Adaptive Cooperative Highway Platooning and Merging

by

Feyyaz Emre Sancar

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2017

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner	Professor Ya-Jun Pan
Supervisor(s)	Associate Professor Baris Fidan Professor Jan P. Huissoon
Internal Member	Associate Professor Soo Jeon Professor Amir Khajepour
Internal-external Member	Associate Professor Nasser Lashgarian Azad

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

As low-cost reliable sensors are introduced to market, research efforts in autonomous driving are increasing. Traffic congestion is a major problem for nearly all metropolis'. Assistive driving technologies like cruise control and adaptive cruise control are widely available today. While these control systems ease the task of driving, the driver still needs to be fully alert at all times. While these existing structures are helpful in alleviating the stress of driving to a certain extent, they are not enough to improve traffic flow. Two main causes of congestion are slow response of drivers to their surroundings, and situations like highway ramp merges or lane closures. This thesis will address both of these issues.

A modified version of the widely available adaptive cruise control systems, known as cooperative adaptive cruise control, can work at all speeds with additional wireless communication that improves stability of the controller. These structures can tolerate much smaller desired spacing and can safely work in stop and go traffic. This thesis proposes a new control structure that combines conventional cooperative adaptive cruise control with rear end collision check. This approach is capable of avoiding rear end collisions with the following car, as long as it can still maintain the safe distance with the preceding vehicle. This control structure is mainly intended for use with partially automated highways, where there is a risk of being rear-ended while following a car with adaptive cruise control. Simulation results also shows that use of bidirectional cooperative adaptive cruise control also helps to strengthen the string stability of the platoon. Two different control structures are used to accomplish this task: MPC and PD based switching controller.

Model predictive control (MPC) structure works well for the purpose of bidirectional platoon control. This control structure can adapt to the changes in the plant with the use of a parameter estimator. Constraints are set to make sure that the controller outputs are always within the boundaries of the plant. Also these constraints assures that a certain gap will always be kept with the preceding vehicle. PD based switching controller offers an alternative to the MPC structure. Main advantage of this control structure is that it is designed to be robust to certain level of sensor noise. Both these control structures gave good simulation results.

The thesis makes use of the control structures developed in the earlier chapters to continue developing structures to alleviate traffic congestions. Two merging schemes are proposed to find a solution to un-signaled merging and lane closures. First problem deals with situations where necessary levels of communication is not present to inform surrounding drivers of merging intention. Second structure proposes a merging protocol for cases where two platoons are approaching a lane closure. This structure makes use of the modified cooperative adaptive cruise control structures proposed earlier in the thesis.

Acknowledgments

First and foremost, I would like to express my thanks to my supervisors Prof. Baris Fidan and Prof. Jan P. Huissoon for their endless support of my Ph.D. study. I am in debt for all their contribution of time, ideas, and funding to make my Ph.D. experience productive and stimulating.

I would also like to thank Prof. Ya-Jun Pan, Prof. Soo Jeon, Prof. Amir Khajepour and Prof. Nasser Lashgarian Azad, for taking time to being in my committee and guiding me with their helpful comments.

In addition, I would like to thank Prof. Steven Waslander for his support and guidance during our collaborations on building scale-model vehicles.

Here, I would also like to thank Prof. Paolo Falcone, Robert Hult and all my team mates from Chalmers University of Technology for their hospitality and support during the Grand Cooperative Driving Competition.

I would like to thank to NSERC and Nuvation Inc., the industrial partner in the NSERC CRD project "Autonomous Driving Strategies for Urban and Highway Environments", which financially supported this PhD thesis study.

Finally, I would like to extend my sincere thanks to Mehdi Jalalmaab for being a great teammate in countless works we have collaborated.

Dedication

To my mother and the loving memory of my father

Table of Contents

List of Figures

List of Abbreviations

AAC	Advisory Acceleration Controller
ACC	Adaptive Cruise Control
CACC	Cooperative Adaptive Cruise Control
CACC+	Modified Cooperative Adaptive Cruise Control
CC	Cruise Control
DSRC	Dedicated Short Range Communication
GCDC	Grand Cooperative Driving Competition
LCAC	Lane Change Assistance Function
LQR	Linear Quadratic Regulator
MIO	Most Important Object
MPC	Model Predictive Control
PD	Proportional-Derivative Controller
RCC	Road Closure Counter
rFMIV	Right-Forward Most Important Vehicle
rRMIV	Right-Rear Most Important Vehicle
RSU	Road Side Unit
SPM	Static Parametric Model

List of Symbols

a	Acceleration
$f(bs)$	Blind-Spot Avoidance Function
c	Relative Lateral Spacing
d	Relative Longitudinal Spacing
D	Desired Relative Longitudinal Spacing
D_0	Constant Spacing
δ	Spacing Error
γ_c	Ego-Vehicle Collision Check
$G(s)$	Vehicle Transfer Function
η_m	Measurement Noise
h	Time Headway
$H(s)$	Spacing Policy Transfer Function
J	MPC Cost Function
$\bar{k}$	Proportional Control Constant
K	Plant Gain
t	Time [sec]
τ	Closed-Loop Bandwidth of the Plant
t_{col}	Time-to-Collision Index
u	Acceleration Input

v	Velocity
$\bar{v}$	Velocity Error
Υ	Constraint Hardness Vector
W	Controller weight
x	Position Along x-Axis
X	Vehicle State Vector
y	Position Along y-Axis

Chapter 1

Introduction

1.1 Scope, Motivation and Objective

With the continually increasing number of vehicles on the roads, traffic congestion is becoming a major socio-economic issue. Unproductive time is spent by drivers waiting in traffic jams, with an accompanying increase in stress levels. A recent study reported that the total length of all daily traffic jams in the Netherlands can exceed the length of the country itself [1].

There are essentially two types of traffic jams: those caused by the inadequacy or incapacity of the road itself to physically accommodate all the cars; and ghost traffic jams. A ghost traffic jam can occur when human drivers follow each other closely in highway rush-hour traffic, and a small braking action by one of the cars is magnified as following drivers react in succession. This can even cause the traffic to come to a complete stop [1]. This phenomenon is explained by "string instability" in control theory. When a system is "string stable", small disturbances are attenuated as they propagate upstream, which in turn provides smooth traffic flow with less congestion [2]. In designing controllers for platoons, string stability is one of the key points that should be taken into consideration. Also it is essential that these controllers be reliable.

Adaptive cruise control (ACC) is available in many vehicles today. This technique was initially intended to improve safety and comfort, and was not developed to increase traffic efficiency [3]. Interest in ACC research accelerated in the 1980s with the California PATH research program, the aim of which was to use advanced technology to increase highway safety and throughput [4]. To increase roadway capacity and traffic efficiency, the inter-vehicle distances should be small, while ensuring the string-stability of the platoon. This

task is difficult to achieve with typical ACC because the platoon becomes string unstable as the inter-vehicle spacing is reduced. Cooperative adaptive cruise control (CACC), where the vehicles communicate wirelessly, can help alleviate this issue.

The main difference between ACC and CACC is the introduction of wireless communication of vehicle location, speed, and acceleration between vehicles in an ad-hoc network. CACC has been shown to allow smaller inter-vehicular spacing while maintaining string stability [3]. However, a generic CACC controller does not yet exist, and most controllers available are optimized for specific scenarios.

Chapters 3 and 4 of this thesis focus on modified CACC controllers that have additional rear-end collision avoidance capabilities. We refer to the vehicle of interest in a task definition as the ego-vehicle, the vehicle immediately ahead of the ego-vehicle as the preceding vehicle, and the vehicle immediately behind the ego-vehicle as the following vehicle. Current CACC control structures only consider the spacing between the ego-vehicle and the vehicle (or vehicles) ahead of it.To avoid possible rear end collision cases, a particular strategy is to reduce the spacing to the preceding vehicle, while ensuring that this does not lead to a front-end collision.

Using a string stable control structure, along with a good sensor fusion algorithm, eliminates the aforementioned ghost traffic jams. In order to eliminate the other type of traffic jams, due to poor infrastructure or lane changes, additional cooperation is required between vehicles. These traffic jams usually occur at specific locations in the traffic such as lane closures or highway entrance ramps. Recent research has shown that such congestions play an important role in blockage of upstream traffic [5]. Chapters 5 and 6 of the thesis focus on one cause of this type of traffic jams, highway merging. Most of the existing cooperative highway merging research has focused on on-ramp merging. While there are several different merging protocols available for on-ramp merging, the topic of lane closures has been left mostly untouched. A generic vehicle-lane merging control structure where vehicles do not have sufficient level of wireless communication is proposed in this thesis.

Chapter 6 of this thesis focuses on designing a merging scheme for cooperative autonomous vehicles. This merging scheme utilizes the switching controller introduced in the previous chapters. One of the most prominent merging protocols for highway merging was introduced by the organizers of the GCDC (Grand Cooperative Driving Challenge 2016) [6]. Results of the merging scheme we have proposed are simulated in PreScan.

1.2 Contributions of the Thesis

The work presented in this thesis contributes to the area of platoon formation and highway merging within the area of connected driving. Chapter 2 gives an overview of the current state-of-the-field and identifies where this thesis fits within the more general field of autonomous driving.

Chapter 3: This chapter presents our novel MPC-based approach to bidirectional CACC. The control structure proposed in this section is designed to work both in fully-autonomous and semi-autonomous environments. Another novelty of this control structure is combination of the MPC structure with least square parameter estimator to make the controller robust to changes in the plant parameters. Two separate MPC-based control structures are presented in this section. Parts of the work presented in this section is published in [7].

Chapter 4: Another solution to the problem of bi-directional CACC, introduced in 3, is presented in this chapter. Even though MPC-based structures give good results, due to computational intensity, they are usually not the easiest to implement. We treated the question of bi-directional CACC as a formation control problem to come up with a structure tailor-made for real life use. An important feature that sets this solution apart from others reported in the literature is the use of a switching-control technique to eliminate instabilities caused by sensor noise. This chapter also investigates the effects of asymmetric bidirectional CACC structures, where front and rear vehicles have different weightings. It is concluded that asymmetric case where the weight of the front vehicle is three times the weight of the rear vehicle gives the best results and assures strong string stability. Parts of the work presented in this section is published in [8].

Chapter 5: This chapter investigates a merging problem where vehicles do not have proper channels of communications to indicate their merging intentions. The work presented in this chapter is designed considering that not all vehicles will have equal levels of autonomy at the earlier stages autonomous highways. At the earlier stages of autonomous highways, every self-driving vehicle should be equipped with structures that can detect the intentions of it's surrounding to avoid any foreseeable collision. The structure proposed in this chapter works to decipher the intent of a regular vehicle by observing its states.

Chapter 6: The structure used in the Chapter 5 is extended for fully autonomous highways where two lanes need to merge due to an upcoming lane closure. This chapter proposes the unique use of graph-based formation control to formalize the merging protocol. The merging protocol introduced in this chapter is designed in a way to minimize the effects of the platoon merging on the traffic. Only the first N vehicles of the merging lane actively

takes part in the merging to make sure that the traffic never comes to halt. Also this chapter recommends the use of blind spot avoidance to make sure any two vehicles are never right next to each other.

To summarize, three different highway traffic platooning problems are investigated in this thesis, namely, bi-directional longitudinal spacing control, platoon cut-in by a rogue vehicle, and highway merging protocol for lane-closures. The topics investigated in each chapter are inter-dependent, and each subsequent chapter utilizes the concepts proposed in the previous one.

1.3 Organization of the Thesis

The outline of the thesis is as follows:

Background information of the concepts used in this thesis along with an overview of what other researchers in this field are working on, is provided in **Chapter 2**. An initial concept of string stability is briefly introduced. This is followed by an overview of some important platooning research going on in the world. Differences between conventional *cruise control*, *adaptive cruise control* and *cooperative adaptive cruise control* will be discussed and this will be followed by the different types of control structures used for these. Different types of sensors to be used in this research will also be presented. This section will conclude with a discussion of the effects of communication and actuator delays.

In **Chapter 3**, an improved cooperative cruise control structure is described, which checks for rear end collisions and finds the optimal inter-vehicle spacing. *Model Predictive Control* is used for this purpose. MPC allows the vehicle to check the rear end spacing and act accordingly, while still making sure that no collision will happen with the preceding vehicle. Another advantage of MPC is that all states can be constrained, which is important for the controller to act within the limitations of the vehicle. MPC is implemented using MATLAB's MPC toolbox, which has the option of softening constraints based on their priorities. While the MATLAB's MPC toolbox has advantages like softening of constraints, it makes the controller unpredictable; also the toolbox does not allow the controller model to change once the simulation starts, which makes it impossible to use parameter estimators. The chapter follows with introduction of a second MPC based control structure. Instead of working with MATLAB's MPC toolbox, a C-based implementable MPC code is used for the same purpose. The Plant model is altered to work with this new structure, and most importantly, all the constraints on the system are modified. This new controller does not allow constraint softening, and the use of constraints was limited to only absolutely

necessary ones. A parameter estimator is used to make the system more robust to changes in the environment.

An alternative type of controller with a similar purpose is described in **Chapter 4**. Here switching control is used for the purpose of CACC with rear end check. This control structure is designed in a way to increase the string stability of the platoon while giving good results under sensor noise. This chapter explores the use of asymmetric bidirectional control, where front and rear vehicle have different importances to the control action.

Chapter 5 delves into an issue that might arise in the early stages of automated highways where there are still semi-autonomous vehicles on the road. During these stages a vehicle should not be just relying on communication between the vehicles, but observe their surroundings to understand their actions. This chapter proposes a vehicle-lane merging detection where a rogue vehicle tries to merge in to the neighboring lane without any signal.

The problem defined in Chapter 5 is extended in **Chapter 6** for a fully automated highway where two platoons need to merge due to a lane closure. This chapter proposes a complete merging scheme with longitudinal control structures which were introduced in previous chapters. Merging scheme is designed in a way to cause minimal disruptions to the traffic. This chapter also proposes a novel blind spot avoidance structure which makes sure that any two vehicles are never next to each other.

Finally, **Chapter 7** presents the summary of the works presented in this thesis with possible future directions.

Chapter 2

Background and Literature Review

This chapter introduces the concepts used in this dissertation along with an overview of other research work being done in this field. Initially, necessary vehicle dynamics and spacing equations will be explained in the preliminaries and notations section. Then, the concept of string stability, one of the key elements in platooning and autonomous driving tasks, is introduced without any formal analysis. Later, evaluation of cooperative adaptive cruise control (CACC) is discussed. In the field of CACC and adaptive cruise control (ACC) no consensus has been reached on a type of control structure that performs best, so various types of controllers are still used. These controller designs are introduced, as are their effects on string stability. Even if most of these designs give good results in simulation, actuator and communication delays degrade their performance. Then, the effects of different spacing policies on string stability are presented. Finally highway merging structures will be introduced. These structures will be examined in three categories: *Lane Change Structures*, *Merging Strategies for Mixed Traffic* and *Fully Autonomous Merging Strategies*. The chapter ends with an in-depth explanation of the merging structure from the Grand Cooperative Driving Challenge (GCDC) 2016, which is the baseline structure used to compare to the work of this thesis.

2.1 Preliminaries and Notations

This section introduces the vehicle model used throughout the thesis, along with the equations used for spacing error calculations and a generic CACC controller for platooning.

