using MarkovBad model.

4.6 Robustness to Localization error

In this particular study, the goal is to highlight the impact of localization error on a centralized controller, when the algorithm does not account for localization errors (a non-robust MPC controller) while considering lower level engine behavior (control error) and model mismatch as depicted in Figure 4.21. The performance of



(a) Collision avoidance statistics (b) Discomfort statistics

Figure 4.20: Summary or results for communication model: MarkovGood

the centralized controller which counters model mismatch (a robust MPC controller, introduced in (4.13)) is then evaluated against the non-robust controller. The buffer is used to counter infeasibilities.

When localization errors are not accounted for, control inputs are generated using perceived localization (positioning information computed by the vehicle) which is different from the true localization. Due to the difference in the true localization and the perceived localization, control inputs generated are usually different, which could be a source of collisions. Control inputs generated using perceived information are applied to the vehicles in their true positions in a **non-robust controller**.

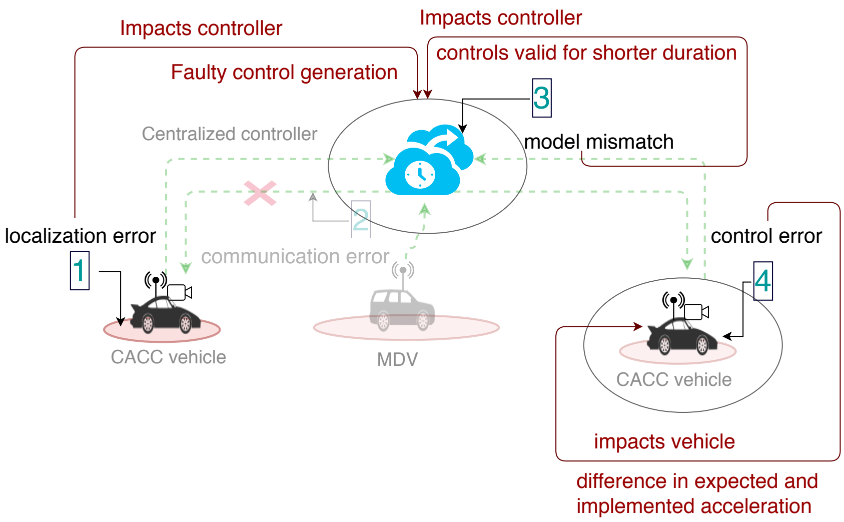


Figure 4.21: Model mismatch, control errors and localization errors impacting centralized controller

4.6.1 Controller model robust to localization error

With the knowledge only about the perceived localization (p_i^*) and the error in the localization (e_i), the 'potential location' ($p_{i,1}$) of the vehicle which is anywhere on the circumference in red is going to be used in our methodology.

We adapt this 2D scenario to a 1D scenario (as shown in the bottom part of the figure 4.22), accounting for errors in longitudinal direction only. Note that ego-vehicle can be assumed to be located anywhere between $p_{i,1}$ and $p_{i,2}$ and thus the potential area which could be occupied by the ego vehicle would be between $p_{i,1}$ to $p_{i,3}$. This is the maximum area which needs to be 'reserved' for this vehicle in order to guarantee collision avoidance with this vehicle. Distance between bounds $p_{i,1}$ and $p_{i,2}$ is 2*e_i. The new length of the vehicle $l_{i,e}$, and the location of the vehicle is assumed to be at $p_{i,1}$ is given by equations (4.10), (4.11).

$$l_{i,e} = l_i + 2 * e_i \tag{4.10}$$

$$p_{i,1} = p_i^* + e_i \tag{4.11}$$

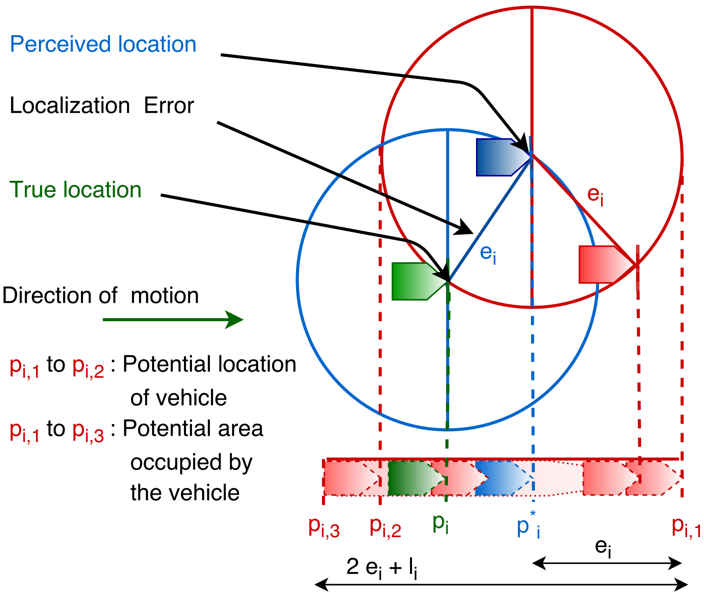


Figure 4.22: Modeling localization errors

where l_i is the actual length of the vehicle. As $e_i > 0$, $l_{i,e} > l_i$. This leads to a change in the perceived distance between the vehicles as well. We incorporate this in the equations (3.1) as follows:

$$\begin{bmatrix} p_{i,1}^{\min} \\ v_i^{\min} \end{bmatrix} \leq x_i(n) \leq \begin{bmatrix} p_{i,1}^{\max} \\ v_i^{\max} \end{bmatrix} \tag{4.12a}$$

$$u_i^{\min} \leq u_i(n) \leq u_i^{\max} \tag{4.12b}$$

Moreover, the robust MPC framework now is as follows:

$$\text{minimize} \quad \sum_{i=1}^{n_v} \sum_{\eta=1}^{N_p} (u_i(\eta) - u_i(\eta - 1))^2 + \rho\, \epsilon \tag{4.13}$$

subject to

(3.3), (3.4), (3.8), (4.2), (4.10), (4.11), (4.12a), (4.12b),

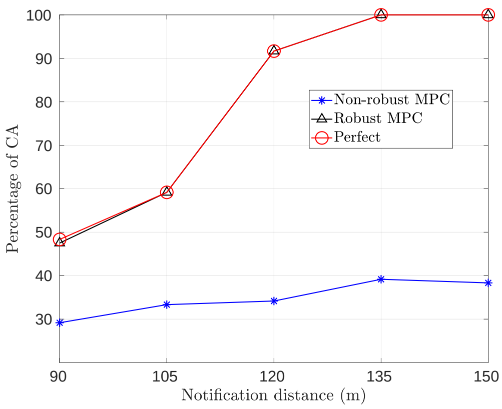
Assumed MDV model: (3.14) (3.15) (3.16) (3.17)

(3.18), (3.19)

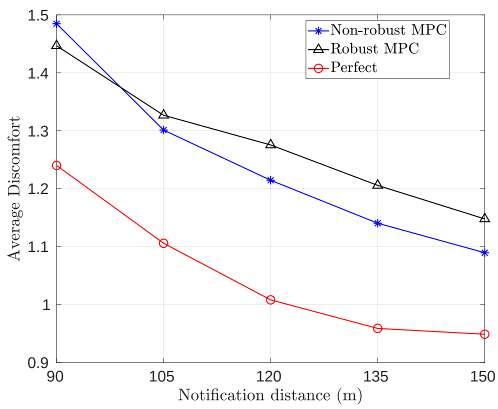
4.6.2 Evaluation of the robust controller

First, simulations are carried out assuming homogeneous localization system, meaning, the standard deviation (STD) of error for all vehicles remains the same. Different standard deviation of error values (ϕ) are chosen from a range of values ($\Phi = [4, 2, 1, 0.5, 0.25]$). $\phi \in \Phi$. Next, simulation results assuming heterogeneous localization system on these vehicles are considered. i.e., We assume CACC vehicles use mapmatching, cameras, lidars, etc. to have precise localization information with zero mean and STD error of 0.25 m whereas MDVs use GPS and have localization errors derived from a distribution with zero mean and STD error of 4 m.

Simulations with heterogeneous localization system



(a) Collision avoidance statistics



(b) Discomfort statistics

Figure 4.23: Comparing a non-robust and a robust centralized controller in presence of localization errors in a system consisting of vehicles with different localization systems (a heterogeneous system).

Next, consider a stream of vehicles with heterogeneous localization system. Figure 4.23a has been plotted to compare the performance of the non-robust centralized controller and the robust centralized controller under heterogeneous localization errors against the perfect localization scenario (no localization or communication error, labeled as Perfect). Note that the scenario which refers to perfect localization condition, has results taken from the simulation with control error implementation type 2 (refer to Figure 4.12) from Section 4.4. Comparing the plots in Figure 4.23a, we observe the percentage of CA with a robust MPC is higher than that of a non-robust case, for all notification distances. This highlights the benefits of implementing a robust MPC controller; it avoids more collisions compared to a non-robust controller by effectively mitigating model mismatches and countering localization errors. From the Figure 4.23a, we can conclude that the performance of the robust controller under the impact of localization errors is similar to the performance of the centralized controller when localization errors are absent. Note that when there are no localization errors, the performance of the robust and non-robust controller is the same. Moreover, for a notification distance of 135 m and more, we can attain almost 100 % CA using a robust controller instead of approximately 39 % with a non-robust controller.

The performance of in terms of discomfort is better when localization errors are absent compared to the performance of the robust or non-robust controller when localization error is present. Moreover, the performance of the robust controller is worse compared to the perfect scenario in terms of discomfort. From Figure 4.23b and Figure 4.23a two conclusions can be drawn: 1. Robust MPC controller can avoid more collisions compared to non-robust MPC controller at the cost of additional discomfort 2. Robust MPC controller can avoid approximately the same number of collisions under localization errors as the MPC based controller without any errors (perfect), but at an additional discomfort.

Simulations with homogeneous localization system

Next, the simulation results for vehicles assuming homogeneous localization system are evaluated. First, non-robust controller's performance is evaluated. CA is more for a smaller value of ϕ as lower standard deviation of localization error can be better handled by a non-robust controller as seen in Figure 4.24. For the robust MPC controller, the number of CA is almost constant over the range of ϕ, refer to Figure 4.25. This implies that the robust controller is capable of mitigating the impact of localization errors. The robust controller is able to achieve 100 % CA for notification distance 135 m or more for the range of ϕ used in the simulations.

The assumed vehicle length and the area over which the vehicle is assumed to be located evolves with every change in the localization error. This results in drastic changes in computed controls. Bigger the change in localization error between two timeslots, bigger is the change in acceleration. We compute discomfort for cases where collisions were avoided as the two-norm of change in acceleration between two time slots per vehicle for cases where CA was successful and plot it in Figure 4.26. We observe that discomfort is higher for higher values of std of localization error. As notification distance increases, discomfort decreases as vehicles, in general, have larger freedom to brake.

To summarize, the performance of the robust controller is better than the non-robust controller especially at high values of standard deviation of localization error. As notification distance increases, collisions avoided increases and discomfort decreases.

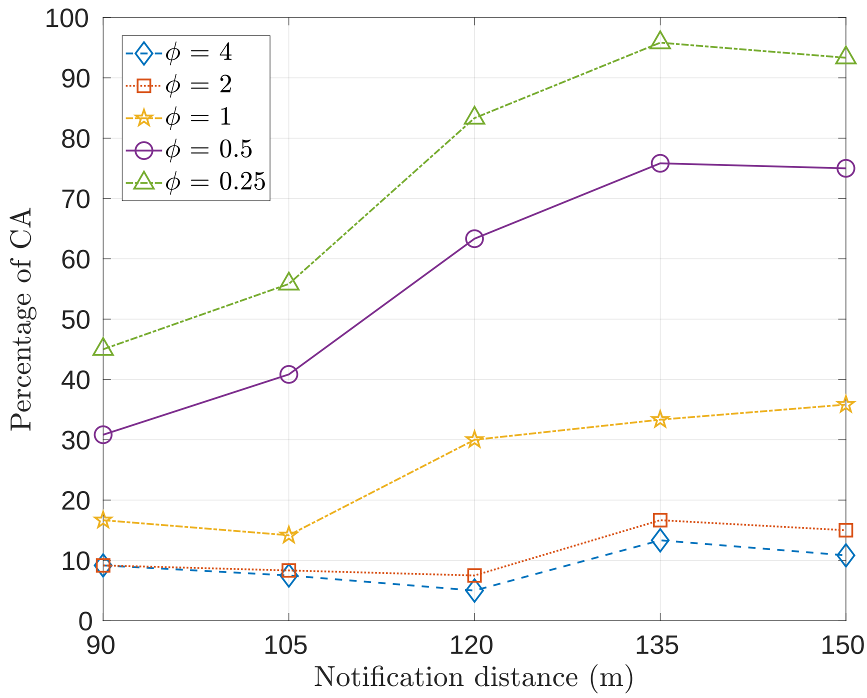


Figure 4.24: Collision avoidance statistics using a non-robust MPC for different values of ϕ and notification distance (for a homogeneous localization system)

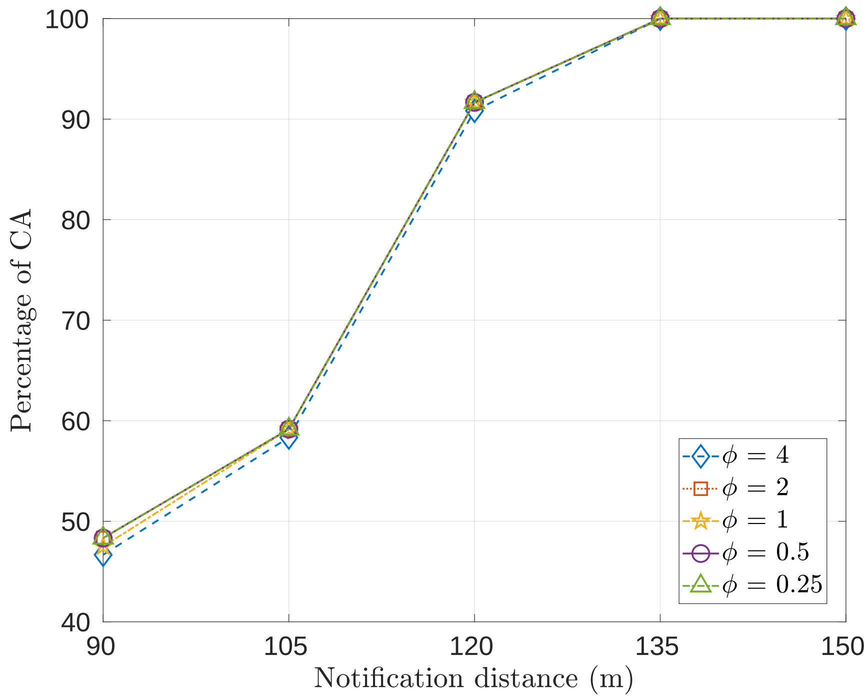


Figure 4.25: Collision avoidance statistics using a robust MPC for different values of ϕ and notification distance (for a homogeneous localization system)