The most commonly used vehicle model for platooning purposes is defined in the form [3]:

$$G_i(s) = \frac{K_i}{(\tau_i s + 1)}, \tag{2.1}$$

where τ_i represents the time constant. Here, subscript i denotes the vehicle number, such that the platoon lead vehicle has subscript 1 and the last vehicle in the platoon has subscript N. The input to the transfer function of Eq. 2.1 is the desired acceleration u_i and the output is the actual vehicle acceleration a_i. In certain sections this transfer function is followed by a double integral so that the model output is the vehicle position x_i. The subscript i represents the ego-vehicle, while its leader is represented with $i-1$. The exact model used in each chapter is presented before the controller design.

The distance between the ego-vehicle i and the preceding vehicle is given by:

$$d_i(t) = x_{i-1}(t) - x_i(t). \tag{2.2}$$

The control inputs for most structures are the deviations from the desired velocity v_i and spacing D_i, and are defined as:

$$\begin{aligned} \delta_i(t) &= d_i(t) - D_i(t), \\ \bar{v}_i(t) &= v_{i-1}(t) - v_i(t). \end{aligned} \tag{2.3}$$

The value of $D_i(t)$ is not constant, but is a linear function of the ego-vehicle's velocity. A constant time headway spacing policy is used for $D_i(t)$, and this is defined as:

$$D_i(t) = D_0 + h_i v_i(t), \tag{2.4}$$

where h_i is the desired headway time, and D_0 is the desired spacing at standstill. Throughout this thesis, D_0 is ignored since it has no effect on the working of the controller.

A generic CACC structure is presented in figure 2.1.

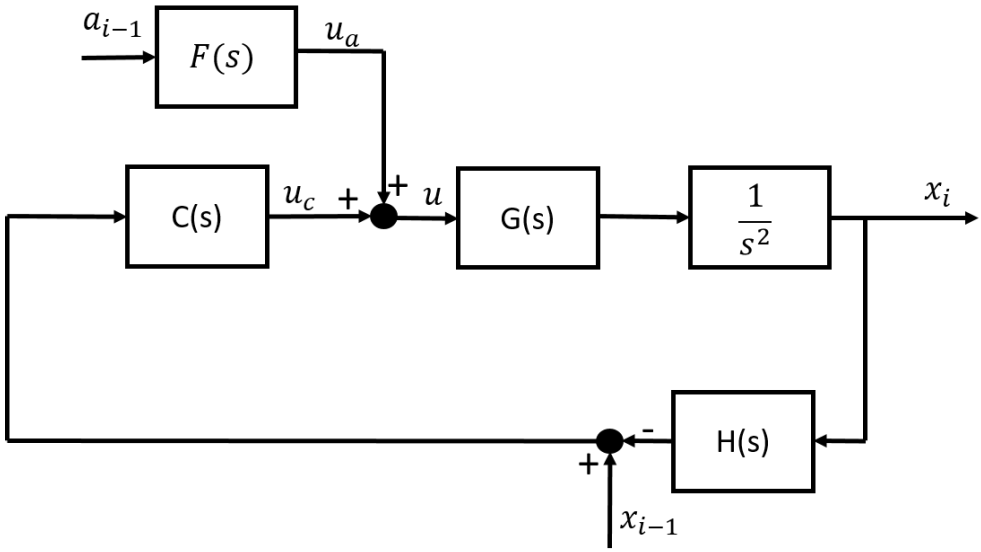


Figure 2.1: Generic CACC control structure.

In this control structure, the spacing policy transfer function is $H(s)$. For a constant time headway spacing policy and with $D_0 = 0$, this transfer function is

$$H_i(s) = 1 + h_i s. \tag{2.5}$$

The input to the plant $G(s)$ is comprised of two signals. The input can be shown as:

$$u = u_c + u_a, \tag{2.6}$$

where u_c is the output of the controller and u_a is the feed-forward component. The actual form of $F(s)$, if it is used, depends on the form of the controller. It is used to compensate for the vehicle dynamics.

Throughout this thesis, unless otherwise specified, all variables are in the time domain.

2.2 String Stability

String stability is one of the key areas of research for autonomous driving systems. In control theory, a system is described as unstable when a bounded input will lead to an

unbounded output. For autonomous driving applications, stability can be categorized as follows [9]:

1. Local,
2. Platoon (String Stability),
3. Traffic Flow.

These concepts are defined in [9]. When a pair of vehicles are following each other, if the spacing error or disturbance decreases with time, then the system is called *Locally Stable*.

When three or more vehicles are involved, string stability is checked on top of local stability. A platoon is called *String Stable* if a perturbation in spacing between two vehicles is attenuated as it propagates upstream (i.e. between the following cars). Some sources also refer to this as being *Asymptotically Stable*.

Platoon stability has been studied both theoretically and empirically. Empirical analysis is conducted by setting up a traffic simulation and introducing disturbances to the system. If the platoon can damp these disturbances, then it is called string stable; if not, the platoon is called unstable [9]. The analytical method for studying string stability is based on the transfer function between the velocity of the preceding vehicle (v_{i-1}) and the velocity of the ego-vehicle (v_i). If the magnitude of this transfer function is less than unity, then the system is string stable [10].

String stability equation for frequency domain is discussed in detail in [3]. The string stability transfer function for decentralized vehicles is defined as following:

$$\Phi_{\Lambda,i}(s) = \frac{\Lambda_i(s)}{\Lambda_{i-1}(s)}, \tag{2.7}$$

where $\Lambda \in u, x, \delta$. The sufficient condition for string stability is given by:

$$\|\Phi_{\Lambda,i}\|_\infty \leq 1, \tag{2.8}$$

for $i > 1$.

Similarly, string stability equations for time-domain are discussed in [11]. Here string stability is defined as:

$$\max_{t\geq 0} |a_{i+1}(t)| \leq \max_{t\geq 0} |a_i(t)| \tag{2.9}$$

where $i = 2, ..., N-1$, N being the number of vehicles in the platoon. This equation states that response of each following vehicle to their leader should be less than or equal to the response of the leader.

For traffic flow stability, Pueboobpaphan mentions that there still is not a single definition and discusses several of the alternative descriptions in the literature. Traffic flow is considered unstable if instability within one platoon is transferred to another. Traffic flow stability is dependent on platoon size, distribution and stability. Also if the traffic flow density is above a critical value, then it is considered unstable [9].

Some of the factors that affect string stability is discussed in [9]. Increasing the following factors tends to stabilize the platoon:

1. Number of automated vehicles in the platoon,
2. Desired spacing (Time headway),
3. Anticipation (Ability to react to changes before the preceding vehicle does),

while the following factors usually have a destabilizing effect on the platoon:

1. Limit of jerk,
2. Time lag of the plant,
3. Total reaction time.

2.3 Cooperative Adaptive Cruise Control

Current research on CACC can be roughly categorized as follows [12]:

1. Longitudinal CACC
   (a) Distance Keeping
   (b) Stop & Go

(c) Platooning

2. Lateral CACC

   (a) Collision Avoidance

   (b) Lane Keeping

   (c) Merging

This thesis mainly focuses on the platooning aspect of CACC. This section will start by briefly describing some of the ongoing platooning projects, then discuss the commonly used sensors for these applications. Later, individual control structures and their performances will be discussed.

2.3.1 Platooning Projects

Some of the major platooning projects currently taking place will be briefly introduced in this section. Each of these projects have different goals and different working requirements. These differences are [13]:

1. Longitudinal and/or lateral control,
2. Heavy duty and/or passenger vehicles,
3. Changes to the existing infrastructure.

PATH is one of the earliest programs to address automated vehicle platooning. The objective was to increase highway capacity with minimum changes to existing infrastructure. All vehicles on the highway are assumed to be automated, including the platoon leader, and separate platoons are formed for different types of vehicles. Both longitudinal and lateral control were addressed, the latter being based on reference markers on the roadway. A recent focus is more on heavy-duty vehicles because of higher energy saving opportunities [13].

GCDC 2011 (Grand Cooperative Driving Challenge) is a competition organized to advance cooperative driving systems. In the 2011 GCDC event, only longitudinal control was done autonomously, with lateral control by the driver. The test ground was shared

between both heavy-duty and passenger vehicles. Both urban and highway driving scenarios were tested during the competition. Other than certain communication constraints and requirements, each car was equipped with different sets of sensors [10].

SARTRE is a project funded by the European Commission, that aims to increase fuel efficiency, reduce congestion and increase safety without changing infrastructure. SARTRE uses mixed heavy-duty and passenger vehicles for platooning. The lead vehicle is driven by a trained driver, while all the followers are automated. Only V2V communication is used, other than the on-board sensors. Both lateral and longitudinal control can be done autonomously [14].

Demo 2000 Cooperative Driving took place in Tsukuba, Japan, where five autonomous vehicles rode on an oval-shaped test track to demonstrate various scenarios. The main focus of this demonstration was stop-and-go, platooning, splitting of the platoon, and merging two platoons into a single platoon [15].

GCDC 2016 is the continuation of the Grand Cooperative Driving challenge that took place in 2011. This competition extends the focus beyond pure platooning. The main scenarios of this competition include platoon merging and cooperative intersection navigation. The platoon merging scenario will be discussed in detail on the upcoming sections [6].

2.3.2 Overview of Sensors

Sensors have one of the most important roles in real life implementation of platooning and CACC. There is still considerable diversity in the selection of sensors for autonomous vehicles. Table. 2.1 shows sensors used in some of the state-of-the-art prototype vehicles by some of the largest car manufacturers.

For the purpose of platooning, CCD cameras or 3D-lidar are not essential. Some researchers have proposed a vision based approach for lateral and longitudinal control [16]. With the availability of high speed embedded computer hardware and low cost CCD cameras, many car manufacturers use these, but they are still not as reliable as other sensors and are a lot more susceptible to changes in the environment and weather.

For the works that are presented in this thesis, sensors required are:

1. GPS/IMU,
2. Lidar (these will be positioned at the front at rear bumpers of the vehicle to measure inter-vehicle spacing),

3. Wireless communication unit.

BMW	Mercedes-Benz	Nissan	Google	General Motors
Video camera	3D stereo camera	Front and side radars	Rooftop LIDAR	Several lateral sensors
Radar sensors	Additional camera for road sign and traffic light detection	Camera	Camera	Differential GPS
Side laser scanners	Short and long range radar	Front, rear and side laser scanners	Front and side radar	Cameras
Ultrasonic sensors	Infrared camera	Four wide angle cameras	IMU	Very accurate map
Differential GPS	Ultrasonic sensors		Wheel encoders	
Very accurate map			Very accurate map	

Table 2.1: Sensors on some of the current prototype cars [17].

2.3.3 CC/ACC/CACC/CACC+

The following clarifies the terminology that will be used when discussing the various forms of cruise control. The most basic, Cruise Control (CC) simply maintains the speed of the vehicle at a desired value, without using any environmental sensors. This functionality is only suitable for use on highways and requires the driver to be fully alert and in control of the vehicle.

Adaptive Cruise Control (ACC) is an extension of Cruise Control. ACC functionality is activated when there is preceding traffic and the controller keeps a desired distance from the preceding vehicle. [18] One of the limitations of typical ACC systems is that they cannot be used below 30-40 km/h and cannot be used in stop-&-go traffic [19]. Also ACC fails to meet string-stability criteria when small time-headways or constant spacing policies are used.

Cooperative Adaptive Cruise Control (CACC) is an extension to ACC with additional wireless communication with the preceding vehicle. This allows the controller to be string-stable at lower speeds and smaller time-headways. The term CACC+ is generally used for controllers that use multiple node ad-hoc wireless communication structures and associated control strategies.

2.3.4 Different Controller Architectures

This section will review some of the different controllers being used for platooning/CACC. In order for these controllers to perform well, three performance objectives should be satisfied. These are [20]:

1. Individual vehicle stability,
2. String-stability,
3. Low steady-state spacing errors.

Fuzzy Logic Control

Fuzzy logic controllers have been used for the purpose of platooning. Some of the advantages of this controller type are its simplicity, flexibility and ease of dealing with nonlinear systems [21]. Naranjo [19] presents a fuzzy based control structure for CC and ACC. This is part of AUTOPIA project from Spain, which aims to develop testbeds for experimenting for various autonomous vehicle applications. The control inputs to the proposed controller are *speed error*, *time gap error*, *change in time gap* and *acceleration*. This controller is tested on Citroen Berlingo vans and the results were promising, even at lower speeds.

Another group working on a similar controller [21], proposed an ACC system with collision avoidance and warning system. The controller gets *time to collision with front vehicle*, *time to collision with rear vehicle* and *acceleration* to use in the fuzzy controller. The novel thing about this work is that it also tries to avoid a collision with the following car, which most ACC systems do not implement. Their controller is not tested on an actual vehicle but simulations show good results.

Olivares-mendez [22] used a fuzzy control structure, but for line following rather than platooning. They devised a *Fuzzy+I* controller that does both lateral and longitudinal control. Limitations in their vision algorithms prevent them from testing their system at higher speeds.

P/PD/PID Controller Structures

Cooperative adaptive cruise controller introduced by Naus et al. [3] is one of the most comprehensive works in terms of mathematical string-stability analysis. The main aspect

that distinguishes this paper is that both communication and actuator delays are considered in the string stability analysis. All the string stability analysis is done in the frequency domain, and a PD controller is used for both the ACC and CACC cases, for which the parameter selection is discussed in depth. The gain for the feed-forward component is chosen to eliminate steady state error. The string stability section is comprised of two parts: the first considers heterogeneous vehicles to find the correct criteria for string-stability, while the second compares *constant spacing* and *variable spacing* policies to conclude that under actuator and communication delays, a variable spacing policy is the better option.

Another interesting work in this field has been done by Senturk et al [23]. Their work does not include a formal string-stability analysis, but they have a novel controller structure that gives good results in simulation. This controller assumes that each vehicle in the platoon transmits its velocity and position instead of acceleration information. The desired acceleration comes from the controller, and is formalized as follows:

$$\begin{gathered} a_{ref} = k_p e_{x,i} + k_v e_{v,i}, \\ e_{x,i} = (1-\alpha_i)(d_{i,i-1} - d_{ref,i}) + \alpha_i(d_{i,i-1} - \textstyle\sum_{n=2}^{i} d_{ref,n}), \\ e_{v,i} = [(1-\alpha_i)(v_{i-1} - v_i) + \alpha_i(v_{ref} - v_i)]sign(e_{x,i}), \\ \alpha_i = \frac{d_{i,i-1}}{d_{ref,i}} \; bounded \; between \; 0-1. \end{gathered} \tag{2.10}$$

In this way the controller acts like an ACC for smaller following distances and like a CACC for larger spacing [24] Ploeg et al presented the design and experimental validation of their controller, which is a slightly improved version of Naus et al's [3] controller. The main difference in the controller is that instead of using a simple PD controller, they choose to also include jerk which made the controller PDD. Both their experimental and theoretical results show that CACC allows lower time headways without compromising string stability. They conclude that with CACC, string stability can be achieved for time headways as short as 0.7s.

Another interesting study on controller design was done by van den Broek et al [1]. They devised a new controller which they called *Advisory Acceleration Controller*. This controller collects the acceleration information coming from all the preceding vehicles in the platoon and uses these in a weighted manner. Selection of these weights or the ideal number of vehicles to include in the controller is not discussed in the paper, but their experimental results show that AAC is a lot less susceptible to delays in communication.

Kianfar et al. [4] worked both on PD and MPC controllers for their GCDC entry. Their paper shows both simulations and experimental results of these controllers. Both controllers give string stable results with small position errors. MPC however is still

superior to PD, since it satisfies the constraints of the system, but the downside of using this controller is that the design procedure is much more complicated compared to PD controllers, and for this reason they chose to use PD control in the GCDC competition.

Another interesting modification to Naus et al.'s [3] work was done by Halmstadt University for the GCDC competition [25]. The novel thing about this controller is instead of considering acceleration information coming from all the vehicles, it just gets it from the preceding and the lead vehicle. The controller compares these acceleration values and chooses the minimum. Another thing different in this controller is that the gain of the feed-forward information is chosen based on making the platoon string stable, not to eliminate steady state error. Results this team got at the competition was satisfactory.

Team Mekar also published a paper with a similar PD controller with a CACC/ACC switch [26]. Their paper does not have any formal string-stability analysis, but they have useful discussion on the implementation of the controller and the problems that they have faced. Lee et al. [27] did testing on a Kia Mohavi using a PID controller. The control structure used only ACC, and no V2V communication was used. The interesting thing is how they decided on the stop distance ($dist_{stop} = dist_{brake} + dist_{thinking}$), where $dist_{brake}$ is computed based on the parameters of the vehicle. The paper has no analysis on how to select safety distance as this effects the string stability, or if it is any better than the existing constant-time-headway method.

Another interesting work from GCDC was done by Team FUTURUM [28]. During the competition they were only able to implement a similar controller to Naus et al. [3], but they proposed another multi-information control structure which they called CACC+. The output to the controller is formulated as follows:

$$a_{di} = (k_{di}e_{vi} + k_{pi}e_{xi}) + \left(\frac{k_{di}}{i-1}\sum_{j=1}^{i-1} e_{v,j}\right). \tag{2.11}$$

Model Predictive Control

Bageshwar et al. investigated MPC for ACC/CC [29], and divided longitudinal following into transitional maneuvers and steady-state operation. Transitional maneuvers occur when there is no leader to follow, so the car switches from ACC to CC. Simulation results validate the method, but also demonstrate that collisions may occur between the lead and follower vehicle.

Corona et al. employed MPC for their non-linear ACC control of a SMART car [30]. The car model included the transmission system, and one of the objectives of the MPC

formulation was to minimize changing gears. A comparison was made between MPC with different degrees of approximations and a PI controller. Simulation results demonstrate that the PI control gives the least desirable results of all configurations, even though computation time is much lower.

Naus et al. also employed explicit MPC for ACC Stop-Go [31]. The problem is posed and solved as a multi-parametric quadratic program, which is used for off-line optimization to avoid expensive computation on-board the vehicle - this restricts the approach to precomputed solutions. Seven cases were used to define an envelope of working conditions and the controller was tested for these scenarios.

Bu et al. demonstrated an on-line Model Predictive controller on an Infinity FX45 vehicle [32]. For the sake of simplicity, the on-line MPC is combined with the existing on-board ACC controller. The output coming from the MPC controller is converted into a virtual range rate command, which is fed into the ACC controller. The two stage approach was demonstrated to be quite effective at managing inter-vehicle spacing.

Kianfar et al. implemented an MPC controller on a Volvo S60 [11], which included additional time domain constraints to assure that the platoon remains string stable.

Summary of Controller Structures

1. Implementing and performing formal string stability-analysis on LQR is nearly as complex as MPC, but LQR does not have the ability to put constraints on the states and the outputs of the system, which is the main advantage of MPC over other controllers.

2. PD controllers are the most commonly used for CACC, because of their ease of implementation and that they also give reasonable experimental and theoretical results. They usually fail to perform well in less powerful vehicles, where the PD controller cannot accommodate the limitations of the actuators.

3. Fuzzy logic controllers have no proven advantage over MPC or PD, also it's a lot more complicated to do formal stability analysis for these controllers.

4. Several different CACC controllers have been proposed that utilize information coming from multiple vehicles. It is has been shown that this gives better performance than only using information from the preceding vehicle, but there is still not a consensus on how many vehicles should be included or how this information should be weighted.

2.4 Spacing Policies

A significant number of studies have been done on different spacing policies and their effects on the string stability of the platoon. Swaroop showed in 1994 that in order to achieve string-stability with constant spacing in a platoon additional inter-vehicle communication is required, but when constant headway control is used string-stability can be achieved without the need of additional inter-vehicle communication [33].

A similar study was done by Klinge [34]. They came to the conclusion that if little or no time headway is used, string-stability deteriorates as the string length increases.

One of the best recent studies in this field was that by Naus et al. [3]. They performed a detailed analysis of constant and time-headway spacing for ACC and CACC under various conditions. For the velocity-independent case where ideal vehicle dynamics is assumed ($G_i(s) = \frac{1}{s^2}$), they concluded that only marginal string-stability can be achieved for ACC (where no feed-forward gain is used) and for CACC (when feed-forward gain is used). If there is a communication delay in the CACC case, marginal string stability can be preserved only for a strict case. [3].

For the case of velocity-dependent constant time-headway case, string-stability can be achieved with ACC if the desired head-way time is larger than a certain minimum head-way time, which is dependent on the controller parameters. If CACC is used, string-stability can be achieved even with smaller minimum head-ways. For example with the same controller parameters, if a 200 ms communication delay is present, the minimum head-way for CACC is 0.8 s, while for ACC it is 2.8 s [3].

2.4.1 Bidirectional Control Structures

Choosing the correct spacing policy for the backward component plays an important role in the string stability of the control structure. One of the earliest studies in this field [35] explores different ACC based bidirectional platoon control strategies to see the effects of the spacing policy. Two bidirectional control structures are presented in this work. The first (BiACC V1) uses the velocity of the preceding vehicle to set the rear desired spacing. The spacing policy for this case is given by:

$$\begin{aligned} |D_{i,front}| &= h_i v_i, \\ |D_{i,rear}| &= h_i v_{i+1}. \end{aligned} \tag{2.12}$$

The second structure (BiACC V2) uses the velocity of the ego-vehicle to set both front and rear desired spacings.

$$\begin{aligned} |D_{i,front}| &= h_i v_i, \\ |D_{i,rear}| &= h_i v_i. \end{aligned} \tag{2.13}$$

[35] compares how the string stability of these two control structures are effected by the change of rear vehicles weight. The work concludes that increasing the weight of the rear vehicle has a stabilizing effect for the BiACC V2, which means that string stability can be achieved for lower time headways. When the same analysis is done for the BiACC V1, it is seen that increasing the weight of the rear vehicle decreases string stability. This work concludes that for bidirectional structures using desired spacing based on the speed of the ego vehicle for both rear and front vehicles have a positive effect on the string stability.

2.5 Highway Lane Change & Platoon Merging

In recent years, research interest in highway merging has increased exponentially, as reliable cost-effective sensors have become available, and substructures, like CACC, are more widely used. According to recent studies, dangerous lane changes cause both unstable traffic flow and 10% of all crash-related traffic delays. Moreover, lane change and merging related accidents make up to 5% of all police-reported collisions [36].

Research on highway lane changes and platoon merging can be generally divided into four categories:

1. Lane Change Structures,
2. Merging Models,
3. Merge Assist,
4. Fully Autonomous Merging Strategies.

Research on *Lane Change Structures* focuses on lateral control techniques to be used during lane changes. Research in the area of *Merging Models* work on identifying and modeling merging behaviors of actual vehicles, an important component of traffic simulation software. Merging models often use parameters such as politeness factor or incentive criteria, to accurately model different types of driving [37]. Unlike merging models, *Merging*

Strategies and *Merge Assist* do not focus on modeling of merging behaviors, and instead focus on creating strategy-level control structures for connected vehicles.

There are several aspects that are of particular interest in our research, the key ones related to merging controllers (listed similarly in [5]) are:

1. Merging Scenario,
2. Control Type,
3. Control Over,
4. Controlled Subject.

Different *Merging Scenarios* have been investigated over the last twenty or so years, the most common being *on-ramp merging* and *general lane change*.

Control Type refers to whether the computation is done in a *centralized* or *decentralized* manner. In a *centralized* control structure, there is a central unit like an RSU or platoon leader that makes decisions for the other vehicles. In a *decentralized* control structure, each vehicle is responsible for its own decision making.

Some research on highway merging focuses on either the vehicles in the merging lane, e.g. the on ramp, or the gap opening lane, e.g. the main-carriageway, and so it is crucial to define what the proposed control structure is *Control Over*. All merging structures listed in the fully autonomous merging section, have control over both lanes.

Earlier research in this field focused mainly on the control of a single vehicle, whereas the current trend has shifted to platoon merging in recent years.

This thesis focuses on designing a strategy and interaction protocol for autonomous vehicles along with a longitudinal control structure that can both assure string-stability of the platoon and facilitate the demands of the merging action. Since the main focus of this research is *Merging Strategies*, research on *Merging Models* is not presented, since this is outside the scope.

2.5.1 Lane Change Maneuvers

On an implementation level, lane change maneuvers constitute a key part in executing highway merging. However, when it comes to traffic stability and efficacy of the connected

merging protocol for a fully automated highway, lane change maneuvers play a less significant role. This thesis focuses on the longitudinal control and stability, along with the required interaction protocol. This section will give a quick overview of the trends in this field and a concise review of the keys works. A recent survey of the field [38] groups lane change maneuvers into two subcategories:

1. Tracking Control Approach,
2. Lateral Guidance Algorithm.

The lateral guidance algorithms employ a desired yaw generator and yaw rate track controller to get the desired yaw rate to complete the maneuver [38]. Yang [39] uses the homing guidance approach, which is used to guide vehicles on the merging ramp to a target gap they had located. This system is designed in a way to minimize the need of infrastructure support. Similar to this work, Kachroo [40] introduced three control structures for the guidance of the lane changing vehicle into the pre-defined gap.

The tracking control approach creates a virtual desired path to be followed by the vehicle [38]. Cao published [41] is one of the recent works in this field. This work used MPC for cooperative merging path generation. Also, recently the lane change maneuver concept of a single vehicle is extended to platoons in [42], in which all vehicles in the platoon need to change their lane. In this work, the vehicle virtually follows its leader while changing lane simultaneously. This paper assumes that the neighboring lane is empty, so no high-level merging strategies are presented. Goli [43] investigated the effects of different lateral control strategies when a vehicle is merging into a platoon.

2.5.2 Merging Assist

Even though no consumer-vehicles came equipped with CACC in the early 1990s, the concept of CACC already existed. As works on fully-automated on-ramp merging structures gained more interest, researchers also started exploring how this CACC structure can be used to help with the on-ramp merging issue. Since then, this research has focused on either the effects of various cruise control structures on traffic throughput during merging, or creating a merging assist structure that can be used in mixed traffic. This section will give a chronological overview of works in this field. Techniques that are applicable to the focus of this thesis will be presented in more detail

A highway entrance simulator was designed by Halle [44] and used to investigate the effects of vehicle type on traffic throughput. This work concluded that the type of vehicle

on both the merging lane and main-carriageway affects the performance of the merging action, and as a result suggests grouping similar vehicle types. CACC has long been proven to increase traffic stability, as discussed in previous sections. VanArem [45] investigated the effects of mixed-CACC use in the case of lane merging caused by a reduction in the number of lanes. Their results show that penetration rates higher than 60% increase the stability whereas rates lower than 40% have minimal effect. Davis [46] similarly tested the effects of ACC penetration on their own on-ramp merging structure, and concluded that traffic performance increases with higher penetration rates.

The merging structure proposed by Wang [47] demonstrated the advantages of using a proactive merging strategy, where the decision point for merging vehicles is before the actual merging point. The concept of proactive merging is furthered by Kanavalli [48], with the use of sliding windows to determine the logical order of vehicles, and which shows a positive effect on traffic throughput. Both these works assume control over the merging vehicles, while having little control over the main-carriageway. Another proactive merging strategy is presented by Choi [49], using state transition diagram based algorithms where limited control over both on-ramp and main-carriageway vehicles is assumed. A decentralized merging assistant is presented by Pueboobpaphan [50], which assumes limited control over the main-carriageway. Automated vehicles on the main-carriageway calculate a safety zone, the safe space needed on the main-carriageway by the merging vehicle, of the incoming vehicles and adjust their speed accordingly.

Ramp metering techniques (the use of a signaling device aimed to regulate the on-ramp traffic on busy highways), have been in use for many years, although this is not an optimal technique. Traditional ramp metering only controls the on-ramp vehicles, while it does not play any role in gap opening. Scarinci [51] introduced the concept of cooperative ramp metering where connected vehicles on the main-carriageway play an active role in opening up the required gaps. Another merge assist structure for on-ramp vehicles was presented in [52], in which the cost of merging was computed to inform the driver when and how to merge into the main-carriageway.

A comprehensive state-machine based merging strategy is presented in [53] for autonomous vehicles, but the main contribution of this work is the analysis of interferences caused by non-automated vehicles and faulty communication. In a recent work by Dang [54], a new ACC structure is proposed (called the Lane Change Assistance function or LCAC), which aims to assist the driver during the lane change maneuver. This MPC-based structure can follow multiple targets to make the lane change smoother, along with lane-change warning for surrounding vehicles.

2.5.3 Merging Strategies

In recent years, interest has increased in developing fully-autonomous, implementable merging strategies. This field has been an objective of research since the early 1990s, when the sensor systems and computational capacity required for these systems became more affordable and accessible. Earlier research mostly focused on highway on-ramp merging; but there has been an increasing interest in merging strategies for general lane change scenarios. Even though the on-ramp merging structures provide a good basis for the study of general lane change problems, the solutions do not always apply to all lane-change scenarios. In an on-ramp merging problem, all the vehicles on the main carriageway that are approaching the on-ramp already recognize that the vehicles on the on-ramp will be merging with the main carriageway. The issue in a more general merging problem, where a random vehicle on the road decides to merge into another lane, is that surrounding vehicles do not recognize the merging intention unless it is either communicated, via a turn signal or wireless communication, or the vehicle in the destination lane constantly scans its environment to assess the intentions of its immediate neighbors. For this reason, advances in these merging scenarios will be investigated separately.

On-Ramp Merging

A distributed control algorithm for both the merging and gap-opening lanes was presented in [55] as a part of the California PATH program. Discrete logical behaviors defined the actions that should be taken by each vehicle. To open up a gap for the merging vehicle, the vehicle on the main-carriageway shifts its state from *Cruise* to *Yield*, which means increasing the time headway. Similarly, Ran [56][57] introduced a logic-based structure, which accounted for platoon formations on the main-carriageway. The platoon leader has considerably more autonomy and control over the merging process in this structure.

A real-time implementation of on-ramp merging is presented by Lu [58], where virtual platooning is utilized for a safe and effective implementation of the algorithm on different roads. This work also includes lateral control design and reference trajectory generation. Another use of virtual vehicles in on-ramp merging can be found in [59]. As a part of the AUTOPIA program, Milanes [60] introduces and tests an RSU based automated ramp entrance system. The proposed architecture controls the longitudinal movement of both the on-ramp and main-carriageway to minimize negative effects of merging. Another RSU-based merging strategy has been proposed by Marinescu [61], where vehicles approaching the on-ramp are assigned a slot. Fig. 2.2 shows the chessboard slot-formation, which

increases possible merging maneuvers using free slots in other lanes of the highway. Vissim simulation showed that this approach increased the traffic throughput.