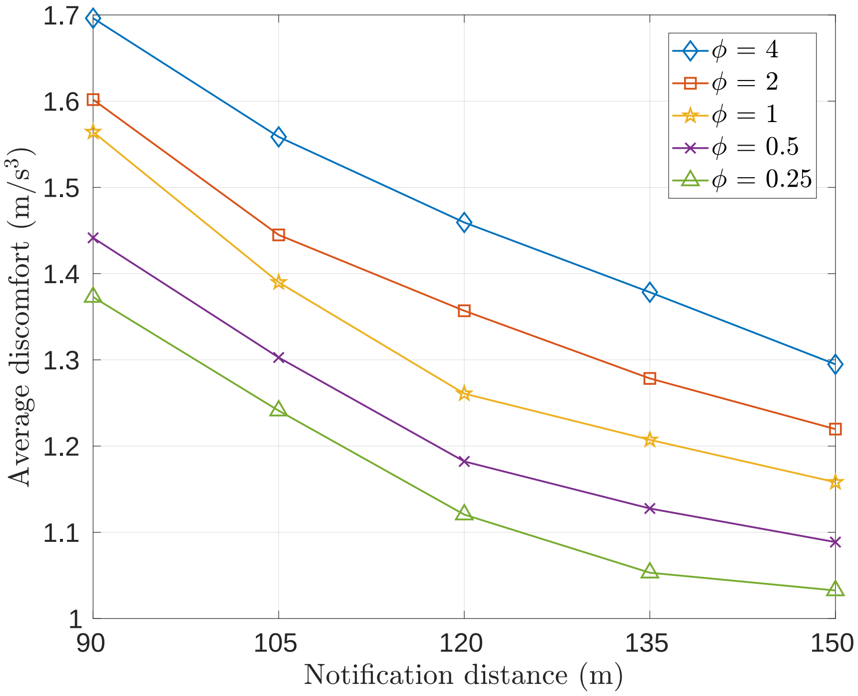


Figure 4.26: Average values of discomfort for different values of ϕ and notification distance (for a homogeneous localization system) using a robust MPC controller

4.6.3 Warning for enhanced safety

In circumstances where vehicles need to brake and come to a halt, they are expected to be able to avoid collisions. However, the presence of localization errors creates uncertainty in collision avoidance. Vehicles must operate such that collision avoidance uncertainty is low. This subsection presents a method that proposes different options with alternate parameters where vehicles should operate such that collision avoidance uncertainty is reduced in an environment with different localization errors [33].

We first create a database which relates localization errors, uncertainty in collision avoidance and flow capacity for a wide range of vehicle state parameters like average velocity, inter-vehicular distances, etc. For a given traffic scenario uncertainty in collision avoidance is computed by referencing the sample data with the database. The proposed method is then implemented to evaluate uncertainty values of three options and the one with the least uncertainty should be chosen. This idea was introduced in [33]. When control computations fail, the simulation was stopped because buffer was not used in this simulation. Although the performance is sub-optimal without the use of the buffer, the general concept proposed in [33] still remains valid.

Initial average velocity and average inter vehicle distance between vehicles is chosen as a parameter and the result of the simulation, whether a collision was avoided or not is taken as the second parameter. Scatter plots of all simulations, each point representing a simulation sample is plotted in a two-dimension velocity/inter-distance domain, with the corresponding traffic flow value in grayscale in Figure 4.27- 4.29. The first objective is to visually illustrate the impact of location errors on the distribution of avoided collisions in the figures and its impact on traffic flows. Accordingly, the methodology is defined.

On the one hand, *Collision Avoided (CA)* samples are samples where control inputs avoided an accident despite localization errors. *CA* samples are represented by 'x'. Would the true position be known, *CA* samples would obviously also avoid the accident.

On the other hand, *Probable Collision (PC)* samples are samples where the controller is unable to provide control inputs. Unavailability of control inputs does not strictly imply collision, but simply defines an uncertainty in the controller.[5] *PC* samples are represented by '·'. A sample might be classified as a *PC* sample in two cases:

(A) **optimization problem is invalid -** due to localization errors, the perceived distance between vehicles is less than or equal to zero. In this case, the algorithm is not run as the constraint represented by Eq. (3.8) is not satisfied.

(B) **optimization problem not solvable -** due to localization errors, the perceived inter-vehicular distance is greater than zero but, and the algorithm is not able to avoid the collision.

Accordingly, we further need to know if *PC* samples would lead to collision avoidance would the true position be known. Collisions which take place (despite the information of true location) are marked as '◯'. However, in real-time operation, controller would not know this and would attempt to avoid this situation at any cost. Table 4.3 summarizes the different icons used.

Table 4.3: Icons used and their significance

Icon	Collisions avoided with localization errors using proposed approach	Collisions avoided with true position information
x	Yes	Yes
·	Uncertain	–
◯	–	No
⊙	No	No

We represent the collision avoidance statistics for each of the 600 samples for $\phi = 4$ and $\phi = 2$ in Figure 4.27 and Figure 4.28. We observe there are different regions with a different concentration of *CA* to *PC* samples. *PC* represents uncertainty as the centralized controller cannot be sure of collision free braking. Thus *PC* should be avoided and the objective is to make CACC vehicles operate in the region with a

[5]Note that buffer was not used and the optimization problem was stopped as soon as the computation was infeasible; such scenarios were classified as PC [33]. This concept can be improved upon with the use of a buffer to remove PC samples, and reclassify them either into collisions avoided or collisions.

majority of *CA* samples. We illustrate such region as a *convex hull*, which is defined with the following parameters:

- α - the target ratio of *CA* to *PC* samples; it represents the confidence in the system.
- β - the ratio of *PC* samples outside the convex hull to the total *PC* samples; this represents the percentage of *PC* samples avoided by operating in the convex hull.
- *margin* - a truncation margin for *CA* to remove extreme *CA* samples. A low margin will provide a more compact hull.

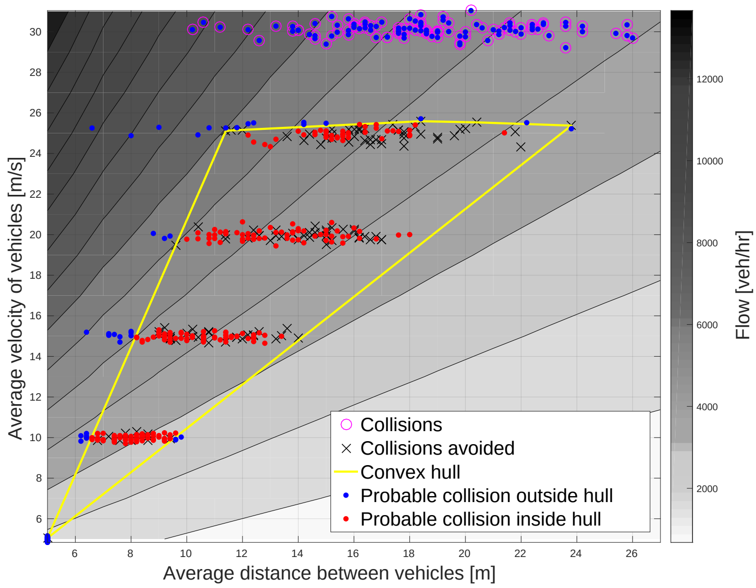


Figure 4.27: CACC only vehicles - collision avoidance statistics for $\phi = 4$ m and a margin of 0.5

Let the *Operating Region (OR)* for the controller be the area inside the *convex hull*. All samples inside *OR* form the reference database. The *margin* parameter is used to avoid regions with low number of *CA* samples and high number of *PC* samples from getting into the *OR*. The controller therefore needs to find a *margin*[6] that creates an *OR* by maximizing α and β. Consider Figure 4.29, and Figure 4.28 depicting collision avoidance statistics for $\phi = 2$ and *margin* values of 0.15 and 0.5. α values are 2.098 and 1.990 and β values are 0.487 and 0.415 respectively for *margin* values of 0.15 and 0.5 which implies the uncertainty is lower with *margin* value 0.15. A lower margin also makes the hull more compact, which also moves the *OR* away from the preferred high traffic flow configuration. This implies that the uncertainty in the system can be reduced by changing the *margin*, but at the cost of a reduced flow capacity.

[6]A change in *margin* affects the *OR* and the reference database changes as well.

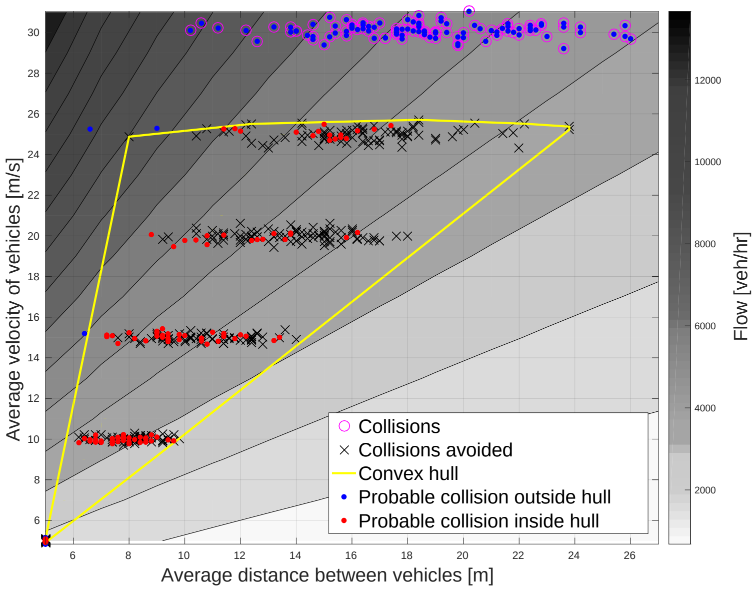


Figure 4.28: CACC only vehicles - collision avoidance statistics for $\phi = 2$ m and a margin of 0.5

Comparing Figure 4.27 and 4.28, we can observe the impact of location error on the convex hull (i.e. the OR) while maintaining the same value of margin. The OR is large for $\phi = 2$, it is significantly reduced for $\phi = 4$. Accordingly, localization errors de facto reduces traffic flow. The number of PC samples signifying uncertainty is more for higher values of ϕ.

Reducing Uncertainty and Impact on Traffic Flow

In the previous subsection, we illustrated the uncertainties created by location errors and creation of a convex hull as the controller operation area with reduced uncertainty. We next outlay a method to further reduce such uncertainty within the convex hull through traffic adaptation and quantify its impact on traffic flow.

The controller can identify uncertainty values in different areas within its convex hull. Conceptually speaking, the approach consists of moving the operational point elsewhere within its convex hull (changing velocity and inter vehicular distance parameters) of a sample from a high uncertainty area to an area with less uncertainty. An automated vehicle controller has various options to do so, such as increasing jerk tolerance, reducing speed or increasing inter vehicular distance. In this paper, we propose to investigate the impact of the latter two.

As illustrated in Figure 4.30, a sample represented by '▲' is classified in the upper circular area, which experiences a high uncertainty and will need to move to a more reliable zone. It can do so by adjusting velocity or inter vehicular distance according to one of the three options, illustrated using three green arrows in Figure 4.30:

- **Maintain Flow -** Reduce the average distance between vehicles and reduce average velocity.

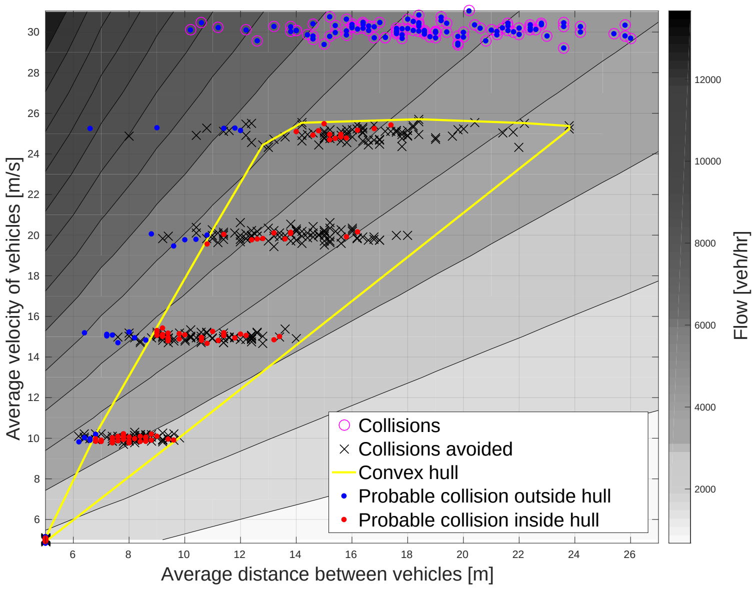


Figure 4.29: CACC only vehicles - collision avoidance statistics for $\phi = 2$ m and a margin of 0.15

- **Maintain Velocity -** Maintain average velocity but reduce the distance between vehicles.
- **Maintain Distance -** Maintain the same average distance between vehicles and reduce the velocity.

As evident, the latter two will reduce the flow, and accordingly impact the benefit of automated vehicles in future automated road transport systems. Depending on the original classification of the sample and the *OR*, the first *Maintain Flow* approach may not be feasible.

More specifically, the proposed flow adaptation is described as follows. Consider a sample (e.g., the triangle in Figure 4.30) obtained from live traffic data. We define *tile* as an area bounded by *CA* samples within a given range of speed and inter vehicular distance. If the ratio α in one of the three options (*tiles*) is bigger than that of the *tile* in which the sample is classified, the operational point should be moved.

Let us consider for example a sample with $15m$ average distance between vehicles and a $25m/s$ average velocity. If the operational point tolerance of 5% is assumed, the *tile* is defined for this sample with distance and speed between 14 *to* $16m$ and 24 *to* $26m/s$ respectively. The α value for the tile in which the sample is classified is 2.6. Compared to this, three other options (potential operational points) are displayed in Figure 4.29, all of which have smaller α values. Accordingly, the operational point must be moved. The sample is actually superimposed on top of Figure 4.29 and a zoomed in image of the same has been plotted in Figure 4.30. α values for options 1 to 3 are 7, 34, infinite respectively. All considered options have better α value compared to the *tile* in which the original sample was classified.

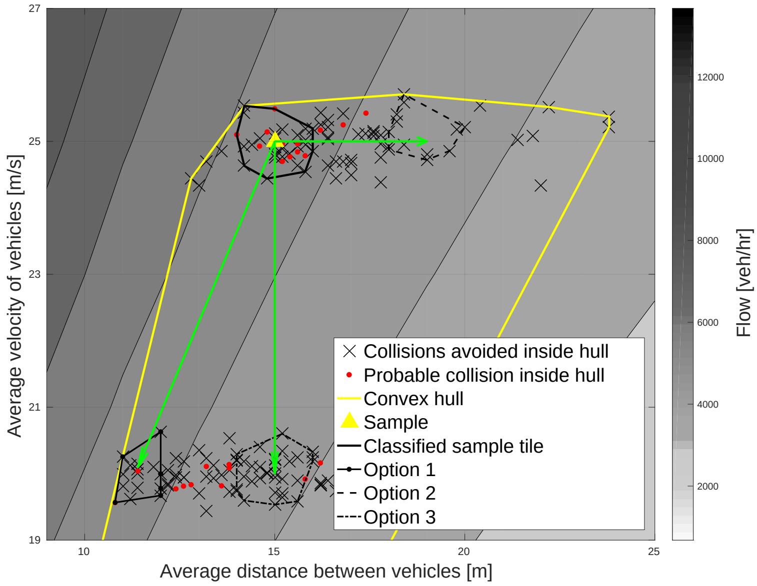


Figure 4.30: Alternative Operational Point (OP) to ensure better collision avoidance

Option 3 should be suggested which is the safest, as the α is the highest, but the flow would be lower. Whereas if the same flow needs to be maintained, option 1 should be suggested. No matter which option, all options are better than the actual sample. In this way, vehicles can operate in a region with less uncertainty.

For clarity, we restate important terms: *OR* is area inside the convex hull; Operating point is the value of the average distance between vehicles and average velocity for any sample; *Tile* is an area bounded by *CAs* within the OR, in which an Operating Point may lie.

4.7 Robustness of the controller to various errors

This section evaluates the centralized controller under the presence of all errors introduced until now as depicted in Figure 4.31. Localization errors are modeled as earlier with ϕ values from 0.25 to 4 m. When a homogeneous localization system is assumed, ϕ value is explicitly mentioned. $\phi_{CACC} = 0.25$m and $\phi_{MDV} = 4$m when an heterogeneous localization system is assumed, as introduced in Section 4.6. Communication profiles introduced in section 3.2.4 namely Bernoulli, MarkovBad and MarkovGood are used. Lower level engine performance and model mismatch are considered. Rest of the simulation parameters remain the same unless specified.

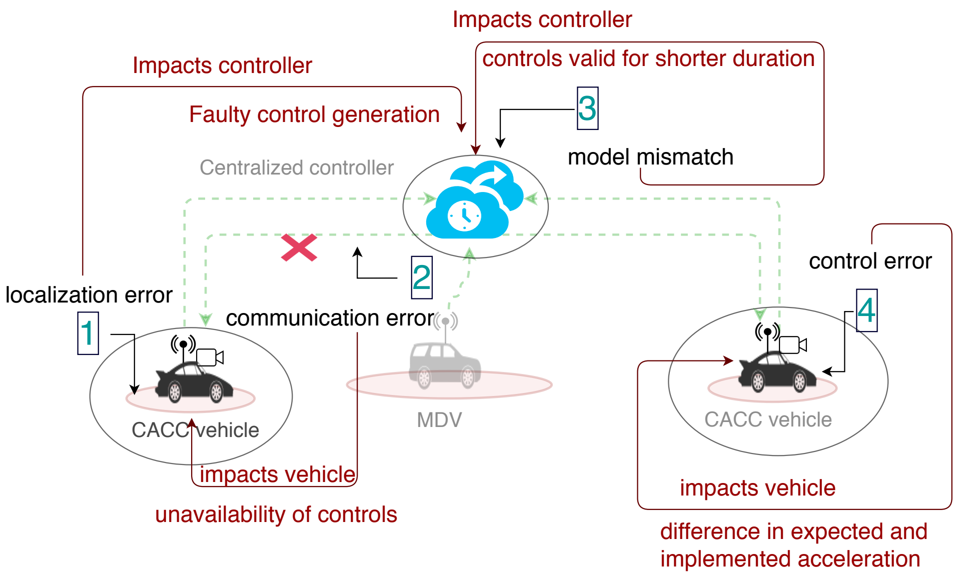


Figure 4.31: Model mismatch, control errors, communication errors and localization errors impacting centralized controller

$$\text{minimize} \quad \sum_{i=1}^{n_v} \sum_{\eta=1}^{N_p} (u_i(\eta) - u_i(\eta-1))^2 + \rho\, \epsilon \tag{4.14}$$

$$\begin{aligned} &\text{subject to} \\ &(4.10),\ (4.11),\ (4.12a),\ (4.12b),\ (3.3),\ (3.4),\ (3.8),\ (4.2), \\ &\text{Assumed MDV model: } (3.14)\ (3.15)\ (3.16)\ (3.17) \\ &(3.18),\ (3.19),\ (3.23),\ (3.24) \end{aligned}$$

First, the performance of the non-robust controller under the influence of all uncertainties listed above is evaluated. The non-robust controller doesn't counter localization errors. Next, for the same parameter values and simulation settings, the performance of the robust controller expressed by (4.14) is analyzed. The performance of the robust controller is compared with the performance of the non-robust controller and is evaluated against the perfect scenario where localization and communication errors are absent. Only a few key results from simulations are plotted to ensure clarity.

Vehicles with homogeneous localization system $\phi = 0.25$

A scenario consisting of vehicles with homogeneous localization system with a relatively low value of $\phi = 0.25$ m is evaluated first. The simulation result has been plotted in Figure 4.32.

If the performance of the robust controller is considered, the performance under MarkovGood and Bernoulli are similar to the perfect scenario (no communication or localization errors), although the jerk values a bit higher. Compared to that,

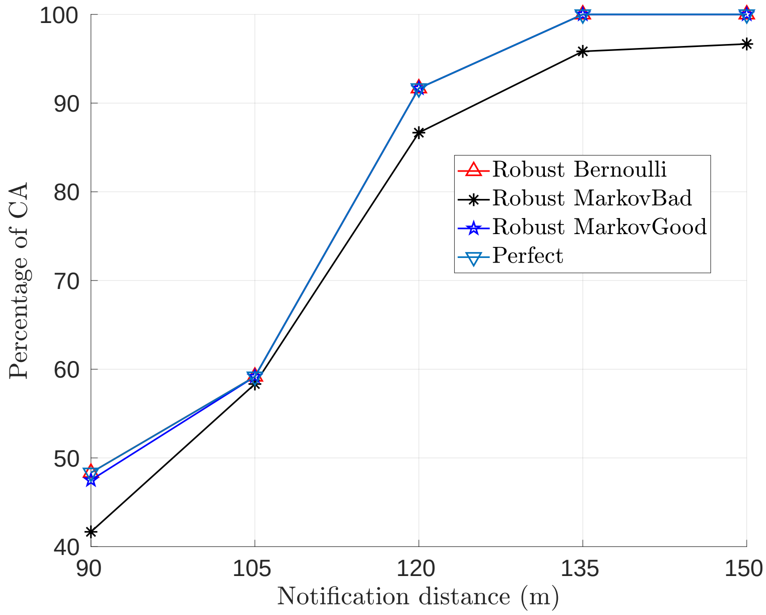


Figure 4.32: Collision avoidance statistics for robust controller under different communication models and $\phi = 0.25$ m

the performance under MarkovBad is worse by approximately 5% when collision avoidance (CA) is considered.

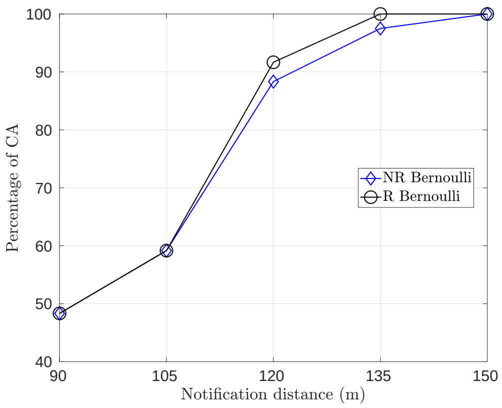
As the value of ϕ is relatively lower, even the non-robust controllers is able to avoid between 40% to 100% of the collisions (refer to Figure 4.33). The performance of the robust controller is in general better than the non-robust controller as expected.

Vehicles with homogeneous localization system $\phi = 4$

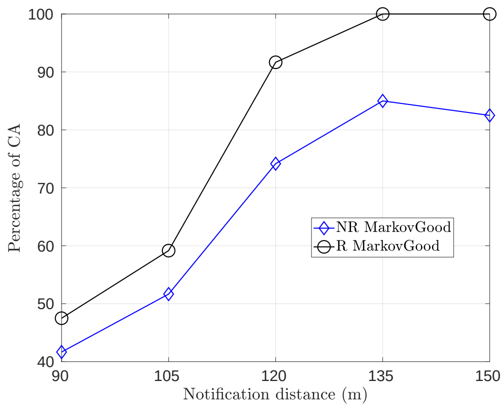
A scenario consisting of vehicles with homogeneous localization system with a relatively high value of $\phi = 4m$ is evaluated. The simulation result has been plotted in Figure 4.34. The performance of the robust controller under Bernoulli and MarkovGood is the same in terms of CA and worse under MarkovBad. Figure 4.34 resembles Figure 4.32 because the performance of the robust controller is similar in terms of CA, for the range of ϕ (0.25 m and 4 m) considered.

When the localization error is large, the performance of the non-robust controller is abysmal compared to the robust controller and sometimes is not even able to avoid a single collision as seen in Figure 4.35. Moreover, due to the combination of various types of errors, the reasons behind the performance cannot be uniquely identified. As the performance of the non-robust controller is inferior, jerks are not compared.

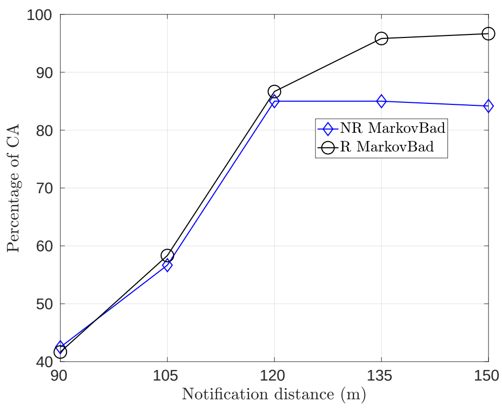
Comparing Figure 4.33 and 4.35, we can conclude that for lower values of localization error, the performance of the non-robust and the robust controller is comparable for specific models of communication error; but for high values of localization error, the performance of the robust controller is distinctly better.



(a) Bernoulli model



(b) MarkovGood model



(c) MarkovBad model

Figure 4.33: Comparing collision avoidance statistics of robust vs. non-robust controller under different communication models for $\phi = 0.25$ m

Vehicles with heterogeneous localization system

In this set of simulations, heterogeneous localization system on vehicles is assumed, ($\phi_{CACC} = 0.25$ m, $\phi_{MDV} = 4$ m). Performance of robust controller under communication channel modeled using MarkovGood and Bernoulli is similar Figure 4.36. Thus we can conclude that, despite facing a lot of random packet loss (using Bernoulli model), the performance is similar to the case where a few consecutive packet loss (burst errors) is faced. CA under MarkovBad is worse by approximately 5% for the robust controller, compared to Bernoulli/MarkovGood.

Non-robust controller performance is not close to the performance of the robust controller. It is important to note that in terms of CA, the performance of the robust controller despite communication errors modeled as Bernoulli, is as good as that of the perfect scenario (no communication or localization errors), although the discomfort values are a bit higher.

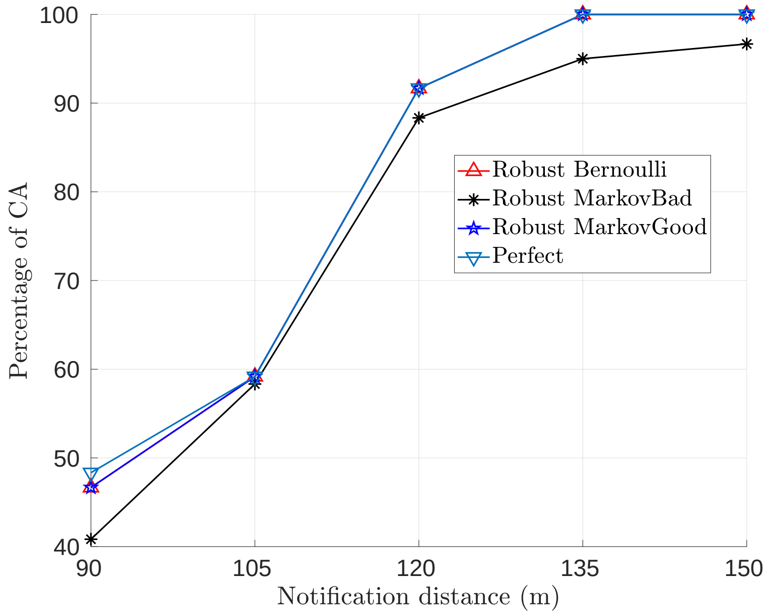


Figure 4.34: Collision avoidance statistics for robust controller under different communication channels and $\phi = 4$ m

Analyzing controller performance for communication channel modeled using Bernoulli model

Figure 4.38a shows that the performance of the non-robust controller degrades as the value of ϕ increases. However, for the robust controller, the performance remains similar and close to the perfect scenario as seen in Figure 4.38a. At the notification distance of 135 meters, 100 % collisions can be avoided.

If the communication channel is modeled using Bernoulli model, based on Figure 4.38b we can conclude that using a robust controller, we can achieve almost the same performance as that of a controller without any communication or localization errors. In other words, the robust controller is effective in mitigating these errors.