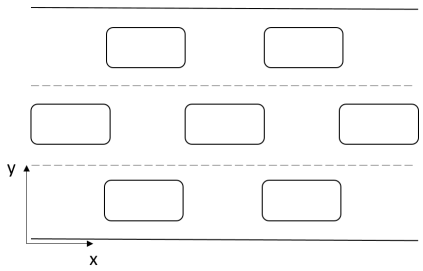


Figure 2.2: The chessboard slot-formation [61].

A proactive merging strategy, presented by Awal [62], used an optimization approach to merge the on-ramp and main-carriageway vehicles. One major drawback of this approach is the use of the platoon leader as the decision maker for the rest of the platoon. A different approach to the same problem comes from [63], in which a cost and prediction function based algorithm is presented. Vehicles get to predict how other agents will be affected by its actions, which is why the authors choose to call it *socially cooperative driving*.

In a comprehensive study investigating automatic merging of vehicles, Raravi [64] presents three different control algorithms, where each have various DSRC communication topologies, and these algorithms are compared for varied traffic densities. This work also incorporates a preliminary solution for dealing with non-autonomous vehicles.

A seminal study on the effects of joining and exiting maneuvers on platoon structures is presented by Fernandes [65]. The algorithm introduced in this work aims to avoid merging vehicles joining into small gaps within the platoon, to improve safety and string-stability.

General Lane Changes

A structure with a two-layer communication protocol has been proposed by Sakaguchi [66] and Uno [67]. In order to decrease the communication traffic, the main communication is platoon-to-platoon while platoons are able to communicate within each other. The high-level merging structure is only explained briefly, and no logic diagrams nor communicated messages are specified. These works also introduced the concept of virtual vehicle following, where the merging vehicle is treated as if it is in the main-carriageway. The concept of two-layer communication is furthered by the work of Halle [44] by investigating

four different communication and coordination methodologies, namely *Hard-Centralized*, *Centralized*, *Decentralized* and *Teamwork*.

In an investigation into fairness in merging order, Baselt [68] argues that the waiting time for each vehicle should be fair. Concepts of *free-flow fairness* and *arrival time* are introduced, which is the time a vehicle would have arrived at the merging point if there was no traffic. In order for the merging order to be fair, cars with earlier free-flow arrival time should go first.

Platoon Merging

Starting in the early 2000s, interest in platoon merging increased significantly. The results of Demo 2000 *Demo 2000*, a cooperative driving demonstration, are presented by Kato [15]. Their novel steady state positioning, shown in Fig. 2.3, makes the merging process smoother. This is one of the earliest demonstrations of two platoons merging without the use of RSUs. Around the same time, Horowitz [69] introduced a new *lane-change within platoons* maneuver, as part of the PATH project. This study also addresses string stability and intra-platoon spacing policies.

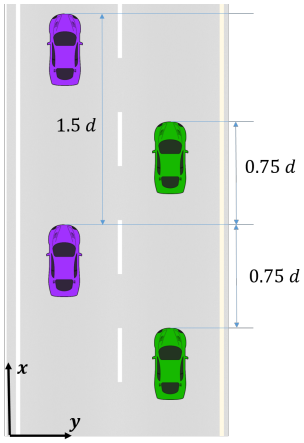


Figure 2.3: Driving formation under steady state [15].

A logic-based merging strategy for platoon merging has been proposed by Lam [70], where platoons are to perform overtaking maneuvers as a single unit. As a special case, hazard detection is discussed where a member of the platoon may temporarily seek refuge in the neighboring platoon.

A comprehensive merging strategy for platoon-merging along with a performance comparison of three different longitudinal controllers is presented in [71]. This work proposes a decentralized communication scheme, although the platoon leader plays the most active role in decision making. The proposed merging structure gives a brief overview of the protocol, although it does not provide details of the messages communicated and which vehicle makes the decisions. Also the use of a data cloud rather than a more common Wi-Fi based communication scheme makes the implementation more difficult to assess.

Grand Cooperative Driving Challenge 2016

One of the most comprehensive interaction protocols for highway platoon merging has been created for the 2016 GCDC. The competition comprised of three scenarios: platoon merging, collaborative intersection negotiation, and emergency vehicle accommodation. For the purpose of this thesis, we will be only focusing on the platoon merging scenario of this competition [6].

Organizers list some of the challenges associated with platoon merging [72] as:

1. If all vehicles were to start opening up a gap simultaneously, there would be a huge deceleration towards the end of the platoon,
2. On the other hand, opening up gaps one at a time would be very inefficient and slow,
3. Based on the communication protocol designed for the competition, safe-to-merge messages are not associated with an ID, so they suggest this might cause confusion.

The merging scenario starts with two platoons, on two neighboring lanes, cruising at different speeds. Upon notification of a lane closure for one of the lanes, both platoons align their speeds with a pace-making protocol. During this phase, platoon leaders will adjust their position for a smooth merge. The next step is simultaneous pairing, where all vehicles in the 'gap opening' lane choose their paired vehicle in the 'merging' lane, and for whom they will be opening up a gap. As the name implies, this pairing is done simultaneously and the pairings do not change throughout the merging action. During this phase all gap opening vehicles will be opening up their gaps with respect to their pairs. During the competition, pair acknowledgment flags were not used even though the original protocol suggested using them. This stage is followed by a sequential pairing phase, where the leader of the merging platoon chooses their forward and backwards pair. The backwards pair refers to the vehicle in the gap opening lane, which has previously paired

with the vehicle in question. A forward pair is the vehicle preceding the backwards pair. Only the platoon leader from the gap opening platoon chooses its pairs and starts opening up respective gaps. After receiving a 'safe-to-merge' flag, it passes the leader flag to the following vehicle. This means that a vehicle on the merging platoon can start making a gap only when the vehicles preceding it have started merging or are ready to start merging [72].

A paper submitted by the organizers prior to the competition [73] suggest the use of estimated arrival time based pairings, but these were not used in the competition. The organizers also suggested the use of an aggregate control strategy, where separate agents will be added to the control input on a need-to basis [72]. Some of the agents suggested by the organizers are listed as follows:

1. **CACC agent** which is the basic cruise control agent,
2. **OA agent** is the agent responsible for opening up the gap with the pair. This should not be confused with the obstacle avoidance agent to be proposed in the upcoming merging chapter,
3. **CA agent** is used to activate the emergency braking in case two vehicles get very close.

All in all, GCDC 2016 proposed a good decentralized merging protocol, but there is still much room left for improvement. Some of the disadvantages of this proposed interaction protocol are as follows:

1. Vehicles cannot void their pairings, which can lead to slowing down or stopping in the case of rogue pairings,
2. If the vehicle leader, for some unforeseen reason, fails to open up the required gap, all other vehicles following it will not be able to start their merge,
3. A simultaneous merging stage alone can cause the last vehicle to slow down too much,
4. The protocol requires the platoon leaders to coordinate the initial position to start the merge,
5. Extensive use of V2V messages.

Chapter 3

MPC-Based Collaborative Adaptive Cruise Control with Rear End Collision Avoidance

3.1 Problem Definition

In the 2011 GCDC challenge most CACC implementations used a linear feedback controller with a feed-forward component, as shown in Fig. 3.1.

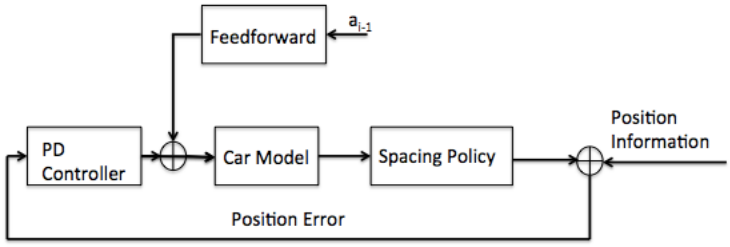


Figure 3.1: Feedback control model.

PD controller gains are usually chosen experimentally and the feed-forward controller is designed to ensure the string stability of the system. This model works reasonably well if the vehicle is on a straight level path without any changes to the surroundings.

This approach assumes a simplified car model, where the parameters have negligible effects on the feedback controller. However, in practice the car model changes based on factors such as the number of passengers in the vehicle, tire traction, road conditions and slope. The feedback controller is tuned for the ideal environment, which will often lead to poor performance under real life conditions. It also does not account for the limitations of the vehicle, such as maximum velocity, acceleration, and braking.

One objective of this work is therefore to use a controller that can be fully adaptive to changes in the car model. This is a prerequisite if the vehicles are to be used in real life scenarios, as no methods exist to finely tune controllers to work in every specific condition a vehicle might encounter.

Another concern with the existing class of PD control methods lies in their inability to incorporate limitations coming from the lateral controller and the slip circle. These fundamental vehicle limits must be addressed for controller deployment, as they are inherent in every vehicle on the road.

Combined with these additional objectives, the main goal of this controller design remains to keep a safe distance with the preceding vehicle. Finally, the formulation of the CACC control using MPC easily admits a rear-end collision check, which is added to increase vehicle safety in mixed equipage environments.

3.2 Dynamic Model

The MPC controller structure used in this work is developed utilizing a similar MPC structure to [4], while expanding the structure from unidirectional CACC to bidirectional CACC.

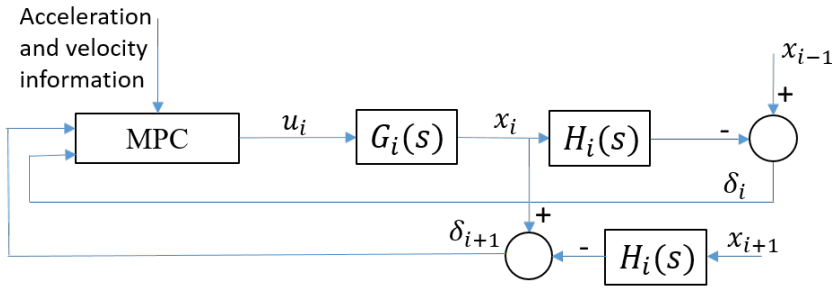


Figure 3.2: Proposed control model.

The car model is defined as:

$$G_i(s) = \frac{K_i}{s^2(\tau_i s + 1)}, \tag{3.1}$$

where τ_i represents the closed-loop bandwidth. We assume a platoon of N vehicles, vehicle 1 being the lead vehicle. Each vehicle is assumed to have 2 lidar sensors, one at the front facing ahead, and one at the rear facing back. The relative distances to the vehicles immediately ahead of and behind vehicle i, as measured by these lidar sensors, are:

$$\begin{aligned} d_i(t) &= x_{i-1}(t) - x_i(t), \\ d_{i+1}(t) &= x_i(t) - x_{i+1}(t). \end{aligned} \tag{3.2}$$

Our control input signal is the error between the actual relative distance (d_i) and the desired relative distance (D_i) between cars, and the velocity errors. These are defined as:

$$\begin{aligned} \delta_i(t) &= d_i(t) - D_i(t), \\ \bar{v}_i(t) &= v_{i-1}(t) - v_i(t). \end{aligned} \tag{3.3}$$

The actual value of $D_i(t)$ is not constant, but is a linear function of the spacing policy. A constant time headway spacing policy is used in our model. The formula for $D_i(t)$ is defined as:

$$D_i(t) = D_0 + h_i v_i(t), \tag{3.4}$$

where h_i is referred to as the desired headway time, and D_0 is the desired spacing at standstill (which is added to the actual length of the car to give a virtual vehicle length used with zero spacing policy at zero velocity). The spacing policy transfer function is then:

$$H_i(s) = 1 + h_i s. \tag{3.5}$$

The input to this transfer block is the position of the car. Each car is also equipped with a wireless data receiver, which receives the acceleration information of both the preceding and following vehicles.

3.3 MPC Control Model with Constraint Softening

Instead of using a single MPC, we adopted an MPC with switching in order for the error to not accumulate when the rear car is far away. The switch is based on $\delta_i(t)$. If $\delta_{i+1}(t)$ is smaller than 10m, then an MPC model with 6 states and rear-end collision check is used. Otherwise, a 4 state MPC, developed by [4], with only a preceding vehicle collision-check is used. This ensures that our controller performs equivalently to other MPC controllers in most situations, and even better in critical cases. The states used for the first MPC are $X = [\delta_i, \bar{v}_i, a_i, v_i, \delta_{i+1}, \bar{v}_{i+1}]^T$. The state update equations are given by:

$$\begin{aligned} \dot{\bar{v}}_i(t) &= a_{i-1}(t) - a_i(t), \\ \dot{\bar{v}}_{i+1}(t) &= a_{i-1}(t) - a_i(t), \\ \dot{\delta}_i(t) &= \bar{v}_i(t) - a_i h_i, \\ \dot{\delta}_{i+1}(t) &= \bar{v}_{i+1}(t) - a_{i+1} h_i. \end{aligned} \tag{3.6}$$

The state dynamics equation is given by:

$$\begin{aligned} \dot{X}(t) &= AX(t) + Bu(t) + \tilde{B}\tilde{u}(t), \\ Y(t) &= CX(t), \end{aligned} \tag{3.7}$$

where

$$\tilde{u}(t) = [a_{i-1}, a_{i+1}]^T,$$

$$A = \begin{bmatrix} 0 & 1 & -h_i & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1/\tau_i & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ K_i/\tau_i \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tilde{B} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -h_i \\ 0 & -1 \end{bmatrix}, \tag{3.8}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The state dynamics equation is discretized before using it in the MPC. This ensures that the platoon leader is followed without any delays.

The cost function for the MPC controller is defined as follows:

$$\begin{aligned} J(\Delta u) = &\sum_{i=0}^{p-1} [y(k+i+1) - r(k+i+1)]^T Q [y(k+i+1) - r(k+i+1)] \\ &+ \delta u(k+i)^T R_{\delta u} \delta u(k+i) + [u(k+i)^T R_u u(k+i)], \end{aligned} \tag{3.9}$$

where k represents the current time instance and p is the controller prediction horizon [74]. This optimization problem is bounded by constraints [74]:

$$\begin{aligned} u_{i,min} - \epsilon \Upsilon^U_{i,min} \leq u_i(t) \leq u_{i,max} - \epsilon \Upsilon^U_{i,max}, \\ \delta u_{i,min} - \epsilon \Upsilon^{\delta U}_{i,min} \leq \delta u_i(t) \leq \delta u_{i,max} - \epsilon \Upsilon^{\delta U}_{i,max}, \\ y_{i,min} - \epsilon \Upsilon^y_{i,min} \leq y_i(t) \leq y_{i,max} - \epsilon \Upsilon^y_{i,max}. \end{aligned} \tag{3.10}$$

Here ϵ represents the panic variable, which is set to 1 if the controller cannot get a solution within the bounds. The vector Υ defines which constraints are hard and which are soft, and their level of softness.

The second MPC controller has nearly the same properties as the first, the only differences being the states and state update equations. The states used for the second MPC are $X = [\delta_i, \bar{v}_i, a_i, v_i]^T$ The state update equations are given by:

$$\begin{aligned} \dot{\bar{v}}_i(t) &= a_{i-1}(t) - a_i(t), \\ \dot{\delta}_i(t) &= \bar{v}_i(t) - a_i h_i. \end{aligned} \tag{3.11}$$

The state dynamics equation is:

$$\begin{aligned} \dot{X}(t) &= AX(t) + Bu(t) + \tilde{B}\tilde{u}(t), \\ Y(t) &= CX(t), \end{aligned} \tag{3.12}$$

where

$$\tilde{u}(t) = [a_{i-1}]^T,$$

$$A = \begin{bmatrix} 0 & 1 & -h_i & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & -1/\tau_i & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ K_i/\tau_i \\ 0 \end{bmatrix}, \tilde{B} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \tag{3.13}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

3.3.1 Constraints on the System

A critical aspect of using an MPC is choosing the right constraints and right level of softness for each.

δ_i and δ_{i+1} are the distance errors between vehicles. Ideally, these should be zero with both the preceding and the following vehicle, but the minimum constraint for both of these are set as $-h_i\bar{v}_i/4$. This allows the vehicle to get closer to the preceding vehicle to avoid a collision with the follower, while still maintaining a safe distance of $3h_i\bar{v}_i/4$. A maximum value is defined for the preceding vehicle while there is no upper limit for the follower vehicle.

$\bar{v}_i$ and $\bar{v}_{i+1}$ are the velocity errors and should be kept within certain limits to ensure performance. There is a lower and upper limit for the preceding vehicle, while for the follower vehicle there is only a lower limit.

u is the input to the actuator and its limits are defined by the limits of the vehicle itself. v_i is the velocity of our vehicle. It also has its own limitations, both coming from the actuator and the lateral controller, to maintain vehicle stability.

To ensure that the system is string stable we have to make certain that the acceleration of the vehicle is always less than that of the preceding vehicle if the acceleration is positive. If the acceleration is negative then it becomes the lower limit. This constraint is selected in a order to assure string stability by following the definition of string stability by [11].

All constraints regarding the 2nd follower, position and velocity errors, are defined as semi-soft constraints. The priority of the controller is still to keep a safe distance with

the preceding vehicle, and the rear end collision check is just an additional functionality. Actuator and velocity errors are defined as hard constraints, while the acceleration is semi-hard. When the following vehicle gets closer to the ego-vehicle and the preceding vehicle is moving at constant speed, preset constraint limits the ego-vehicle to have zero acceleration. In this case the acceleration constraint will be broken.

The velocity error with the lead vehicle is also defined as a soft constraint. The minimum position error with the preceding vehicle is defined as a hard constraint, while the maximum is semi-hard.

3.3.2 Simulation

Two cases will be shown to demonstrate the operation of the proposed controller. All the simulations are done in MATLAB/Simulink using the Model Predictive Control Toolbox.

Velocity profiles of the first and third vehicles are set manually; only the second (ego) vehicle is equipped with the MPC controller. In each of the plots, $V1$ represents the lead vehicle, while $V3$ represents the following vehicle.

The car model used is based on a Volvo S60 [11]. For this vehicle, K is set to 1 and τ_i to 0.4. Time headway h is defined as 1 sec. The weights of the cost function are as follows: $R_{\Delta u} = 0, R_u = 0$ and the diagonals of matrix Q are 10, 8, 3, 0, 6 ,8.

When the 3rd vehicle is within 10 meters of the 2nd vehicle, the weights given previously are used with the 6-state vector. Otherwise the state vector of the MPC is modified to be: $X = [\delta_i, \bar{v}_i, a_i, v_i]^T$ and the new diagonal for Q will be 10, 8, 3, and 0. When the follower vehicle goes out of bounds, then the regular MPC takes on the controller.

Constraints on the system are as follows:

$$\begin{aligned} -4.5 &\leq u \leq 2.5, \\ -h_i\bar{v}_i/4 &\leq \delta_i \leq 2, \\ -2.5 &\leq \bar{v}_i \leq 2.5, \\ -4.5 &\leq a \leq 2.5, \\ 0 &\leq v_i \leq 40, \\ -h_i\bar{v}_i/4 &\leq \delta_{i+1}, \\ -2.5 &\leq \bar{v}_i. \end{aligned} \tag{3.14}$$

The level of hardness for each of these constraints can be seen from $\Upsilon^y_{i,min}$ and $\Upsilon^y_{i,max}$, which were defined in Eq. 3.10. Zero represents a hard constraint, while 1 is a soft

constraint. The smaller the number gets, the harder the constraint gets. Constraints coming from the limitations of the vehicle itself had to be set as hard. The spacing error with the preceding vehicle (δ_i) also has a hard constraint. For the 6-state MPC the hardness values are:

$$\Upsilon^y_{i,min} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \Upsilon^y_{i,Max} = \begin{bmatrix} 0.1 \\ 1 \\ 0.5 \\ 0 \\ 1 \\ 1 \end{bmatrix}. \tag{3.15}$$

For the 4-state MPC, the hardness values are:

$$\Upsilon^y_{i,min} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \Upsilon^y_{i,Max} = \begin{bmatrix} 0.1 \\ 1 \\ 0.5 \\ 1 \end{bmatrix}. \tag{3.16}$$

In the first case, the first and third vehicle start accelerating with the same acceleration and keep constant speed after the 30th second. However, the third vehicle has a small sinusoidal disturbance in its velocity profile. Fig. 3.3 shows the velocity profiles of all three vehicles, and how the second vehicle adjusts its velocity to keep a safe distance to both vehicles.

As specified in the constraints, the position error with the first vehicle never goes below $-h_i\bar{v}_i/4$, but since the constraints for the second vehicle were soft it can go below this if necessary, even though it's not desired. Position error (δ_{i+1}) going below zero does not mean a collision has occurred, but rather indicates that the desired spacing policy has been violated. This is mainly because the priority of this controller is still the first vehicle. Fig. 3.4 shows the position error graphs for both vehicles.

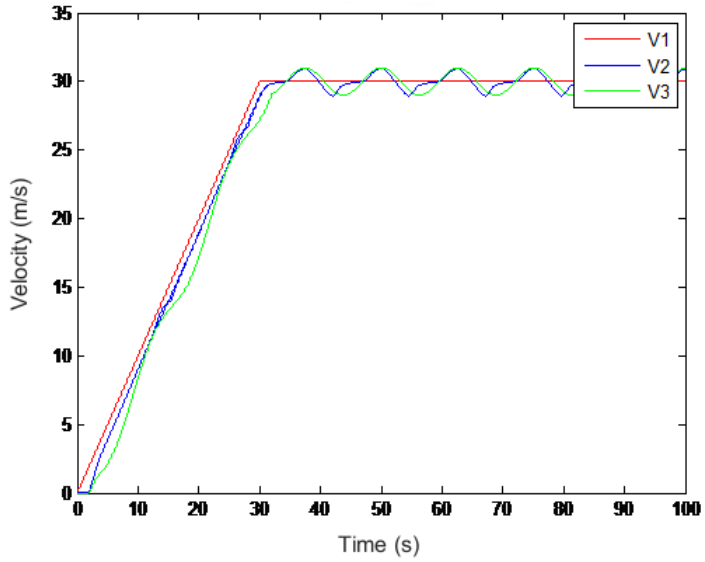


Figure 3.3: Velocity profiles of all three vehicles for case 1.

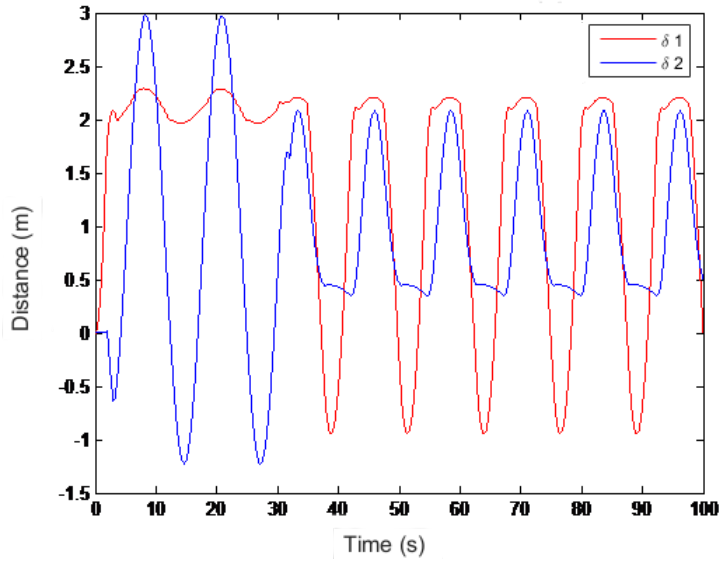


Figure 3.4: Position errors for the second vehicle for case 1.

Case 2 shows the need for MPC switching by temporarily disabling the 4-state MPC.

In the case that the third vehicle slows down or stops, the MPC still keeps adding that error to the cost function and tries to minimize it by increasing the spacing with the first vehicle. This is an undesired effect because it is important for us to minimize the total platoon length. Figures 3.5 and 3.6 shows what would happen if no switching is used and figures 3.7 and 3.8 show the actual controller with switching.

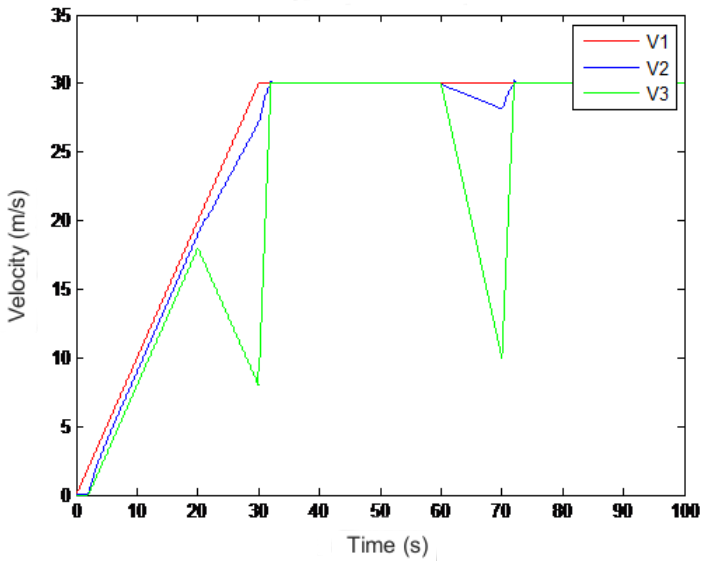


Figure 3.5: Velocity profile for case 2.

As can be seen from Fig. 3.6, even when the third car is very far behind the second vehicle, MPC still keeps optimizing the distance with respect to the third vehicle. Figures 3.7 and 3.8 show the behavior of the the cumulative MPC structure where the switching between two MPC structures allows activating unidirectional MPC once the third car gets sufficiently far. This effect is especially obvious after the 60th second, where the speed of the third vehicle drops down significantly.

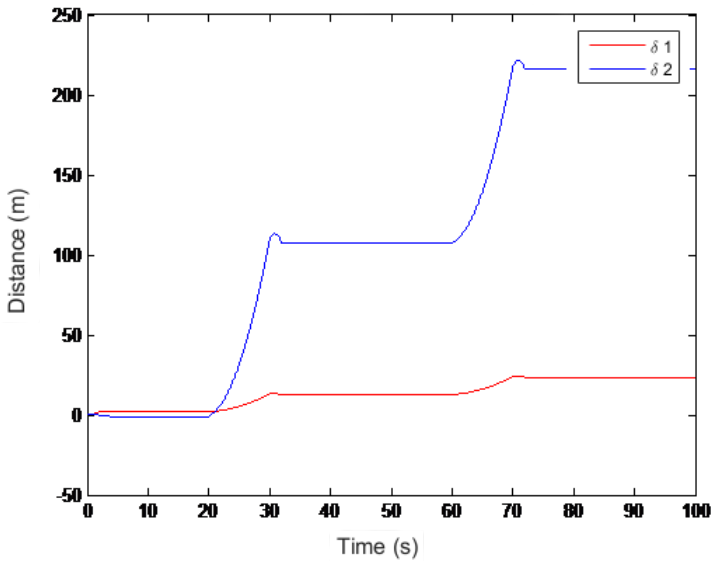


Figure 3.6: Position errors for the second vehicle for case 2.

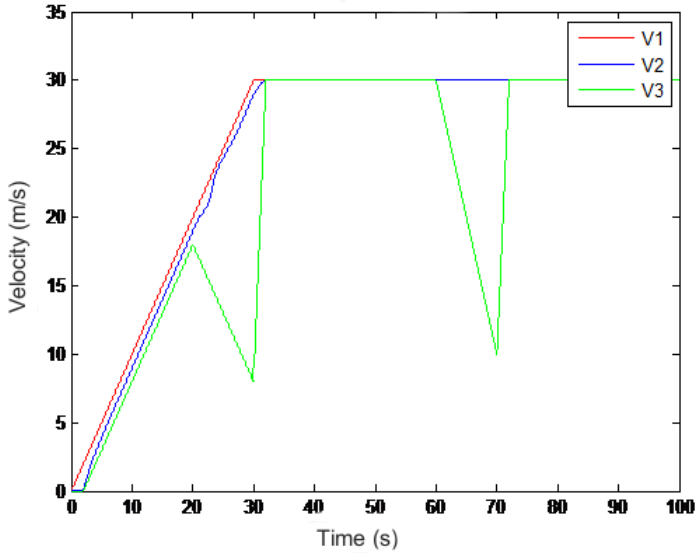


Figure 3.7: Velocity profile for MPC with switching)

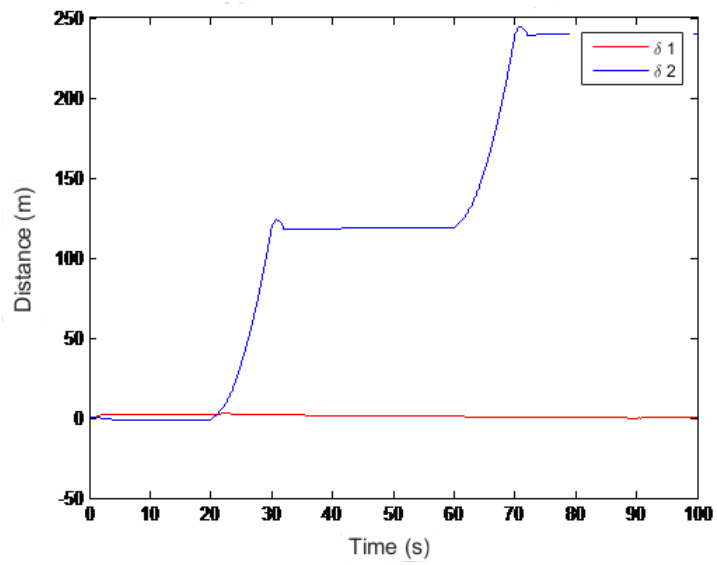


Figure 3.8: Position errors for the second vehicle for MPC with switching

3.4 Adaptive MPC Control Model

In the previous section, an MPC based CACC controller with soft constraints was introduced, and simulated using MATLAB's Simulink MPC toolbox. The results from the MPC controller were positive but the toolbox itself has a few disadvantages.