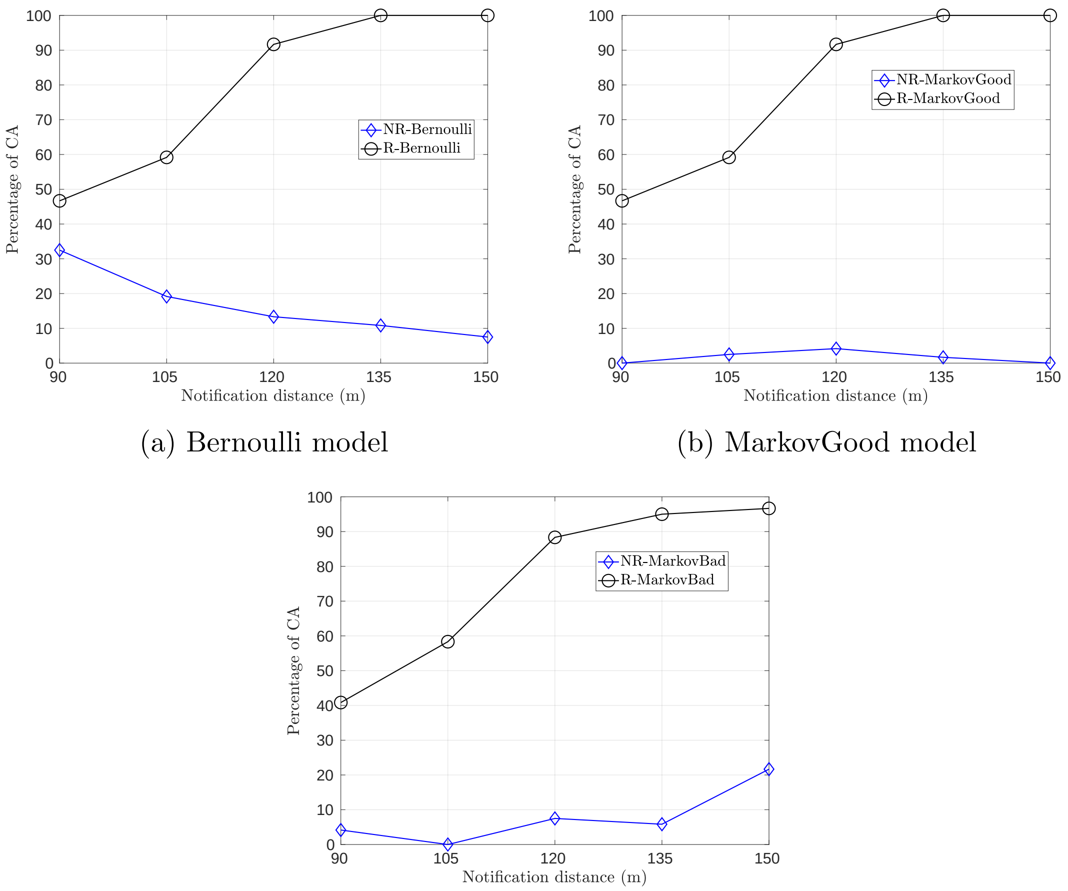
Higher ϕ, results into higher the value of jerk in general as seen in 4.39. Moreover, larger the notification distance, lower is the discomfort. The presence of localization and communication errors has increased the discomfort no matter what the notification distance (compared to perfect scenario).

Similar observations can be made for the performance of the controller for communication channel modeled using MarkovBad and MarkovGood models, and the discussion has been skipped because the trend is similar.

Comparing robust controller's performance for different localization systems

Simulation results corresponding to homogeneous localization systems with high localization error ($\phi = 4$ m) and heterogeneous localization systems ($\phi_{CACC} = 0.25$ m, $\phi_{MDV} = 4$ m) are considered.

Figure 4.40 is plotted to compare the performance of the controller for each



(a) Bernoulli model

(b) MarkovGood model

(c) MarkovBad model

Figure 4.35: Comparing collision avoidance statistics of robust vs. non-robust controller under different communication models for $\phi = 4$ m

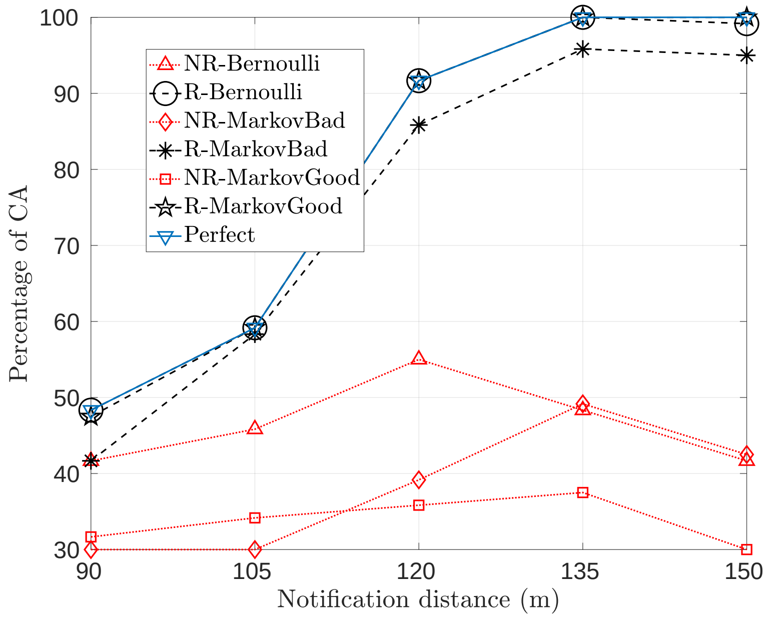


Figure 4.36: Collision avoidance statistics for heterogeneous localization systems

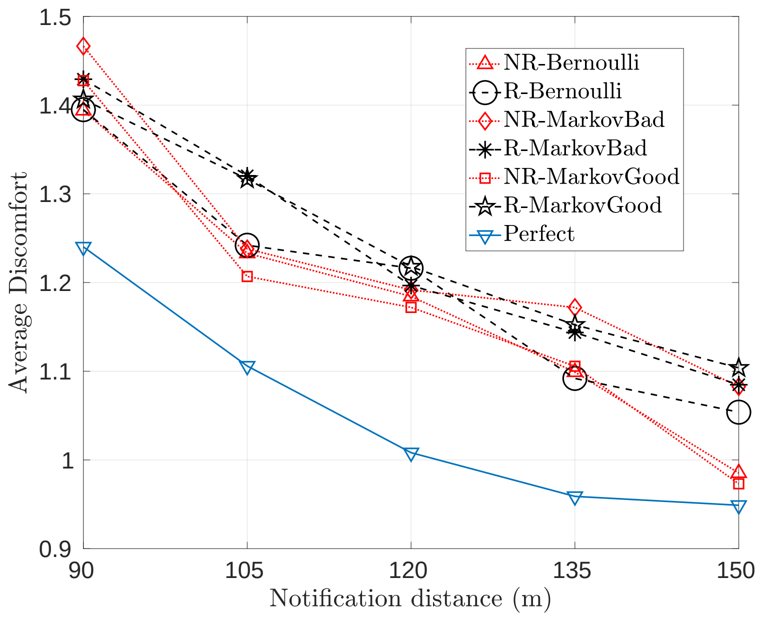
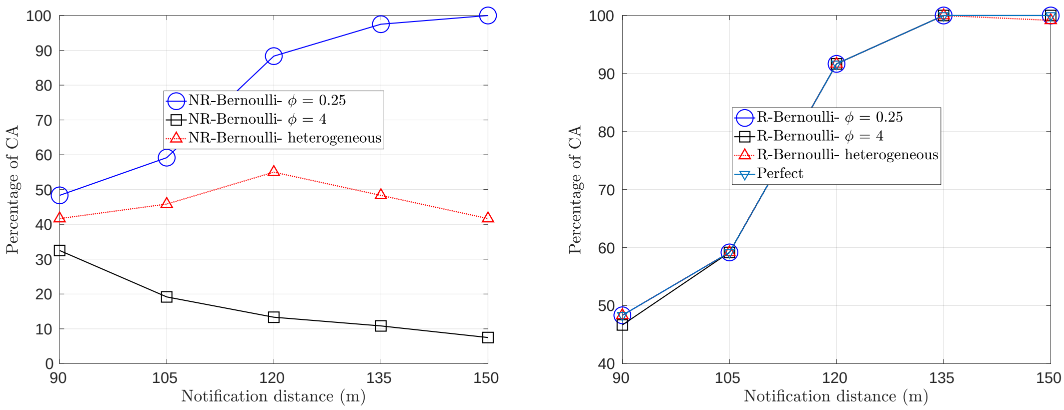


Figure 4.37: Jerk statistics for heterogeneous localization systems



(a) Evaluation of the non-robust controller (b) Evaluation of the robust controller

Figure 4.38: Comparing collision avoidance statistics for different localization systems under communication error modeled using Bernoulli model

type of communication model. If the Bernoulli model is considered, the controller performance is similar for both homogeneous and heterogeneous vehicle localization scenario. A similar observation can be drawn for MarkovBad and MarkovGood communication model. The performance of the robust controller under Bernoulli and MarkovGood communication models is similar to the perfect scenario. The percentage of CA under MarkovBad communication model is approximately 5% less compared to the perfect scenario on an average.

4.7.1 Communication Overhead

Comfort and safety in such a centralized braking system comes at the cost of additional data overhead on the downlink. For the system described in this work, N_p values per CACC vehicle are transmitted on the downlink at 10 Hz. Considering a control value as a double variable (8 Bytes), data overhead is 64 Kbps per CACC vehicle. Without loss of generality, considering a target 60 % channel load (at 6 Mbps), 3.6 Mbps data rate can theoretically be allocated, allowing approximately 56 CACC vehicles to be coordinated. As reference consider the experiment accomplished in [133], where a decentralized algorithm implemented a 25 Hz communication system with a prediction horizon of 30 slots leading to 48 Kbps of 'CAM' like data per CACC vehicle in a pure CACC traffic. The load contribution of the centralized braking application depends on the prediction horizon, number of vehicles and the required communication update frequency. This additional load may lead to communication failure. Risk analysis of collisions between vehicles because of reducing either prediction horizon or communication frequency is left to future work.

4.8 Parameter Sensitivity

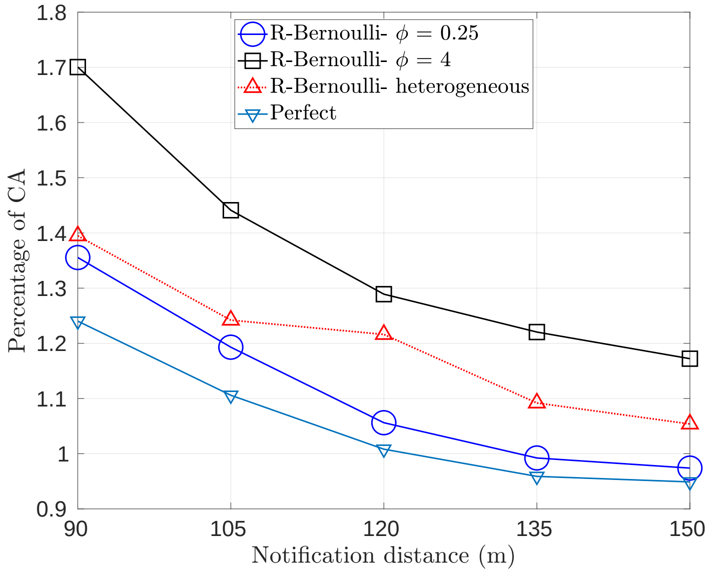


Figure 4.39: Discomfort statistics for homogeneous and heterogeneous localization systems with robust controller under communication error modeled using Bernoulli model

As the simulations performed in this work have a lot of parameters, these parameters can be changed to have simulations replicating different scenarios in real life. The results of the simulations using different communication parameters could be different. As already introduced, there are 3 critical components of the centralized control system: the localization module, the communication module and, the control module. Different methods of modeling errors in these three models and different model mismatches can give rise to different results. These are analyzed next.

Localization error is related to the uncertainty in the localization module. Different vehicles can implement different localization algorithms, example, the perception and localization module of different automated vehicles can have different performances, especially under different conditions like rain, fog, low lighting, etc. for manually driven vehicles using GPS, the variation of the performance of GPS under different environments has already been well documented. The relative position of the MDV in front can be determined by neighboring CACC vehicles using different sensors like lidars or radars, and the accuracy of localization of MDVs, in this case, will be much better, compared to the accuracy of the localization of MDVs using GPS in general. Thus, accurate information on the method using which localization is computed is essential. These methods shall have different standard deviation of localization errors which can be used for more appropriate simulations. Moreover, during real time simulations, the accuracy of the localization error estimate would be important in implementing the proposed coordinated safe braking application.

The communication channel in this work has been modeled using Bernoulli and Markov chain models to characterize a channel which has random packet losses and

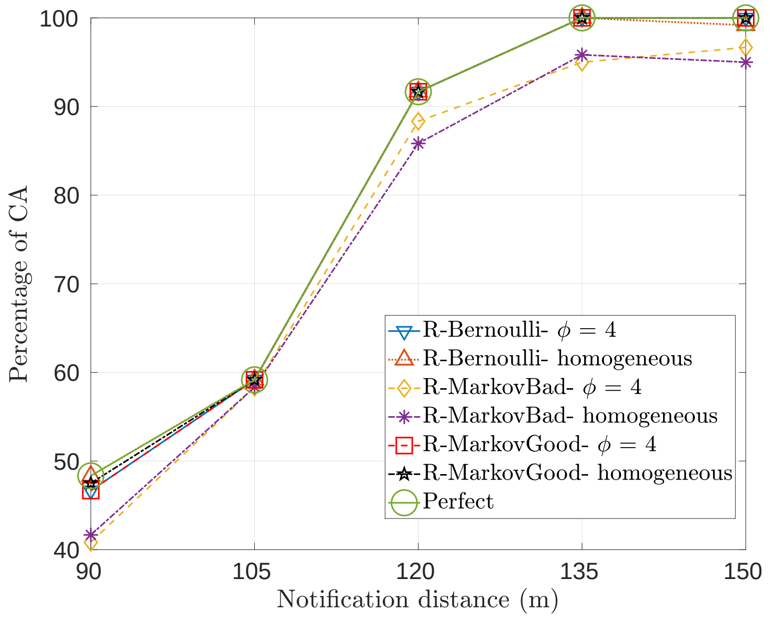


Figure 4.40: Collision avoidance statistics under different communication channels compared for homogeneous and heterogeneous localization systems with robust controller

burst errors respectively. Moreover, parameters of the Markov model have been varied such that burst packet losses of low and high magnitude can be realized. Based on real life communication channel performance, appropriate parameters for the communication models can be used. Alternately, an altogether different communication model can be used as well. Although the use of a buffer helps counter communication errors implemented using any model, the simulations can be made more realistic by using communication models suited for typical environments.

Control error in this work has been modeled as the difference in actuated control and the desired control arising due to the lower level engine behavior. The lower level engine behavior has been kept constant. In reality, different vehicles will have different engine characteristics, and thus the behavior will be different. If the type of vehicle, brand, age, and lower level engine behavior is known, that can be used. Although the computation of controls in a receding horizon fashion in MPC helps counter these control errors, effective implemented controls depend on the lower level engine behavior.

Model mismatch in this work has been implemented using two different models, first for generating predictions for MDV behavior and second for generating the actual value of the control implemented. The use of different models for generating predictions and for generating actual controls will give rise to different controls and thus different magnitudes of model mismatch. As mathematical models cannot capture human behavior perfectly, which may depend on factors like drowsiness, tiredness, alertness, etc. the use of humans in experiments to collect and use human control behavior in various circumstances is essential. To collect real human control behavior to be used as actual control, the centralized controller was interfaced with a driving simulator [29]. It would be even better if actual real-world experiments can be carried out with MDVs being driven by humans, but it may be too expensive and risky. In this case, to generate more realistic MDV control predictions, MDV behavior can be modeled online, and based on the online model controls can be generated. Although the computation of controls in a receding horizon fashion in MPC helps counter these model mismatch, the controls generated are sensitive to different kinds of mathematical models used to simulate a model mismatch.

Other parameters like the distance at which the vehicle is notified about a potential obstacle or the intent to brake (notification distance) also influence the simulations. Larger the notification distance more is the freedom to brake, and smoother can be the braking.

The impact of velocity on collision avoidance was studied in [33]. The number of collisions avoided in the presence of localization errors was seen to increase with velocity up to a certain limit. This comes from the fact that the higher velocity is, the bigger could be the inter vehicle distance (based on the time-gap inter vehicle distance policy). However, if the velocity is too high, vehicles will not be able to stop before the presumed obstacle regardless of the inter vehicle distance at given jerk restrictions.

Permitted levels of jerk is another parameter which controls the maximum change in acceleration. The permitted jerk value can be restricted either due to the capacity of the vehicle to implement a change in acceleration or alternately due to the jerk-tolerance value of the human body. In the simulations which resulted in collisions, collisions could still be avoided by adjusting the permitted jerks used in the control computation.

Similarly, maximum achievable braking strength, which is the braking strength which can be implemented by vehicles, road friction, weather conditions impacting visibility, etc. are other parameters to which the introduced system is sensitive; these factors have not been considered in this work.

To summarize, this chapter begins with an introduction to various configurable parameters of the centralized controller and explains the simulation. The implementation of the buffer was explained not only mathematically but also with graphically. Next, the chapter introduces robust and non-robust controllers to different types of errors and evaluates their performance. First, the impact of model mismatch was evaluated on the robust and non-robust controller, using different kinds of models, the model which has the most realistic behavior was chosen. The experiment where a driving simulator was interfaced with a centralized controller to simulate a mixed vehicle braking scenario was explained. The impact of different ways of modeling control errors was evaluated, and the model which represents the lower level engine behavior was chosen for further simulations. The next section introduces different ways of modeling communication error and different ways of countering communication error. The impact of communication error alongside inherent model mismatch and control errors, on the robust centralized controller was also evaluated. Controllers robust to localization error was introduced, and the performance of the robust and the non-robust controller was evaluated. An application of these simulations can be suggestions for enhanced safety based on the simulation results was also explained. In the end, the impact of all errors together on the centralized controller is evaluated and analyzed. The last part of the chapter goes through the parameters to which the algorithm is sensitive and changing these parameters would result into different results.

Chapter 5

Conclusions and Perspectives

After the simulations and evaluations, this chapter summarizes and relists important conclusions. Different perspectives left for future work is discussed in the second part of this chapter. To close the chapter, contributions to this topic and publications are summarized.

5.1 Conclusions

This thesis proposed a model predictive control based centralized controller which computes controls for CACC vehicle braking among manually driven vehicles in a receding horizon fashion. The controller has been made robust to different errors like packet losses, localization uncertainties, lower level engine behavior, and inherent model mismatch.

We list some of the key conclusions from the thesis:

- A centralized controller may have to intervene a braking procedure in a mixed vehicle scenario consisting of autonomous and manually driven vehicles
- An autonomous vehicle will have different modules like a localization module, a communication module, and a control module
- Errors originating in any of those modules can produce undesired consequences and also lead to collisions
- A robust MPC controller can be implemented in a receding horizon fashion to counter uncertainties and mitigate the impact of different errors impacting the centralized control operation
- From section 4.3, we conclude that model mismatch is an inherent property of the *predictive* controller used in this work. Model mismatch can lead to the generation of controls which if not updated, can result in collisions. The magnitude of model mismatch depends on the model used for prediction and the actual control model of MDVs. It will be different for different human drivers. Section 4.4 demonstrates a way of considering the lower level engine behavior (modeled as control error) and including it in simulations. Considering lower level controller behavior is necessary for completeness and realisticness of the

simulations. Model mismatch and control errors cannot be eliminated, instead can only be mitigated by frequent control computation using updated state parameters. Receding horizon based MPC is one of the ways to mitigate the impact of model mismatch and lower level control behavior.

- Section 4.5 introduces different communication models and different fallback strategies when packets are not received. Simulations help us draw the following conclusions: communication errors realized using Bernoulli model results into random packet losses whereas the use of Markov model results into multiple consecutive packet losses in the form of bursts of errors. By adjusting the parameters of the Markov model, the percentage of packet loss can be changed.

  The impact of consecutive packet loss modeled as Markov model is more severe and results into more collisions compared to random packet losses implemented using Bernoulli model, despite having approximately 10% lower packet losses with the former model. It is found that fall back to ACC helps avoid more number of collisions compared to other fall back strategies like the use of a buffer or retaining previous acceleration when the communication channel is poor and is facing multiple consecutive packet losses. This is because when the vehicle falls back to ACC in case of communication loss, there is a change in acceleration from the applied acceleration based on control received from the centralized controller to the control derived using a mathematical model of ACC. After that, controls are continuously computed based on the ACC model, using the relative distance and velocity of the vehicle in the front. Thus the following conclusions can be drawn: 1. there is continuous computation when there is a fallback to ACC which usually ensures collision avoidance 2. there might be a significant change in acceleration during the switch to the fallback strategy and back to the CACC mode, leading to high jerk values. Because of these reasons, fall back to ACC is beneficial only during extensive periods of consecutive packet loss; switching back and forth to ACC-CACC will only make the ride more uncomfortable, and the lower level engine controller may not be able to implement desired controls which can ensure collision avoidance. This validates Ploeg's claim that in the case of poor communication, it is advised not to use communication techniques at all for CACC algorithms and instead switch to ACC algorithms [100, 106].

  Under all other cases when the communication channel is realized using Bernoulli model (with approximately 50% packet loss) or Markov model (with approximately 2% packet loss), the performance of the buffer based fallback strategy was the best, not only in terms of collision avoidance but also in terms of discomfort.

- Section 4.6 introduces the proposed method to model localization errors and to make the controller robust to localization errors. In general, the simulation results lead us to the conclusion that the performance of the Robust MPC controller is similar to the performance of the MPC controller under perfect localization information and it is far better than the performance of the non-robust controller. The difference in the percentage of collision avoidance between the robust and non-robust controller is more when the localization error is larger. In terms of discomfort, larger the standard deviation of localization error, larger can be the change in localization error which results into changes in the area over which the vehicle could be located, and this results into larger changes in controls. Thus, larger localization error usually results in more discomfort. As apparent, discomfort decreases and the percentage of collisions avoided increases as the notification distance increases.

- Section 4.7 simulates and evaluates the proposed robust controller under all kinds of errors namely: localization, communication, model mismatch, and control error. At low standard deviation of localization error, irrespective of the communication error, the performance of the robust controller is similar to the case where there is neither localization nor communication error. The performance of the non-robust controller is similar to the robust controller when random packet loss is assumed and is slightly worse when consecutive packet loss is assumed.

  At high standard deviation of localization error, irrespective of the communication error, the performance of the robust controller is similar to the case where there is neither localization nor communication error. The non-robust controller is not able to avoid a significant number of collisions, and the performance of the non-robust controller is abysmal compared to the robust controller, irrespective of the communication model used. Because of the complexity of the system and the different types of errors, it is difficult to single out the reason behind the performance of the non-robust controller.

  Although at an additional cost of increased discomfort, in general, we can conclude that the robust controller can perform almost as well under different sources of errors, as a legacy (non-robust) MPC controller without any errors.

- Interfacing the developed algorithm with a driving simulator is one of the ways of including human drivers in the experiments. It not only allows us to check whether the controller adapts correctly with the behavior of other human drivers but also to verify whether human drivers can coexist on the same road as autonomous vehicles. Theoretical Matlab based simulations usually give optimistic results compared to experiments; the real-time performance of the developed algorithm may not be optimal and thus needs to be validated using driving simulators or real-life experiments.

5.2 Perspectives

Next, a list of potential future work and perspectives is summarized.

- The introduced algorithm only focusses on longitudinal collision avoidance as control of vehicles on a single lane is considered. The developed algorithm can be expanded to a multi-vehicle multi-lane safe braking problem. One way of resolving this issue is to consider multiple single lane safe brake problems, but this would not be optimal. To ensure maximal collision avoidance, it becomes necessary to consider the global scenario with multiple vehicles coming to a halt. This necessitates longitudinal and lateral control both to ensure collision free braking. Moreover, this problem will even allow lane changing and possibility to overtake other vehicles which introduces another level of complexity. This multi-lane problem appears to be a non-convex problem and thus can be challenging to solve in real time. For situations like this, it might be necessary to switch to distributed or decentralized control strategies. This is one of the future work which is of particular interest.

- The number of collisions avoided in these simulations is derived for a fixed value of the maximum permitted change in acceleration between two time slot. This is to ensure the change in acceleration is within the activation limit of the controller and the jerk tolerance capacity of the human. Due to this jerk limitation, control computations may return infeasible. Jerk can be put as a slack variable (a soft constraint in place of a hard constraint) to avoid cases of computational infeasibility. It shall make sure that the computations complete successfully each time. One of the issues of this approach is the feasibility of implementation of the controls computed as it may or may not be feasible depending on the change in acceleration compared to the last time slot (due to engine constraints).

- Communication errors in a centralized controller can take place on either the uplink or the downlink. This work only focusses on downlink communication failures. Uplink communication failures need to be integrated, and the performance of the controller needs to be verified. Localization error at different time slots is generated using normal distribution and parameters mentioned in section 3.2.3. This results into values of localization error at consecutive time slots being hugely different. In future, these localization errors need to be smoothened to have a more realistic distribution of localization error.

- This thesis only focussed on a multi vehicle braking scenario, but other road traffic conditions like roundabouts, ramp mergings, etc. should be addressed next. Although it involves the development of entirely different control algorithms, the general idea of coordination of vehicles remains.

- The algorithm developed in this thesis focusses on a set number of vehicles (taken at a snapshot) and optimization of the controls of these vehicles until they come to a halt before reaching the intersection of the obstacle. However, as vehicles continuously approach the intersection one after another in traffic, the true scenario represents a continuous operation with a varying number of vehicles. When a new vehicle enters the communication range (of the RSU at the intersection or the DENM transmission range of the broken vehicle), the number of vehicles whose control needs to be optimized increases. Once any vehicle comes to a halt, its controls are no longer required to be optimized, and thus that vehicle and related computation can be dropped from the optimization problem. Another exciting aspect is the location before which the first vehicle in the stream of moving vehicles must come to a halt keeps changing. At the start of the optimization problem, the first vehicle must come to a halt before reaching the obstacle or the intersection. When the first vehicle has come to a halt, other vehicles in the stream must come to a halt before reaching the rear end position of that vehicle. Thus, the terminal position constraint of the optimization problem changes as well. This issue is challenging as it requires optimization parameters to be changed dynamically and frequently, example: the number of vehicles, parameters related to vehicles, terminal position constraints, etc.

- One of the issues faced during implementation of the developed algorithm was that the total delay (computational delay and communication delay to transfer control information from the driving simulator to the centralized controller) was longer than expected. This results in undesired consequences and even collisions as explained in Section 4.3.1. It is thus imperative to improve the computational speed, or more appropriate solvers need to be used.

- As introduced previously, the proposed system has many parameters. Values to these parameters can be tuned, and the simulation results are thus sensitive to different parameters as introduced in Section 4.8. Going forward, it would be essential to have simulation results with different velocity, inter vehicle distance, etc. to evaluate the system is the most sensible to which parameter. Moreover, there are different ways of modeling different errors which represent different scenarios. Simulations adapted to particular scenarios with particular vehicle types and characteristics need to be performed.

- Warning for enhanced safety [33]: A database can be generated from simulation results which contains the probability of collisions for a particular set of average velocity and distance between vehicles, as introduced in Section 4.6.3. An application running on the RSU can use the database to provide suggestions to reduce the probability of collisions by changing either the velocity or distance with the vehicle in front or both. The impact of such a passive safety monitoring application needs to be verified.

To summarize, the first part of this chapter highlighted different conclusions from various simulations, and the next part highlighted options to improve and extend this work. Next, a short list of contributions which have been published is listed.

Appendix A

List of Publications and Contributions

This thesis contributes to the literature by evaluating the impact of one or multiple errors influencing the centralized control operation. Novel ideas to counter these errors, mitigate them and the uncertainties are proposed. These proposed ideas are implemented and evaluated as a controller robust to those errors. The key contributions are as follows:

- We device a MPC based controller specific to the case of braking for autonomous vehicles
- We provide a framework to integrate different kinds of errors into the MPC based framework
- Different kinds of errors are modeled and integrated into the MPC framework to simulate the impact of these errors on a centralized controller, like in a real world scenario
- We propose solutions to counter these errors and make the controller robust
- The performance of the robust controller is compared to the non-robust controller

Different parts of these contributions can be found in various publications. The first paper [30] introduces a braking strategy for an autonomous vehicle in a mixed vehicle scenario to avoid front-end and rear-end collisions. During the early deployment phase of autonomous vehicles, autonomous vehicles will share roads with conventional manually driven vehicles. The paper focusses on a scenario where an autonomous vehicle is being followed by a manually driven vehicle, both these vehicles have to come to a halt before reaching the intersection. The challenge was to produce a braking strategy such that the autonomous vehicle doesn't brake too hard, else the following vehicle would collide with it; at the same time, it can not brake too slow, else it will not be possible for the autonomous vehicle to come to a halt before the obstacle in front. The paper proposes a heuristic way of ensuring

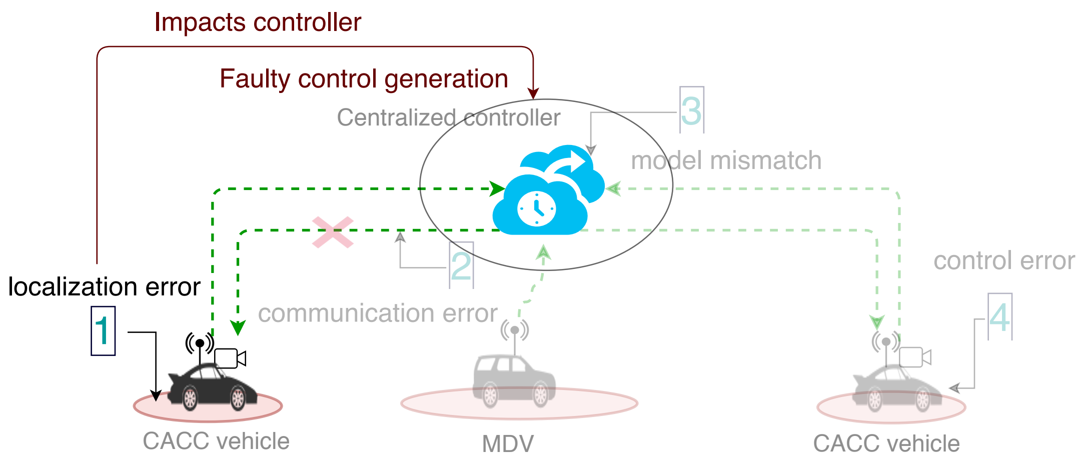


Figure A.1: Localization error impacting centralized controller

collision avoidance at both ends for an autonomous vehicle. This approach is computationally inexpensive and quick, but the issue is, it would be difficult to extend it for more than two vehicles.