One of the biggest disadvantages is that the MPC plant model is constant throughout the simulation time. This is a problem if a parameter estimator is intended to be used along with the controller, which is important for real world implementation. One of the main reasons for using an MPC structure is to have a controller that is adaptable to changes in the car parameters and also to avoid the follower car rear-ending the ego-vehicle.

Another problem was that this was not implementable on the scaled model vehicles we use for testing. For this we needed an MPC controller that coded in C and that can be run on an embedded controller.

All the plant models used in this section are the same as those in sections 3.2 and 3.3; only the MPC structure itself is changed and a parameter estimator is added to the system.

3.4.1 Implementing MPC

The easiest way of implementing an MPC is to convert the problem into a Quadratic form and solve the quadratic problem using an open-source solver.

The generic form of a quadratic problem is:

$$\begin{aligned} &min \left\{\tfrac{1}{2}u^T H u + u^T g\right\}, \\ &s.t. \quad lbA \leq AX \leq ubA, \\ &\qquad lb \leq X \leq ub. \end{aligned} \tag{3.17}$$

Future states of the plant can be formulated as follows:

$$\begin{aligned} \begin{bmatrix} X_0 \\ X_1 \\ . \\ . \\ X_N \end{bmatrix} = \begin{bmatrix} B & . & . & . & 0 \\ . & . & & & . \\ . & & . & & . \\ . & & & . & . \\ A^{N-1}B & . & . & . & B \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ . \\ . \\ u_N \end{bmatrix} + \begin{bmatrix} A \\ A^2 \\ . \\ . \\ A^N \end{bmatrix} x(0) \\ + \begin{bmatrix} B_2 & . & . & . & 0 \\ . & . & & & . \\ . & & . & & . \\ . & & & . & . \\ 0 & . & . & . & B_2 \end{bmatrix} \begin{bmatrix} a_{i-1} \\ a_{i-1} \\ . \\ . \\ a_{i-1} \end{bmatrix}, \end{aligned} \tag{3.18}$$

where a_{i-1} represents the acceleration information coming from the preceding and the following vehicles. At each calculation cycle of the MPC this a_{i-1} is assumed to stay constant. Eq. 3.18 is substituted into Eq. 3.9 to get the implementable cost function. The cost function is in the form of $J = J_{cons} + J_{var}$. The quadratic problem only requires the J_{var}, the variable part that depends on the input to the plant. J_{var} has the following form for this MPC problem:

$$J_{var} = U^T \left(S^{u^T} Q S^u + R\right) + U^T \left(2X(0)^T S^{x^T} Q S^u + 2C^T Q S^u\right)^T, \tag{3.19}$$

where

$$S^x = \begin{bmatrix} A \\ A^2 \\ . \\ . \\ A^N \end{bmatrix}, S^u = \begin{bmatrix} B & . & . & . & 0 \\ . & . & & & . \\ . & & . & & . \\ . & & & . & . \\ A^{N-1}B & . & . & . & B \end{bmatrix},$$
$$C = \begin{bmatrix} B_2 & . & . & . & 0 \\ . & . & & & . \\ . & & . & & . \\ . & & & . & . \\ 0 & . & . & . & B_2 \end{bmatrix} \begin{bmatrix} a_{i-1} \\ a_{i-1} \\ . \\ . \\ a_{i-1} \end{bmatrix}. \tag{3.20}$$

3.4.2 Constraints on the System

One of the disadvantages of the MPC control structure described in Section 3.3 is that the softness of the constraints cannot be adjusted in real-time. In MATLAB's MPC toolbox, constraints are assigned a level of softness from 0 to 1. Level of softness is used to whether or not a constraint will be broken, or which constraint should be broken first. The MPC controller described in this chapter only allows for hard constraints; boundaries are used only for the essential states, which solved the problem. This turned out to give more control over the optimal controller, since many constraints made the cost function redundant as the controller tried to stay within the boundaries while ignoring the cost.

δ_i and δ_{i+1} are the deviations from the ideal spacing policy, and these should be zero for both the preceding and the following vehicles. A minimum value is defined for the preceding vehicle while the rest is left unbounded.

$$-h_i v_i/2 \leq \delta_i \quad m. \tag{3.21}$$

$\bar{v}_i$ and $\bar{v}_{i+1}$ are both left unbounded in order to ensure that the MPC will be able to find a solution without constraints being broken.

u is the input to the actuator and its limits are defined by the limits of the car itself. a is the actual acceleration of the car, and is also subjected to the same constraints.

$$\begin{aligned} &-4.5 \leq u \leq 2.5, \\ &-4.5 \leq a \leq 2.5 \quad m/s^2. \end{aligned} \tag{3.22}$$

v is the velocity of the ego-vehicle. It also has its own limitations, both coming from the actuator and the lateral controller, to maintain vehicle stability.

$$0 \leq v \leq 40 \quad m/s. \tag{3.23}$$

3.4.3 Parameter Estimator

One of the major shortcomings of the control structure from Section 3.3 is that it assumes the model of the car does not change over time, which is often not the case in real life. In order to solve this problem, a parameter estimator is added to the car model to estimate two of the defining parameters, although these parameters do not represent any actual physical car parameters. The generic car model is:

$$\begin{gathered} Car \quad model \quad G_i(s) = \frac{K_i}{s^2(\tau_i s+1)}, \\ x_i = \frac{K_i}{s^2(\tau_i s+1)} u_i, \\ x_i \tau_i s^2 + x_i s^2 = K_i u_i. \end{gathered} \tag{3.24}$$

Since the derivatives of x_i are unknown, both sides of the equation will be filtered with $\frac{1}{\Lambda(s)} = \frac{1}{(s+1)^3}$.

$$\frac{1}{(s+1)^3}[s_i \tau_i s^2] + \frac{1}{(s+1)^3}[x_i s^2] = \frac{1}{(s+1)^3}[K_i u_i]. \tag{3.25}$$

Using Eq. 3.25, static parametric model (SPM) is created, following the design guidelines presented in [75]:

$$\begin{aligned} z &= \theta^{*T} \phi \\ &= \frac{1}{(s+1)^3}[x_i s^2], \end{aligned} \tag{3.26}$$

$$\theta^* = \begin{bmatrix} \tau_i \\ K_i \end{bmatrix}, \phi = \begin{bmatrix} -\frac{1}{(s+1)^3}[x_i s^2] \\ \frac{1}{(s+1)^3}[u_i] \end{bmatrix}.$$

For this work, a least squares algorithm is used.

$$\begin{aligned} \hat{z} &= \theta^T \phi, \\ \epsilon(t) &= \frac{z(t) - \hat{z}(t)}{m_s^2(t)}. \end{aligned} \tag{3.27}$$

where $m_s^2(t) = 1 + \alpha\theta^T\theta$ and $\alpha = 0.1$

The adaptive law for the least-squares algorithm is as follows:

$$\begin{aligned} \dot{\theta}(t) &= P(t)\epsilon(t)\phi(t), \\ \dot{P}(t) &= \beta P(t) - P(t)\frac{\phi(t)\phi(t)^T}{m_s^2(t)}P(t), \end{aligned} \tag{3.28}$$

where the initial value of P(t)is $P(0)$:

$$p(0) = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}. \tag{3.29}$$

The control structure has the following form after the implementation of the parameter estimator:

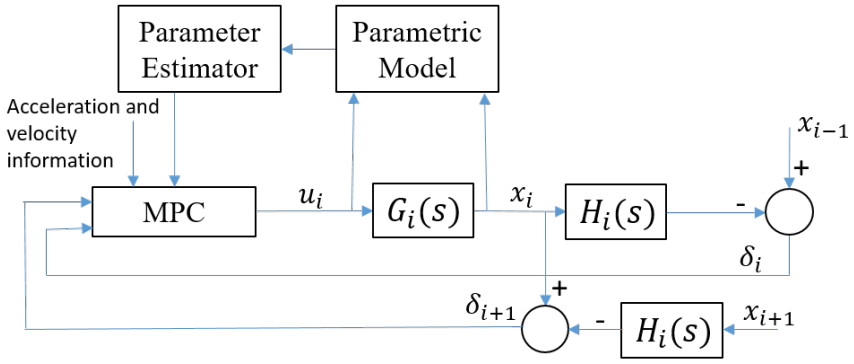


Figure 3.9: Control structure with parameter estimator.

3.4.4 Simulations

In the simulation results presented, the velocity profiles for the first and third vehicles are set manually; only the second vehicle is equipped with the MPC controller. In each

simulation, the first and third vehicles start accelerating with the same acceleration and keep constant speed after $t = 30$ sec. As before, the third vehicle has a small sinusoidal disturbance in its velocity profile. The following figures show the velocity profiles of all three vehicles, and how the second vehicle adjusts its velocity to keep a safe distance to both vehicles. This is the ideal case where the MPC knows the correct plant parameters and parameter estimation is not needed. The plant parameters for all simulations are Kg=1 and τ=0.4. All the simulations presented in this section, start from the initial resting positions.

Fig. 3.10 shows the velocity profiles of all three of the vehicles while the the Fig. 3.11 shows the spacing error of the second vehicle with respect to both front and rear targets.

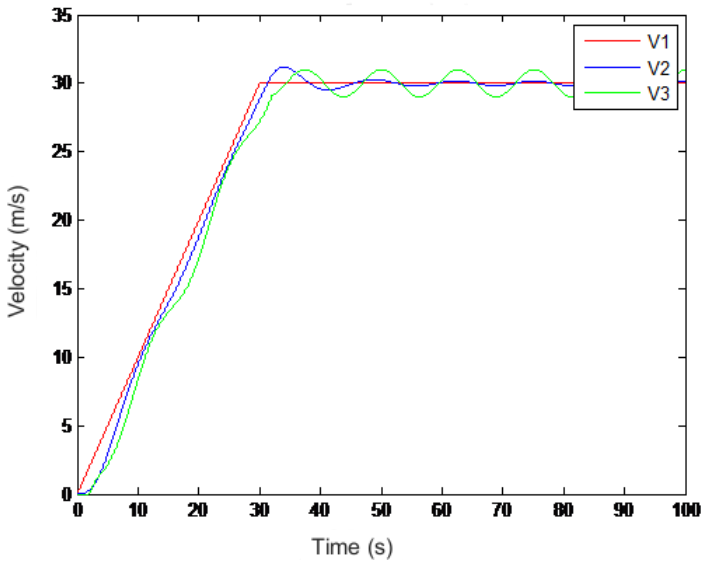


Figure 3.10: Velocity profiles.

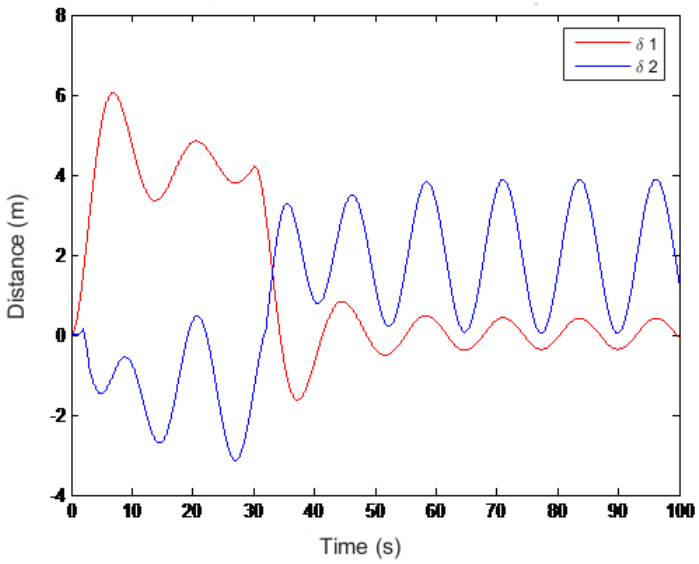


Figure 3.11: Position errors for the second vehicle.

Simulation presented above shows that MPC structure gives good results if the correct plant parameters are known ahead of time. Next, effects of MPC having wrong plant parameters will be discussed. Fig. 3.12 & 3.13 show how the controller will perform if the MPC has the wrong plant parameters. For this simulation, the MPC plant parameters are Kg=0.9 and τ=1, as opposed to the actual plant parameters Kg=1 and τ=0.4. The following two figures show the effects on position errors for the preceding and following vehicles for both the correct and incorrect parameter values:

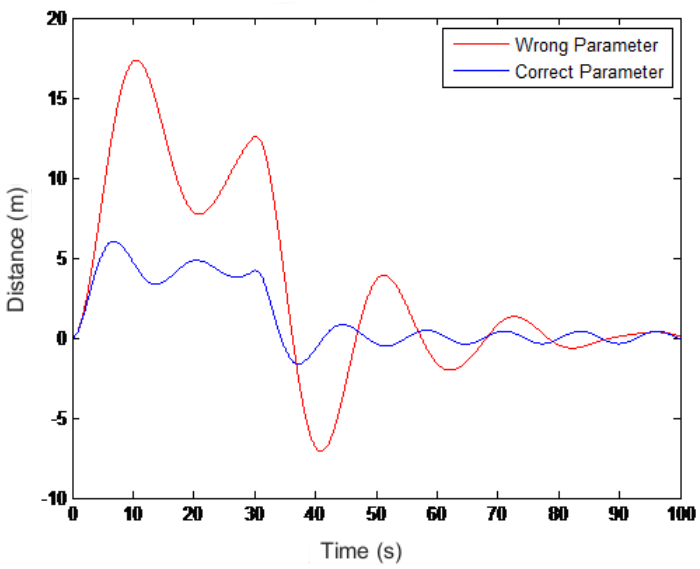


Figure 3.12: Position error with preceding vehicle.

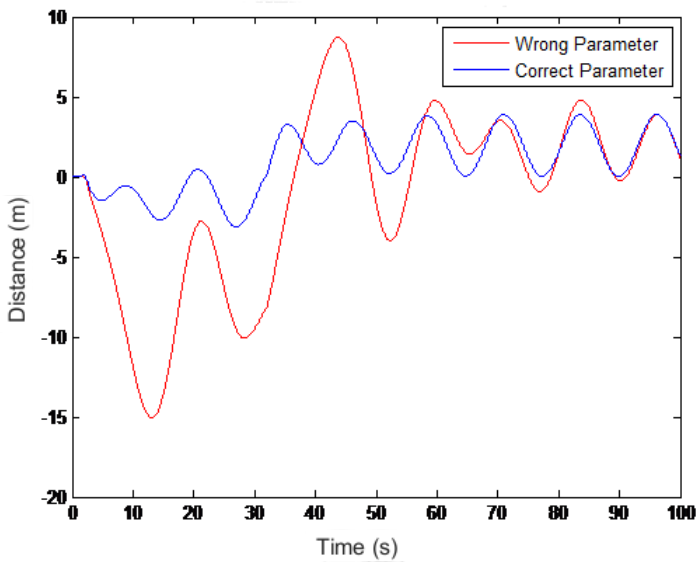


Figure 3.13: Position error with following vehicle.

As can be seen from the above figures, it is important that the MPC has the correct

parameters in order to give good results. In order to do this, the parameter estimator is tested. Fig. 3.14 shows the parameter estimator settling to the correct values, which are Kg=1 and τ=0.4.