This was the motivation to look for alternate strategies to solve the multi vehicle braking problem. A centralized collision mitigation strategy based on model predictive strategy was proposed which can handle more than two vehicles and ensure collision avoidance while braking [32]. This work considers various human factors like perception response time, limited visibility and limited jerk sustainability which contribute to collisions. The relation between percentage of collisions avoided and the percentage penetration of autonomous vehicles is also studied.

Such a centralized braking algorithm usually has different dependencies. [31] accounts for the impact of localization errors (Figure A.1) on the centralized controller. The performance of the centralized centralized control algorithm is poor and results into collisions when localization errors are not accounted for. An algorithm to counter localization errors has been proposed and the performance of the controller has been evaluated. The drawback of this approach is that the vehicles are assumed to occupy a larger area than they really do, resulting into a reduced road traffic throughput.

The impact of countering localization errors as introduced in [31], on the flow capacity has been studied in [33]. The simulated data can be stored in form of collision occurrence for the pair of average velocity of the vehicles and average distance between vehicles. This data can be used to transmit suggestions like reduce velocity or increase inter-vehicle distance, to the vehicles as a safety precaution to reduce the chances of a potential collision for an acceptable value of probability of collision.

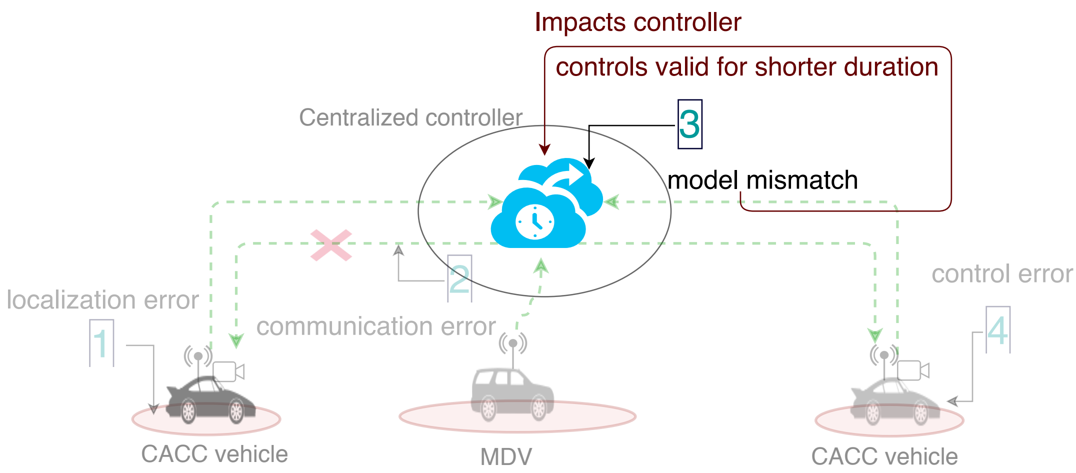


Figure A.2: Model mismatch impacting centralized controller

The impact of model mismatch (Figure A.2), another type of error arising due to the difference in the assumed and the actual trajectory of the manually driven vehicle is evaluated next [29]. The proposed algorithm was implemented on matlab and interfaced with a driving simulator. The mixed vehicle simulation with a human diver and an autonomous vehicle is performed with different participants having multiple opportunities to have the feeling of driving on the same road as autonomous vehicles. The human driver reacts to the vehicle in front and the controls are extracted using the driving simulator which is connected to the matlab based centralized controller. Compared to theoretical simulations, experiments with human drivers showed a decrease in collisions avoided by 46 % and an increase in discomfort by 91 %. Issues like communication and computational delays were found to be one of the reasons for the degraded performance.

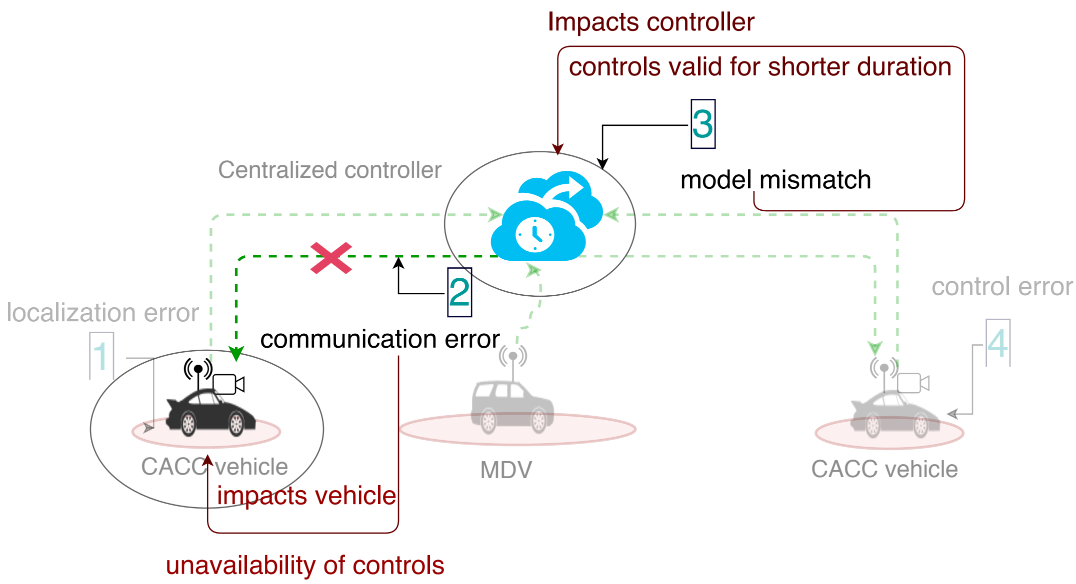


Figure A.3: Model mismatch and communication error impacting centralized controller

Communication errors like packet losses and delays can influence a centralized controller and the impact of communication errors in presence of model mismatch (Figure A.3) on the centralized controller operation has been analyzed in [34]. The use of a buffer is proposed to store future controls computed using MPC and these controls are to be used when fresh controls from the centralized controller are not available due to communication errors.

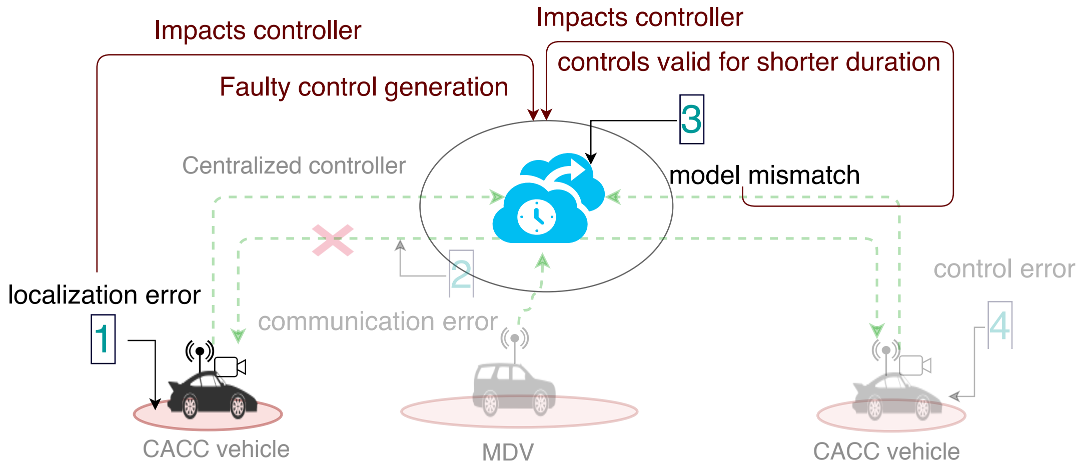


Figure A.4: Localization error and model mismatch impacting centralized controller

The impact of model mismatch and localization errors (Figure A.4) on a centralized controller is evaluated in [35]. First the impact of model mismatch is studied. Control recomputations using MPC mitigates the impact of model mismatch. Next, the performance of the robust buffer aided controller and a non-robust controller are evaluated under the influence of model mismatches and localization errors. The performance of the robust controller despite different errors can be similar to that of the non-robust controller without any errors.

List of publications can be found next:

[30] R. H. Patel, J. Härri, and C. Bonnet, “Braking Strategy for an Autonomous Vehicle in a Mixed Traffic Scenario,” in Proceedings of the 3rd International Conference on Vehicle Technology and Intelligent Transport Systems - Volume 1: VEHITS, INSTICC. SciTePress, 2017, pp. 268275.

[32] R. H. Patel, J. Härri, and C. Bonnet, “A Collision Mitigation Strategy for Intelligent Vehicles to compensate for Human Factors Affecting Manually Driven Vehicles,” in 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Oct 2017, pp. 114119.

[31] R. H. Patel, J. Härri, and C. Bonnet, ”Accounting for localization errors in a mixed-vehicle centralized control system,” in Mobil TUM 2017, International Scientific Conference on Mobility and Transport, July 4-5, 2017, Munich,

Germany, 06 2017, available at http://www.eurecom.fr/fr/people/patel-raj-haresh-1/publications.

[33] R. H. Patel, J. Härri, and C. Bonnet, “Impact of Localization Errors on Automated Vehicle Control Strategies,” in Vehicular Networking Conference (VNC), 2017 IEEE. IEEE, 2017, pp. 6168.

[29] M. Aramrattana, R. H. Patel, C. Englund, J. Härri, J. Jansson, and C. Bonnet, “Evaluating Model Mismatch Impacting CACC Controllers in Mixed Traffic using a Driving Simulator,” Intelligent Vehicles Symposium (IV), IEEE, 2018, available at: http://www.eurecom.fr/fr/people/patel-raj-haresh-1/publications.

[34] R. H. Patel, J. Härri, and C. Bonnet, “Centralized Model Predictive CACC Control Robust to Burst Communication Errors,” Vehicular Technology Conference (VTC-fall), IEEE, 2018, available at: http://www.eurecom.fr/fr/people/patel-raj-haresh-1/publications.

[35] R. H. Patel, H. Wymeersch, J. Härri, and C. Bonnet, “Buffer-Aided Model Predictive Controller to Mitigate Model Mismatches and Localization Errors, IEEE Transactions on Intelligent Vehicles, vol. 3, no. 4, pp. 501-510, Dec2018.

Bibliography

[1] "Britain's average daily commute is one hour and 38 minutes as poll reveals we love listening to music on our journeys, dislike smelly passengers and think the transport is too expensive," [Online; accessed 19-July-2018]. [Online]. Available: http://www.dailymail.co.uk/news/article-3335945/Britain-s-average-daily-commute-ONE-HOUR-38-MINUTES.html

[2] B. Van Wee, P. Rietveld, and H. Meurs, "Is average daily travel time expenditure constant? in search of explanations for an increase in average travel time," *Journal of transport geography*, vol. 14, no. 2, pp. 109–122, 2006.

[3] G. Cookson and B. Pishue, "Inrix global traffic scorecard," *INRIX Research, February*, 2017. [Online]. Available: http://inrix.com/scorecard/

[4] F. Guerrini, "Traffic congestion costs americans $ 124 billion a year, report says," *Forbes, October*, vol. 14, 2014.

[5] "Smog, soot, and other air pollution from transportation," [Online; accessed 17-July-2018]. [Online]. Available: https://www.epa.gov/transportation-air-pollution-and-climate-change/smog-soot-and-local-air-pollution

[6] R. D. Brook, B. Franklin, W. Cascio, Y. Hong, G. Howard, M. Lipsett, R. Luepker, M. Mittleman, J. Samet, S. C. Smith *et al.*, "Air pollution and cardiovascular disease: a statement for healthcare professionals from the expert panel on population and prevention science of the american heart association," *Circulation*, vol. 109, no. 21, pp. 2655–2671, 2004.

[7] S. A. Stansfeld and M. P. Matheson, "Noise pollution: non-auditory effects on health," *British medical bulletin*, vol. 68, no. 1, pp. 243–257, 2003.

[8] B. W. Smith, "Human error as a cause of vehicle crashes," *Center for Internet and Society, http://cyberlaw. stanford. edu/blog/2013/12/human-error-cause-vehicle-crashes*, 2013.

[9] "Nsc motor vehicle fatality estimates," [Online; accessed 17-July-2018]. [Online]. Available: https://www.nsc.org/Portals/0/Documents/NewsDocuments/2018/December_2017.pdf

[10] U.S. Department of Transportation, "Adas features expand role, v2v looms," [Online; accessed 20-July-2018]. [Online]. Available: https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety

[11] E. ETSI, "Intelligent transport services for road (its)," *Vehicular Communications.*

[12] I. Rashdan, F. de Ponte Muller, and S. Sand, "Analysis of communication requirements for cacc in stop-and-go behavior for energy efficient driving," in *Vehicular Technology Conference (VTC-Fall), 2016 IEEE 84th.* IEEE, 2016, pp. 1–5.

[13] N. Kaempchen, B. Schiele, and K. Dietmayer, "Situation assessment of an autonomous emergency brake for arbitrary vehicle-to-vehicle collision scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 4, pp. 678–687, 2009.

[14] R. A. Cunard, G. W. Davis, P. Demosthenes *et al.*, "Intersection safety issue briefs," *Washington, DC: ITE (Institute of Traffic Engineers, US)/FHWA (Federal Highway Administration, US) Publication*, pp. 20–22, 2004.

[15] I. T. S. Data and A. Group, "Road safety annual report 2013," *International Traffic Forum*, 2013.

[16] T. Zhang and L. Delgrossi, *Vehicle safety communications: protocols, security, and privacy.* John Wiley & Sons, 2012, vol. 103.

[17] Wikipedia, "Multiple-vehicle collision," [Online; accessed 25-July-2018]. [Online]. Available: https://ec.europa.eu/inea/sites/inea/files/2017_blending_its.pdf

[18] N. Sugiyama and T. Nagatani, "Multiple-vehicle collision in traffic flow by a sudden slowdown," *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 8, pp. 1848–1857, 2013.