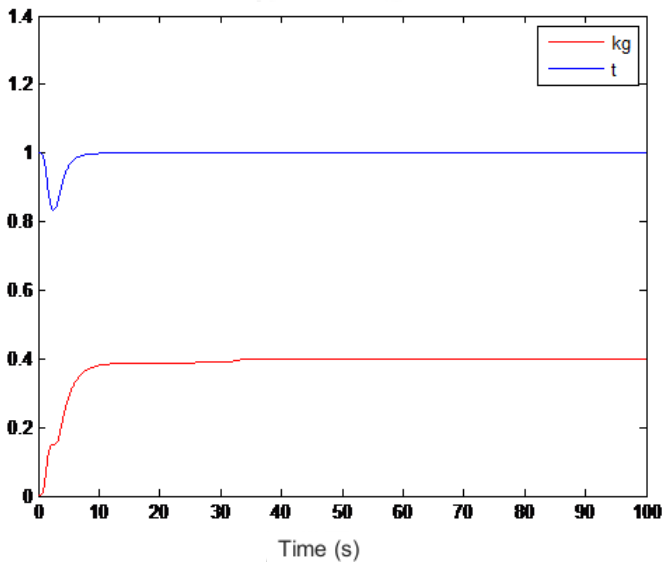


Figure 3.14: Parameter estimator.

When the parameter estimator is combined with the actual MPC controller the results are as shown below:

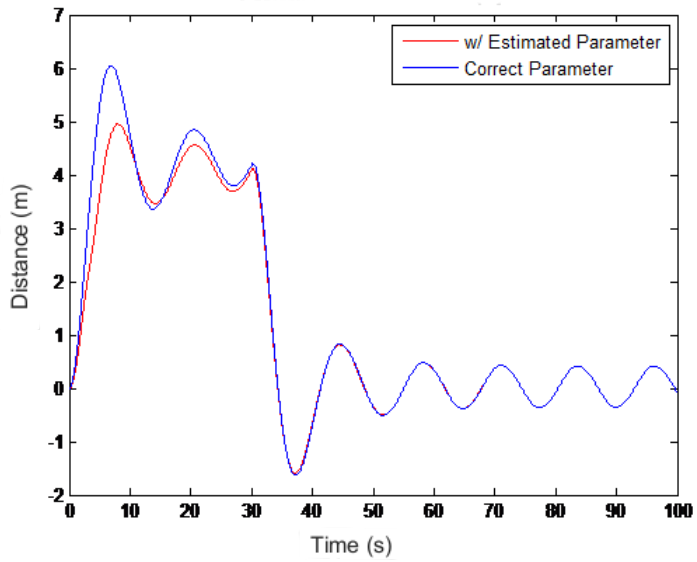


Figure 3.15: Position error with preceding vehicle.

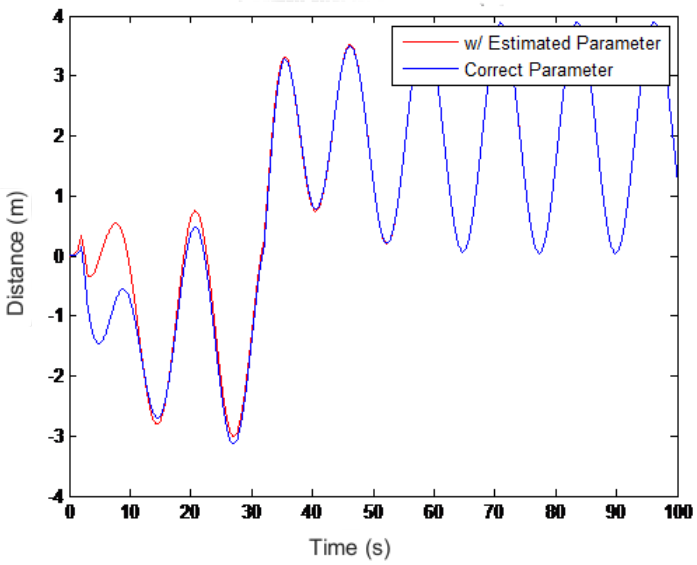


Figure 3.16: Position error with following vehicle.

It can be seen from Fig. 3.15 & 3.16 that it takes around ten seconds for the parameters

to converge to the real values. This is happening due to fact that simulations are started in an state where the parameter estimators initial values are very different from the vehicles actual values. In real-time experiments, changes in these two parameters will be happening gradually. Time required to converge to the vehicles parameters when the changes are gradual will be a lot faster than the results shown in this section.

3.5 Summary and Remarks

In this chapter, we have investigated the use of a bi-directional CACC controller which tries to avoid rear end collisions with the following vehicle by breaking the spacing policy with the preceding vehicle just enough to avoid a collision, while making sure that no real danger is caused to the preceding vehicle.

The first approach was based on the MATLAB MPC Toolbox. This did not allow the plant model to be changed once the simulation is started, which is needed in order to implement the parameter estimator. An advantage of this toolbox is the ability to use softened constraints. Initial results from the simulations were satisfactory, which shows that MPC control structure works well for this purpose.

The second approach has used a quadratic problem solver to create a design which is implementable and compatible with a parameter estimator. Simulations show that incorrect plant dynamics might lead to poor control action from the MPC. To eliminate this problem, the use of parameter estimator was proposed. Simulation results showed that with use of the parameter estimator, the control response was greatly improved even if the initial parameter estimates are incorrect, which makes this structure more reliable.

The advantages of the MPC based control structure are:

1. Its ability to put constraints on the controller, which means that the controller knows the limitations of the plant and the outputs of the controller will always be within these limits,

2. Most other control structures are finely tuned for a particular case and disregard future changes in the plant model. MPC can be combined with a parameter estimator to track the changes in the plant and ensure that the control keeps up with these changes,

3. The use of constraints allows the gap with the preceding vehicle to have a minimum value, while there is no constraint for the gap with the follower vehicle.

The disadvantages of this structure are:

1. Complexity of implementation and tuning,
2. In order to get optimal solutions from the MPC, the plant model should be perfectly known, which requires a robust parameter estimator that will give the correct parameters at all times,
3. To implement MPC real-time, higher computation power is required compared to most control structures (eg. PD).

Even though the second controller design can be implemented in real time, most car manufacturers prefer to use controllers that are computationally less intensive and easier to tune. The next chapter introduces an alternative to the MPC-based control structure.

Chapter 4

Switching Control Based CACC structure with Rear-End Collision Check

Chapter 3 introduced a new kind of CACC control structure that considers the spacing to the following vehicle to avoid rear end collisions. To accomplish this, an MPC has been employed in the previous chapters, but it has certain drawbacks such as high computational intensity. In this chapter, a similar problem will be addressed using a PD-based switching controller.

4.1 Problem Definition

This problem considers a platoon of N vehicles longitudinally following each other. V_1 is the lead vehicle of the platoon and is controlled manually (as in, its velocity profile is predefined). It is desired to keep a certain distance $D_{i,i+1}$ between each vehicle $x_i - x_{i+1}$. The value of $D_{i,i+1}$ can either be constant or be based on the velocity of the vehicle V_i; both of these cases will be considered in this chapter. This platooning problem will be formally formulated within the framework of the switching control problem as presented by Fidan and Anderson [76], in which a switching controller was designed for platoons with multi-agents to maintain desired spacing under a certain level of noise. The switching controller was originally designed for velocity controlled agents, but it can be adapted for acceleration controlled systems. Simulations will be done for both velocity controlled and acceleration controlled vehicles.

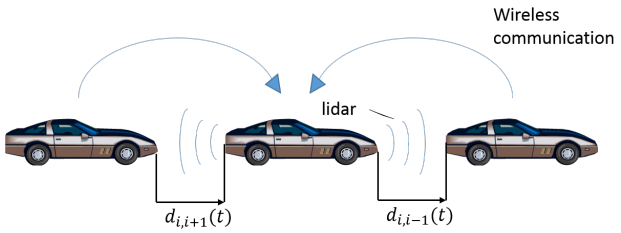


Figure 4.1: Bidirectional CACC structure.

4.2 Modeling of Car Dynamics and Platoon

Similar platoon dynamics to those presented in Section 2.1 are used in this chapter. The car model is defined in the form:

$$G_i(s) = \frac{K_i}{s(\tau_i s+1)}, \tag{4.1}$$

where τ_i represents the closed-loop bandwidth. This plant model is used for both the velocity and acceleration controlled vehicle.

Fig. 4.2 shows how the platoon formation is defined. It is assumed that there is a lead vehicle, which is controlled manually (predefined velocity profile).

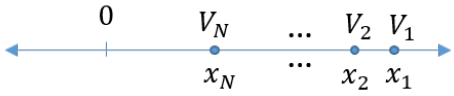


Figure 4.2: Illustration of the platoon.

Considering a platoon of N vehicles, vehicle 1 being the lead vehicle and vehicle i being $i-1$ vehicles behind the lead vehicle. Each vehicle is again assumed to have 2 lidar sensors, one located in front and one at the back. The relative distance measured from vehicle i to vehicle j (for both cases where j is $i-1$ or $i+1$) is defined as

$$d_{ij}(t) = x_i(t) - x_j(t) + \eta_{m_{ij}}(t), \tag{4.2}$$

where $\eta_{m_{ij}}$ represents measurement noise. This noise is assumed to be bounded by some known $\bar{\eta_m} > 0$. Spacing policy is defined as D_{ij}. It can be constant-spacing or constant time-headway depending on the case. The spacing error is defined as the following:

$$\delta_{ij} = d_{ij} - D_i. \tag{4.3}$$

4.3 Constant Spacing Policy

4.3.1 Base Design for Velocity Integrator Model

In this section, a control structure for velocity controlled vehicles is presented. This controller requires the velocity of the preceding vehicle to be transmitted wirelessly (which could be via an ad-hoc network).

Control Structure

As mentioned above, the control structure proposed in this chapter utilizes the switching control structure introduced in [76] with an additional feed-forward component to compensate for the lead vehicle of the platoon being in motion. The controller structure is shown in Fig. 4.3.

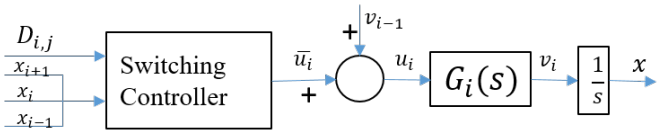


Figure 4.3: Control Structure for velocity integrator model.

Considering the simple dynamic model (4.1) with a velocity input, the control signal is selected as

$$u_i = v_{i-1} + \bar{u}_i, \tag{4.4}$$

where v_{i-1} is for compensating the bias due to the velocity of the platoon. The control action is calculated as follows:

$$\bar{u}_i = -\sigma(\delta_{i,i-1}) - \sigma(\delta_{i,i+1}). \tag{4.5}$$

The switching function σ is defined in the following manner [76]

$$\sigma(\delta) = \begin{cases} 0 & if \quad |\delta| \leq \bar{\eta_m} \\ \bar{k}\frac{\delta - \bar{\eta_m}}{\bar{\delta}}\delta & if \quad \bar{\eta_m} < |\delta| \leq \bar{\eta_m} + \bar{\delta} \\ \bar{k}\delta & if \quad |\delta| > \bar{\eta_m} + \bar{\delta} \end{cases} \tag{4.6}$$

where $\bar{\eta_m} > 0$ is the known upper bound on the distance measurement noise and $\bar{\delta} > 0$ is a design constant. $\bar{k} > 0$ is the proportional control constant.

Simulation

The proposed controller is simulated in Matlab. The velocity profile of the first vehicle is predefined, while all other vehicles in the platoon are equipped with the controller. The velocity profile of the lead vehicle is shown in Fig. 4.4.

The desired spacing and the noise levels for all vehicles are set as 10m and 0.1m respectively. The controller constants were $\bar{\eta_m} = 0.1$, $\bar{\delta} = 0.02$ and $\bar{k} = 3$. The vehicle parameters are $K = 1$ and $\tau = 0.4$.

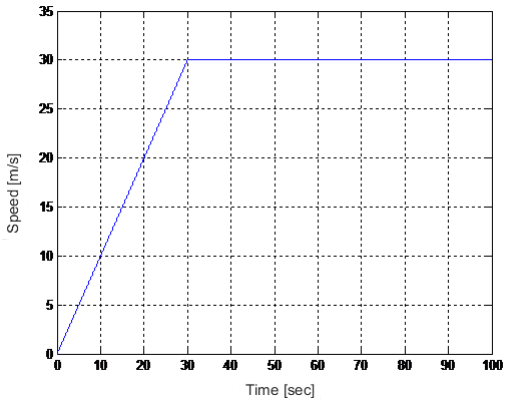


Figure 4.4: Velocity profile of the lead vehicle.

Fig. 4.5 shows the position error, which can be found by $\delta_{ij}(t) = d_{ij}(t) - D_{ij}(t)$. It can be seen that the error quickly converges to a low value. Also the use of equal controller gains for the front and rear vehicles causes high transient position errors for vehicles closer to the leader. This problem is caused by equal weighting and will be further discussed in the following sections.

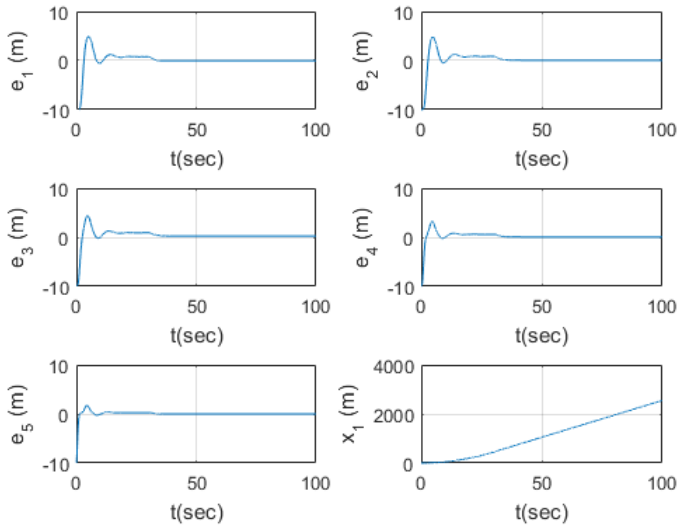


Figure 4.5: Position errors with respect to the leader for all five followers and the position of the leader

4.3.2 Acceleration Integrator Vehicles

In order for this controller to be tested with a realistic simulator or on a real vehicle, it requires an acceleration command input. The dynamic model is similar to the previous model (4.1) with the addition of an integrator.

$$G_i(s) = \frac{K_i}{s^2(\tau_i s + 1)}. \tag{4.7}$$

Control Structure

Ideally the derivative of the controller Eq. 4.4 with respect to time should give the same results as the previous case.

$$\begin{aligned} \frac{du_i}{dt} &= \frac{d\bar{u}_i}{dt} + \frac{dv_{i-1}}{dt}, \\ &= \frac{d\bar{u}_i}{dt} + a_{i-1}. \end{aligned} \tag{4.8}$$

The term $\frac{d\bar{u}}{dt}$ can be differentiated partially as follows:

$$\frac{d\bar{u}_i}{dt} = -\frac{dk(\delta)}{d\delta}\dot{\delta}_{i,i-1} - \frac{dk(\delta)}{d\delta}\dot{\delta}_{i,i+1}, \tag{4.9}$$

where $\frac{dk(\delta)}{d\delta}$ is defined as:

$$\frac{dk(\delta)}{d\delta} = \begin{cases} 0 & if \quad |\delta| \leq \bar{\eta_m} \\ \frac{\bar{k}}{\bar{\delta}}(2\delta\dot{\delta} - \bar{\eta_m}\dot{\delta}) & if \quad \bar{\eta_m} < |\delta| \leq \bar{\eta_m} + \bar{\delta} \\ \bar{k}\dot{\delta} & if \quad |\delta| > \bar{\eta_m} + \bar{\delta} \end{cases} \tag{4.10}$$

This form of Eq. 4.10 is discontinuous. In order to have a continuous acceleration profile, the following form is implemented:

$$\frac{dk(\delta)}{d\delta} = \begin{cases} 0 & if \quad |\delta| \leq \bar{\eta_m} \\ \frac{\bar{k}}{\bar{\delta}}(\delta\dot{\delta} - \bar{\eta_m}\dot{\delta}) & if \quad \bar{\eta_m} < |\delta| \leq \bar{\eta_m} + \bar{\delta} \\ \bar{k}\dot{\delta} & if \quad |\delta| > \bar{\eta_m} + \bar{\delta} \end{cases} \tag{4.11}$$

Simulation

As before, the desired spacing and noise levels for all vehicles are set as 10m and 0.1m respectively. The controller constants were $\bar{\eta_m} = 0.1$, $\bar{\delta} = 0.02$ and $\bar{k} = 3$.