[19] R. Ramakers, K. Henning, S. Gies, D. Abel, and H. M. A. Max, "Electronically coupled truck platoons on german highways," in *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, Oct 2009, pp. 2409–2414.

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

[21] A. Gattami, A. Al Alam, K. H. Johansson, and C. J. Tomlin, "Establishing safety for heavy duty vehicle platooning: A game theoretical approach," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 3818–3823, 2011.

[22] R. Rajamani and C. Zhu, "Semi-autonomous adaptive cruise control systems," *IEEE Transactions on Vehicular Technology*, vol. 51, no. 5, pp. 1186–1192, 2002.

[23] Q. Xu and R. Sengupta, "Simulation, analysis, and comparison of acc and cacc in highway merging control," in *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE.* IEEE, 2003, pp. 237–242.

[24] N. H. T. S. Administration *et al.*, "Preliminary statement of policy concerning automated vehicles," *Washington, DC*, pp. 1–14, 2013.

[25] ISO, "Iso 26262: Road vehicles-functional safety," *International Standard ISO/FDIS*, 2011.

[26] S. O.-R. A. V. S. Committee *et al.*, "Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems," 2018.

[27] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[28] P. Daviet and M. Parent, "Longitudinal and lateral servoing of vehicles in a platoon," in *Intelligent Vehicles Symposium, 1996., Proceedings of the 1996 IEEE*, Sep 1996, pp. 41–46.

[29] M. Aramrattana, R. H. Patel, C. Englund, J. Härri, J. Jansson, and C. Bonnet, "Evaluating Model Mismatch Impacting CACC Controllers in Mixed Traffic using a Driving Simulator," *Intelligent Vehicles Symposium (IV), IEEE*, 2018, available at: http://www.eurecom.fr/fr/people/patel-raj-haresh-1/publications.

[30] R. H. Patel, J. Härri, and C. Bonnet, "Braking Strategy for an Autonomous Vehicle in a Mixed Traffic Scenario," in *Proceedings of the 3rd International Conference on Vehicle Technology and Intelligent Transport Systems - Volume 1: VEHITS,*, INSTICC. SciTePress, 2017, pp. 268–275.

[31] R. H. Patel, J. Härri, and C. Bonnet, "Accounting for localization errors in a mixed-vehicle centralized control system," in *Mobil TUM 2017, International Scientific Conference on Mobility and Transport, July 4-5, 2017, Munich, Germany*, 06 2017, available at http://www.eurecom.fr/fr/people/patel-raj-haresh-1/publications.

[32] R. H. Patel, J. Hrri, and C. Bonnet, "A Collision Mitigation Strategy for Intelligent Vehicles to compensate for Human Factors Affecting Manually Driven Vehicles," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Oct 2017, pp. 114–119.

[33] R. H. Patel, J. Härri, and C. Bonnet, "Impact of Localization Errors on Automated Vehicle Control Strategies," in *Vehicular Networking Conference (VNC), 2017 IEEE*. IEEE, 2017, pp. 61–68.

[34] R. H. Patel, J. Härri, and C. Bonnet, "Centralized Model Predictive CACC Control Robust to Burst Communication Errors," *Vehicular Technology Conference (VTC-fall), IEEE*, 2018, available at: http://www.eurecom.fr/fr/people/patel-raj-haresh-1/publications.

[35] R. H. Patel, H. Wymeersch, J. Härri, and C. Bonnet, "Buffer-Aided Model Predictive Controller to Mitigate Model Mismatches and Localization Errors," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 4, pp. 501–510, Dec 2018.

[36] C. Bonnet and H. Fritz, "Fuel consumption reduction in a platoon: Experimental results with two electronically coupled trucks at close spacing," SAE Technical Paper, Tech. Rep., 2000.

[37] B. Van Arem, C. J. Van Driel, and R. Visser, "The impact of cooperative adaptive cruise control on traffic-flow characteristics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 429–436, 2006.

[38] S. Santini, A. Salvi, A. Valente, A. Pescape, M. Segata, and R. L. Cigno, "A consensus-based approach for platooning with inter-vehicular communications," in *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2015, pp. 1158–1166.

[39] S. Ashley, "Robot truck platoons roll forward," [accessed 19-March-2016]. [Online]. Available: http://www.bbc.com/future/story/20130409-robot-truck-platoons-roll-forward

[40] R. Ramakers, K. Henning, S. Gies, D. Abel, and H. M. A. Max, "Electronically coupled truck platoons on german highways," in *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, Oct 2009, pp. 2409–2414.

[41] S. team, "The sartre project," [accessed 19-March-2016]. [Online]. Available: http://www.sartre-project.eu/en/Sidor/default.aspx

[42] J. Mårtensson, M. Nybacka, J. Jerrelind, and L. Drugge, "Evaluation of safety distance in vehicle platoons by combined braking and steering," in *Proceedings of 11th International Symposium on Advanced Vehicle Control, Sept 9-12, 2012, Seoul, Korea :*. Japan Society of Mechanical Engineers (JSAE), 2012, qC 20130611.

[43] Y. Z. C. C. Chien and C. Y. Cheng, "Autonomous intelligent cruise control using both front and back information for tight vehicle following maneuvers," in *American Control Conference, Proceedings of the 1995*, vol. 5, Jun 1995, pp. 3091–3095 vol.5.

[44] C. Appert-Rolland and L. Santen, "Accidents in platoons of vehicles," in *Traffic and Granular Flow05*. Springer, 2007, pp. 623–631.

[45] X.-Y. Lu, J. Wang, S. E. Li, and Y. Zheng, "Multiple-vehicle longitudinal collision mitigation by coordinated brake control," *Mathematical Problems in Engineering*, vol. 2014, 2014.

[46] X.-Y. Lu and J. K. Hedrick, "Practical string stability for longitudinal control of automated vehicles," *Vehicle system dynamics*, vol. 41, pp. 577–586, 2004.

[47] V. Milanés and S. E. Shladover, "Modeling cooperative and autonomous adaptive cruise control dynamic responses using experimental data," *Transportation Research Part C: Emerging Technologies*, vol. 48, pp. 285–300, 2014.

[48] S. Öncü, N. Van De Wouw, and H. Nijmeijer, "Cooperative adaptive cruise control: Tradeoffs between control and network specifications," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on.* IEEE, 2011, pp. 2051–2056.

[49] S. nc, J. Ploeg, N. van de Wouw, and H. Nijmeijer, "Cooperative adaptive cruise control: Network-aware analysis of string stability," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 4, pp. 1527–1537, Aug 2014.

[50] M. Segata, R. L. Cigno, H. M. M. Tsai, and F. Dressler, "On platooning control using ieee 802.11p in conjunction with visible light communications," in *2016 12th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, Jan 2016, pp. 1–4.

[51] A. Ali, G. Garcia, and P. Martinet, "Safe platooning in the event of communication loss using the flatbed tow truck model," in *Control Automation Robotics & Vision (ICARCV), 2014 13th International Conference on.* IEEE, 2014, pp. 1644–1649.

[52] A. Scheuer, O. Simonin, and F. Charpillet, "Safe longitudinal platoons of vehicles without communication," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, May 2009, pp. 70–75.

[53] D. Pan and Y. Zheng, "Velocity difference control based on dynamic tracking of safe following distance in the process of vehicle following," *IET Intelligent Transport Systems*, vol. 8, no. 3, pp. 232–243, May 2014.

[54] P. Li, L. Alvarez, and R. Horowitz, "Ahs safe control laws for platoon leaders," *IEEE Transactions on Control Systems Technology*, vol. 5, no. 6, pp. 614–628, Nov 1997.

[55] R. Miller and Q. Huang, "An adaptive peer-to-peer collision warning system," in *Vehicular Technology Conference, 2002. VTC Spring 2002. IEEE 55th*, vol. 1, 2002, pp. 317–321 vol.1.

[56] Y. Zhang, E. K. Antonsson, and K. Grote, "A new threat assessment measure for collision avoidance systems," in *2006 IEEE Intelligent Transportation Systems Conference*, Sept 2006, pp. 968–975.

[57] A. Cabrera, S. Gowal, and A. Martinoli, "A new collision warning system for lead vehicles in rear-end collisions," in *2012 IEEE Intelligent Vehicles Symposium*, June 2012, pp. 674–679.

[58] A. Ferrara and C. Vecchio, "Collision avoidance strategies and coordinated control of passenger vehicles," *Nonlinear Dynamics*, vol. 49, no. 4, pp. 475–492, 2007. [Online]. Available: http://dx.doi.org/10.1007/s11071-006-9110-4

[59] S. K. Gehrig and F. J. Stein, “Collision avoidance for vehicle-following systems,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 233–244, June 2007.

[60] T. C. Ng, J. I. Guzman, and M. D. Adams, “Autonomous vehicle-following systems : a virtual trailer link model,” in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Aug 2005, pp. 3057–3062.

[61] M. Treiber, A. Hennecke, and D. Helbing, “Congested traffic states in empirical observations and microscopic simulations,” *Physical review E*, vol. 62, no. 2, p. 1805, 2000.

[62] A. Kesting, M. Treiber, and D. Helbing, “Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity,” *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 368, no. 1928, pp. 4585–4605, 2010.

[63] R. Krajewski, P. Themann, and L. Eckstein, “Decoupled cooperative trajectory optimization for connected highly automated vehicles at urban intersections,” in *Intelligent Vehicles Symposium (IV), 2016 IEEE*. IEEE, 2016, pp. 741–746.

[64] J.-Q. Wang, S. E. Li, Y. Zheng, and X.-Y. Lu, “Longitudinal collision mitigation via coordinated braking of multiple vehicles using model predictive control,” *Integrated Computer-Aided Engineering*, vol. 22, no. 2, pp. 171–185, 2015.

[65] Q. Chao, Z. Deng, and X. Jin, “Vehicle–pedestrian interaction for mixed traffic simulation,” *Computer Animation and Virtual Worlds*, vol. 26, no. 3-4, pp. 405–412, 2015.

[66] C. Roncoli, I. Papamichail, and M. Papageorgiou, “Model predictive control for motorway traffic with mixed manual and vacs-equipped vehicles,” *Transportation Research Procedia*, vol. 10, pp. 452–461, 2015.

[67] K. Yang, S. I. Guler, and M. Menendez, “Isolated intersection control for various levels of vehicle technology: Conventional, connected, and automated vehicles,” *Transportation Research Part C: Emerging Technologies*, vol. 72, pp. 109–129, 2016.

[68] J. Monteil, J. Sau, and M. Bouroche, “Adaptive pid feedback control for the longitudinal dynamics of driver-assisted vehicles in mixed traffic,” in *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE, 2016, pp. 1634–1641.

[69] H. Jiang, J. Hu, S. An, M. Wang, and B. B. Park, “Eco approaching at an isolated signalized intersection under partially connected and automated vehicles environment,” *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 290–307, 2017.

[70] E. Ward, N. Evestedt, D. Axehill, and J. Folkesson, “Probabilistic model for interaction aware planning in merge scenarios,” *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 2, pp. 133–146, 2017.

[71] M. Treiber, A. Kesting, and D. Helbing, “Delays, inaccuracies and anticipation in microscopic traffic models,” *Physica A: Statistical Mechanics and its Applications*, vol. 360, no. 1, pp. 71–88, 2006.

[72] W. J. Schakel, B. Van Arem, and B. D. Netten, “Effects of cooperative adaptive cruise control on traffic flow stability,” in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*. IEEE, 2010, pp. 759–764.

[73] N. Chen, M. Wang, T. Alkim, and B. van Arem, “A robust longitudinal control strategy of platoons under model uncertainties and time delays,” *Journal of Advanced Transportation*, vol. 2018, 2018.

[74] S. Krauß, “Microscopic modeling of traffic flow: investigation of collision free vehicle dynamics.” 1998.

[75] M. Treiber and A. Kesting, “Traffic flow dynamics,” *Traffic Flow Dynamics: Data, Models and Simulation, Springer-Verlag Berlin Heidelberg*, 2013.

[76] P. Ranjitkar, T. Nakatsuji, and A. Kawamua, “Car-following models: an experiment based benchmarking,” *Journal of the Eastern Asia Society for Transportation Studies*, vol. 6, pp. 1582–1596, 2005.

[77] J. Rios-Torres and A. A. Malikopoulos, “A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps,” *IEEE Transactions on Intelligent Transportation Systems*, 2016.

[78] R. Hult, G. R. Campos, E. Steinmetz, L. Hammarstrand, P. Falcone, and H. Wymeersch, “Coordination of cooperative autonomous vehicles: Toward safer and more efficient road transportation,” *IEEE Signal Processing Magazine*, vol. 33, no. 6, pp. 74–84, 2016.

[79] G. R. de Campos, P. Falcone, and J. Sjberg, “Autonomous cooperative driving: A velocity-based negotiation approach for intersection crossing,” in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, Oct 2013, pp. 1456–1461.

[80] G. R. de Campos, P. Falcone, R. Hult, H. Wymeersch, and J. Sjberg, “Traffic coordination at road intersections: Autonomous decision-making algorithms using model-based heuristics,” *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 1, pp. 8–21, Spring 2017.

[81] L. Makarem and D. Gillet, “Fluent coordination of autonomous vehicles at intersections,” in *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2012, pp. 2557–2562.

[82] H. Kowshik, D. Caveney, and P. Kumar, "Provable systemwide safety in intelligent intersections," *IEEE transactions on vehicular technology*, vol. 60, no. 3, pp. 804–818, 2011.

[83] R. Hult, G. R. Campos, P. Falcone, and H. Wymeersch, "An approximate solution to the optimal coordination problem for autonomous vehicles at intersections," in *2015 American Control Conference (ACC)*. IEEE, 2015, pp. 763–768.

[84] J. Rios-Torres, A. Malikopoulos, and P. Pisu, "Online optimal control of connected vehicles for efficient traffic flow at merging roads," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, 2015, pp. 2432–2437.

[85] E. Steinmetz, R. Hult, G. R. de Campos, M. Wildemeersch, P. Falcone, and H. Wymeersch, "Communication analysis for centralized intersection crossing coordination," in *Wireless Communications Systems (ISWCS), 2014 11th International Symposium on*. IEEE, 2014, pp. 813–818.

[86] M. Nazari, T. Charalambous, J. Sjöberg, and H. Wymeersch, "Communication-aware control for intersection coordination of automated vehicles," in *ITSC 2017*.

[87] G. Schildbach, M. Soppert, and F. Borrelli, "A collision avoidance system at intersections using robust model predictive control," in *Intelligent Vehicles Symposium (IV), 2016 IEEE*. IEEE, 2016, pp. 233–238.

[88] M. Dring, K. Franke, R. Balaghiasefi, M. Gonter, M. Belkner, and K. Lemmer, "Adaptive cooperative maneuver planning algorithm for conflict resolution in diverse traffic situations," in *2014 International Conference on Connected Vehicles and Expo (ICCVE)*, Nov 2014, pp. 242–249.

[89] E. Debada, L. Makarem, and D. Gillet, "Autonomous coordination of heterogeneous vehicles at roundabouts," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2016, pp. 1489–1495.

[90] L. Wu, G. Guo, W. Yue, and Z. Gao, "Cooperative adaptive cruise control with communication constraints," in *Control Conference (CCC), 2015 34th Chinese*. IEEE, 2015, pp. 8015–8020.

[91] L. Zhang, H. Gao, and O. Kaynak, "Network-induced constraints in networked control systemsa survey," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 403–416, 2013.

[92] M. Wang, S. P. Hoogendoorn, W. Daamen, B. van Arem, B. Shyrokau, and R. Happee, "Delay-compensating strategy to enhance string stability of adaptive cruise controlled vehicles," *Transportmetrica B: Transport Dynamics*, pp. 1–19, 2016.

[93] M. Nazari, T. Charalambous, J. Sjöberg, and H. Wymeersch, "Remote control of automated vehicles over unreliable channels," in *IEEE WCNC 2018*.

[94] M. Mazzola, G. Schaaf, A. Stamm, and T. Krner, "Safety-critical driver assistance over lte: Toward centralized acc," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 9471–9478, Dec 2016.

[95] A. Geiger, M. Lauer, F. Moosmann, B. Ranft, H. Rapp, C. Stiller, and J. Ziegler, "Team annieway's entry to the 2011 grand cooperative driving challenge," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1008–1017, 2012.

[96] H. Zhou, R. Saigal, F. Dion, and L. Yang, "Vehicle platoon control in high-latency wireless communications environment: Model predictive control method," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2324, pp. 81–90, 2012.

[97] M. di Bernardo, P. Falcone, A. Salvi, and S. Santini, "Design, analysis, and experimental validation of a distributed protocol for platooning in the presence of time-varying heterogeneous delays," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 2, pp. 413–427, 2016.

[98] G. Naus, R. Vugts, J. Ploeg, R. van de Molengraft, and M. Steinbuch, "Cooperative adaptive cruise control, design and experiments," in *American Control Conference (ACC), 2010*. IEEE, 2010, pp. 6145–6150.

[99] J. Ploeg, B. T. Scheepers, E. Van Nunen, N. Van de Wouw, and H. Nijmeijer, "Design and experimental evaluation of cooperative adaptive cruise control," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. IEEE, 2011, pp. 260–265.

[100] J. Ploeg, N. van de Wouw, and H. Nijmeijer, "Fault tolerance of cooperative vehicle platoons subject to communication delay," *IFAC-PapersOnLine*, vol. 48, no. 12, pp. 352–357, 2015.

[101] L. Xu, L. Y. Wang, G. Yin, and H. Zhang, "Communication information structures and contents for enhanced safety of highway vehicle platoons," *IEEE Transactions on vehicular Technology*, vol. 63, no. 9, pp. 4206–4220, 2014.

[102] C. M. Massera, M. H. Terra, and D. F. Wolf, "Safely optimizing highway traffic with robust model predictive control-based cooperative adaptive cruise control," *arXiv preprint arXiv:1605.07493*, 2016.

[103] K. C. Dey, L. Yan, X. Wang, Y. Wang, H. Shen, M. Chowdhury, L. Yu, C. Qiu, and V. Soundararaj, "A review of communication, driver characteristics, and controls aspects of cooperative adaptive cruise control (cacc)," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 491–509, 2016.

[104] A. Saxena, H. Li, D. Goswami, and C. B. Math, "Design and analysis of control strategies for vehicle platooning," in *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE, 2016, pp. 1805–1812.

[105] K. Li, W. Ni, E. Tovar, and M. Guizani, "Lcd: Low latency command dissemination for a platoon of vehicles," *arXiv preprint arXiv:1801.06153*, 2018.

[106] J. Ploeg, E. Semsar-Kazerooni, G. Lijster, N. van de Wouw, and H. Nijmeijer, "Graceful degradation of cacc performance subject to unreliable wireless communication," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, Oct 2013, pp. 1210–1216.

[107] E. Consortium *et al.*, "Enhanced digital mapping project," *Final Report*, 2004.

[108] D. Green, J. Gaffney, P. Bennett, Y. Feng, M. Higgins, and J. Millner, "Vehicle positioning for c-its in australia (background document), austroads project no. nt1632," Tech. Rep., 2013.

[109] S. E. Shladover and S.-K. Tan, "Analysis of vehicle positioning accuracy requirements for communication-based cooperative collision warning," *Journal of Intelligent Transportation Systems*, vol. 10, no. 3, pp. 131–140, 2006.

[110] SAE, "Dsrc implementation guide, a guide to users of sae j2735 message sets over dsrc."

[111] U. DoD, "Global positioning system standard positioning service performance standard," *Assistant secretary of defense for command, control, communications, and intelligence*, 2008.

[112] ——, "Global positioning system standard positioning service performance standard," *Assistant secretary of defense for command, control, communications, and intelligence*, 2001.

[113] "Egnos safety of life (sol), service definition document," 2014. [Online]. Available: egnos-portal.gsa.europa.eu/sites/default/files/EGNOS_SOL_SDD_2.1.pdf

[114] J. Aponte, X. Meng, C. Hill, T. Moore, M. Burbidge, and A. Dodson, "Quality assessment of a network-based rtk gps service in the uk," *Journal of Applied Geodesy*, vol. 3, no. 1, pp. 25–34, 2009.

[115] F. Jimnez, J. E. Naranjo, F. Garca, and J. M. Armingol, "Can low-cost road vehicles positioning systems fulfil accuracy specifications of new adas applications?" *The Journal of Navigation*, vol. 64, pp. 251–264, 4 2011. [Online]. Available: http://journals.cambridge.org/article_S0373463310000470

[116] N. Kuter and S. Kuter, "Accuracy comparison between gps and dgps: A field study at metu campus," *Italian Journal of Remote Sensing*, vol. 42, no. 3, pp. 3–14, 2010.

[117] T. Schönberg, M. Ojala, J. Suomela, A. Torpo, and A. Halme, "Positioning an autonomous off-road vehicle by using fused dgps and inertial navigation," *International Journal of Systems Science*, vol. 27, no. 8, pp. 745–752, 1996.

[118] "Joint galileo optimization and vanet enhancement," [Online; accessed 30-June-2016]. [Online]. Available: http://www.gsa.europa.eu/joint-galileo-optimization-and-vanet-enhancement

[119] "High precision positioning for cooperative-its," [Online; accessed 4-April-2017]. [Online]. Available: http://hights.eu/

[120] H. Li and F. Nashashibi, "Multi-vehicle cooperative localization using indirect vehicle-to-vehicle relative pose estimation," in *Vehicular Electronics and Safety (ICVES), 2012 IEEE International Conference on*, July 2012, pp. 267–272.

[121] E. Arpin, C. Shankwitz, and M. Donath, "A high accuracy vehicle positioning system implemented in a lane assistance system when gps is unavailable," 2011.

[122] G. M. Hoang, B. Denis, J. Härri, D. Slock *et al.*, "Distributed link selection and data fusion for cooperative positioning in gps-aided ieee 802.11 p vanets," *Proc. WPNC*, vol. 15, 2015.

[123] J. Levinson, M. Montemerlo, and S. Thrun, "Map-based precision vehicle localization in urban environments." in *Robotics: Science and Systems*, vol. 4. Citeseer, 2007, p. 1.

[124] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on.* IEEE, 2010, pp. 4372–4378.

[125] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt *et al.*, "Towards fully autonomous driving: Systems and algorithms," in *Intelligent Vehicles Symposium (IV), 2011 IEEE.* IEEE, 2011, pp. 163–168.

[126] R. W. Wolcott and R. M. Eustice, "Visual localization within lidar maps for automated urban driving," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on.* IEEE, 2014, pp. 176–183.

[127] G. Hoang, B. Denis, J. Härri, and D. Slock, "Cooperative localization in gnss-aided vanets with accurate ir-uwb range measurements," in *Positioning, Navigation and Communications (WPNC), 2016 13th Workshop on.* IEEE, 2016, pp. 1–6.

[128] J. Wenger and S. Hahn, "Long range and ultra-wideband short range automotive radar," in *2007 IEEE International Conference on Ultra-Wideband*, Sept 2007, pp. 518–522.

[129] L. H. Eriksson and B. O. As, "A high performance automotive radar for automatic aicc," in *Proceedings International Radar Conference*, May 1995, pp. 380–385.

[130] R. Rajamani, *Vehicle dynamics and control.* Springer Science & Business Media, 2011.

[131] M. Segata, S. Joerer, B. Bloessl, C. Sommer, F. Dressler, and R. L. Cigno, "Plexe: A platooning extension for veins," in *2014 IEEE Vehicular Networking Conference (VNC)*, Dec 2014, pp. 53–60.

[132] F. Liu and Y. Chen, "Improved model predictive control for cooperative adaptive cruise control subject to actuator delay," in *2017 Chinese Automation Congress (CAC)*, Oct 2017, pp. 4747–4722.

[133] E. Van Nunen, J. Verhaegh, E. Silvas, E. Semsar-Kazerooni, and N. Van De Wouw, "Robust model predictive cooperative adaptive cruise control subject to V2V impairments," in *Intelligent Transportation Systems (ITSC), 2017 IEEE 20th International Conference on.* IEEE, 2017, pp. 1–8.

[134] D. Ruina, C. He, Z. Qiang, L. Keqiang, and L. Yusheng, "Acc of electric vehicles with coordination control of fuel economy and tracking safety," in *2012 IEEE Intelligent Vehicles Symposium*, June 2012, pp. 240–245.

[135] S. Li, K. Li, R. Rajamani, and J. Wang, "Model predictive multi-objective vehicular adaptive cruise control," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 3, pp. 556–566, 2011.

[136] I. G. Jin and G. Orosz, "Dynamics of connected vehicle systems with delayed acceleration feedback," *Transportation Research Part C: Emerging Technologies*, vol. 46, pp. 46–64, 2014.

[137] S. Öncü, J. Ploeg, N. Van de Wouw, and H. Nijmeijer, "Cooperative adaptive cruise control: Network-aware analysis of string stability," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 4, pp. 1527–1537, 2014.

[138] F. Gao, S. E. Li, Y. Zheng, and D. Kum, "Robust control of heterogeneous vehicular platoon with uncertain dynamics and communication delay," *IET Intelligent Transport Systems*, vol. 10, no. 7, pp. 503–513, 2016.

[139] W. Xu, J. Pan, J. Wei, and J. M. Dolan, "Motion planning under uncertainty for on-road autonomous driving," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on.* IEEE, 2014, pp. 2507–2512.

[140] I. G. Jin and G. Orosz, "Data-driven parameter estimation for optimal connected cruise control," in *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on.* IEEE, 2017, pp. 3739–3744.

[141] S. B. McLaughlin, J. M. Hankey, T. A. Dingus, and S. G. Klauer, "Development of an fcw algorithm evaluation methodology with evaluation of three alert algorithms," *National Highway Traffic Safety Administration, Tech. Rep*, 2009.

[142] S. Brunson, E. Kyle, N. Phamdo, and G. Preziotti, "Alert algorithm development program: Nhtsa rear-end collision alert algorithm," Tech. Rep., 2002.

[143] M. Barbier, "Experienced network access selection with regards to safety applications," Eurecom, 2013.

[144] L. Svensson and L. Masson, "Safe Stop Trajectory Planning for Highly Automated Vehicles: An Optimal Control Problem Formulation," in *Intelligent Vehicles Symposium (IV), 2018 IEEE*, June 2018, pp. 517–522.

[145] Y. Zheng, S. E. Li, K. Li, F. Borrelli, and J. K. Hedrick, "Distributed model predictive control for heterogeneous vehicle platoons under unidirectional topologies," *IEEE Transactions on Control Systems Technology*, 2016.

[146] M. Wang, W. Daamen, S. P. Hoogendoorn, and B. van Arem, "Driver assistance systems modeling by model predictive control," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*, Sept 2012, pp. 1543–1548.

[147] C. Liu, S. Lee, S. Varnhagen, and H. E. Tseng, "Path planning for autonomous vehicles using model predictive control," in *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 2017, pp. 174–179.

[148] ——, "Path planning for autonomous vehicles using model predictive control," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, June 2017, pp. 174–179.

[149] I. CVX Research, "CVX: Matlab software for disciplined convex programming, version 2.0," http://cvxr.com/cvx, Aug. 2012.

[150] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds. Springer-Verlag Limited, 2008, pp. 95–110, http://stanford.edu/~boyd/graph_dcp.html.

[151] J. Mattingley and S. Boyd, "Cvxgen: A code generator for embedded convex optimization," *Optimization and Engineering*, vol. 13, no. 1, pp. 1–27, 2012.

[152] P. Seiler and R. Sengupta, "Analysis of communication losses in vehicle control problems," in *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*, vol. 2, 2001, pp. 1491–1496 vol.2.

[153] W. B. Qin, M. M. Gomez, and G. Orosz, "Stability and frequency response under stochastic communication delays with applications to connected cruise control design," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 2, pp. 388–403, 2017.

[154] M. Sepulcre and J. Gozalvez, "On the importance of application requirements in cooperative vehicular communications," in *Wireless On-Demand Network Systems and Services (WONS), 2011 Eighth International Conference on*. IEEE, 2011, pp. 124–131.

[155] M. Gudmundson, "Correlation model for shadow fading in mobile radio systems," *Electronics letters*, vol. 27, no. 23, pp. 2145–2146, 1991.

[156] M. E. Renda, G. Resta, P. Santi, F. Martelli, and A. Franchini, "Ieee 802.11 p vanets: Experimental evaluation of packet inter-reception time," *Computer Communications*, vol. 75, pp. 26–38, 2016.

[157] M. Zorzi and R. R. Rao, "On the statistics of block errors in bursty channels," *IEEE transactions on communications*, vol. 45, no. 6, pp. 660–667, 1997.

[158] N. An, T. Gaugel, and H. Hartenstein, "Vanet: Is 95% probability of packet reception safe?" in *2011 11th International Conference on ITS Telecommunications*, Aug 2011, pp. 113–119.

[159] ISO2631-1, "Mechanical vibration and shock - evaluation of human exposure to whole-body vibration - part 1: General requirements," *International Organization for Standardization*, 1997.