It is evident from a visual comparison of Fig. 4.5 and Fig. 4.6 that this controller does not perform as well as the velocity-controlled version, and this will be addressed next.

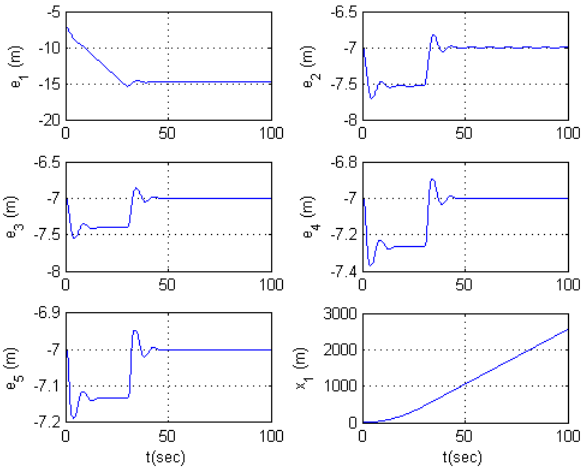


Figure 4.6: Position errors with respect to the leader for all five followers and the position of the leader.

4.3.3 Modified Control Structure for Acceleration Integrator Model

Control Structure

As can be seen from Fig. 4.6, the control structure for acceleration controlled vehicles cannot keep the desired spacing between vehicles. Rather than using the derivative of the velocity-based control as input Eq. 4.4, a PD tracking controller will be utilized to keep the desired velocity given by equation Eq. 4.4.

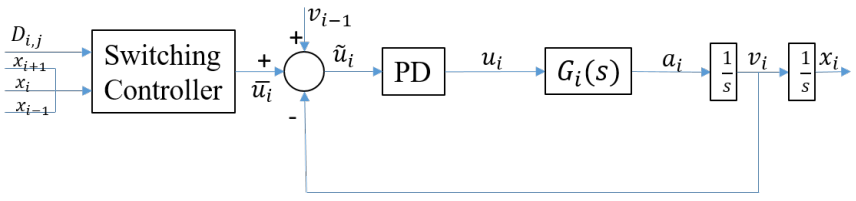


Figure 4.7: Modified control structure.

Fig. 4.7 shows the modified control structure for the acceleration controlled vehicle. The output of the switching controller will be the same as Eq. 4.5. The desired speed

of the vehicle will be given as $\bar{u} + v_{i-1}$, and the input to the PD controller will be the deviation from the desired velocity.

$$\tilde{u}_i = \bar{u} + v_{i-1}. \tag{4.12}$$

The PD controller is in the form:

$$PD(s) = k_p + k_d s. \tag{4.13}$$

Simulation

The same configuration as used in Section 4.3.1 is used for this simulation. Fig. 4.8 shows that results using this control structure are as good as those shown in Fig. 4.5.

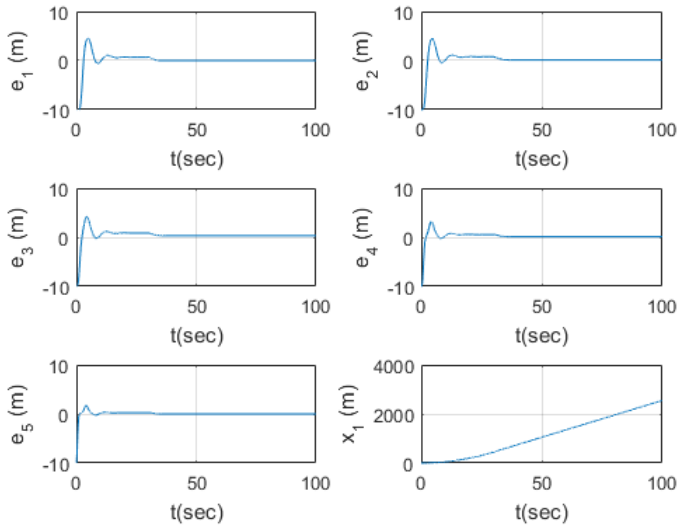


Figure 4.8: Position errors with respect to the leader for all five followers and the position of the leader.

4.4 Variable Spacing Policy

Following the findings presented in Section 2.4.1, throughout the thesis where bidirectional string stability is used, front and rear spacing policies will be set by the speed of the ego-vehicle.

4.4.1 Base Design for Variable Spacing

In this section, instead of using a constant spacing between vehicles, a variable spacing policy will be used for velocity controlled vehicles.

Control Structure

The switching control structure of this Section is the same as Section 4.3.1. The output of the switching controller is fed to the PD controller, and the desired spacing policy is defined as follows:

$$|D_i| = h_i v_i. \tag{4.14}$$

As discussed in the previous section, the desired spacing described by Eq. 4.14 is used for calculating the spacing error for both the preceding and following vehicles.

The control structure for velocity controlled vehicles with variable spacing is shown in Fig. 4.9.

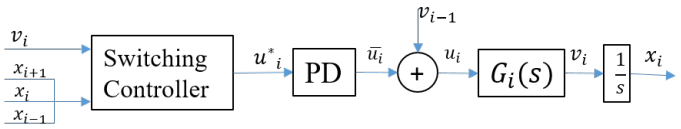


Figure 4.9: Base design for variable spacing.

The controller output takes the following form:

$$\bar{u}_i = -\hat{\sigma}(\delta_{i,i-1}) - \hat{\sigma}(\delta_{i,i+1}), \tag{4.15}$$

where a modified switching function $\hat{\sigma}$ is defined as:

$$\hat{\sigma}(\delta) = \begin{cases} 0 & if \quad |\delta| \leq \bar{\eta_m} \\ \frac{\bar{k}}{\bar{\delta}}(k_p(\delta - \bar{\eta_m})\delta + k_p\dot{\delta}(2\delta - \bar{\eta_m})) & if \quad \bar{\eta_m} < |\delta| \leq \bar{\eta_m} + \bar{\delta} \\ \bar{k}(k_p\delta + k_d\dot{\delta}) & if \quad |\delta| > \bar{\eta_m} + \bar{\delta} \end{cases} \tag{4.16}$$

4.4.2 Modified Control Structure for Acceleration Integrator Model with Variable Spacing Policy

The control structure proposed in Section 4.4.1 is modified in a similar fashion to 4.3.3 to work with acceleration controlled vehicles.

Control Structure

The control structure for an acceleration controlled vehicle with variable spacing is shown in Fig. 4.10.

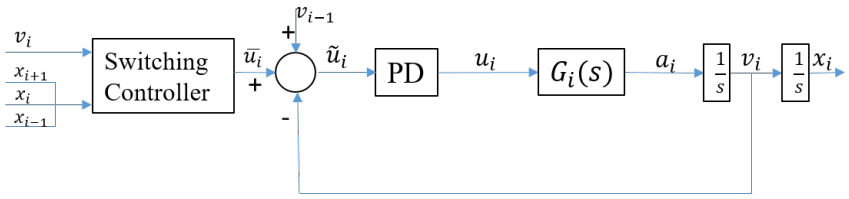


Figure 4.10: Controller design for acceleration integrator model with variable spacing.

The switching control functions takes the following form:

$$\bar{u}_i = -\sigma(\delta_{i,i-1}) - \sigma(\delta_{i,i+1}), \tag{4.17}$$

where a modified switching function σ is defined as in Section 4.3.1. This output is combined with the feed-forward component and the negative of the current velocity to create the velocity error.

$$\tilde{u}_i = \bar{u} + v_{i-1}. \tag{4.18}$$

The deviation from the controlled velocity is fed into the PD controller to get the acceleration input.

4.4.3 Comparison with MPC-based Control Structure

In order to see how this controller performs, the same simulation done for the MPC control structure is repeated. The variable spacing method described in Section 4.4.1 is used. Fig. 4.11 shows the velocity profiles of all three vehicles. As with the MPC simulations, the velocity profiles of the $1st$ and $3rd$ vehicles are predefined, while only the second vehicle is equipped with the switching-control structure.

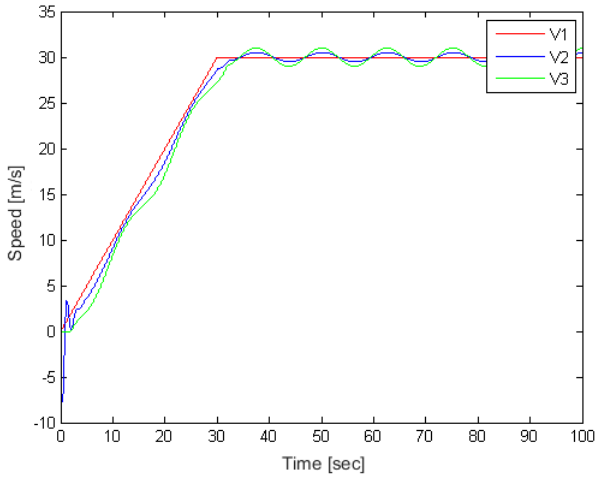


Figure 4.11: Velocity profile of the leader.

Comparing these simulation results with those from Section 3.3.2, it is seen that the switching-based control structure proposed in this chapter performs similarly to the MPC control structure. The results presented in this section are generated with a symmetric control structure, whereas the results from Section 3.3.2 use different weightings for the two vehicles. The next section will investigate the asymmetric control structure.

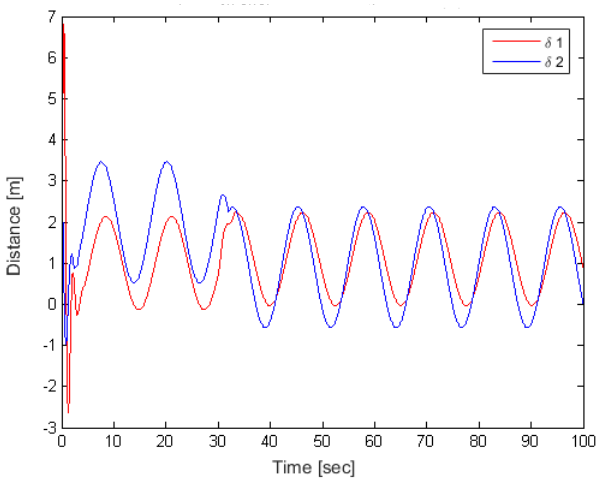


Figure 4.12: Position error.

4.5 Asymmetric Bidirectional CACC

Based on the findings of Niewenhuijze [35] and Herman [77], using an asymmetric control structure, where the front and rear vehicles have different weights, increases the string stability of the platoon and lowers the required minimum time-headway.

To test the effects of different weightings, a simulation scenario with a lead vehicle and five followers is created in PreScan. PreScan is a physics-based simulation platform created for development of connected advanced driver assistance systems. All vehicles are equipped with wireless communication and communicate their positon, velocity and acceleration information. This software allows usage of both high-fidelity vehicle models or user defined models. Unless it is explicitly defined, vehicle models used in simulations are the ones presented in 2.1. All simulations are done with the same leader velocity profile. Simulations presented in this section start from the initial stop positions.

The weights of the front and rear vehicles are denoted as W_f and W_r.

4.5.1 Baseline: Unidirectional CACC Structures

The CACC structure presented in [3], which is also explained briefly in Section 2.1, is used as the baseline.

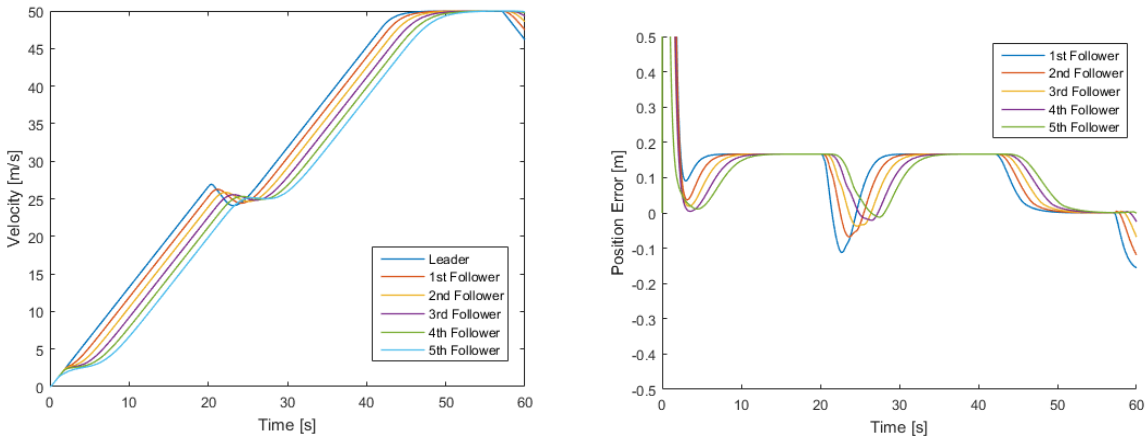


Figure 4.13: Velocity and position error plots for baseline configuration.

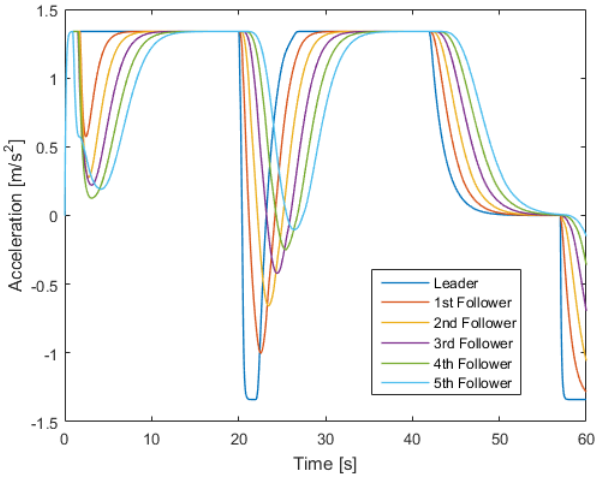


Figure 4.14: Acceleration plot for baseline configuration.

Different asymmetric weighting configurations will be compared with this baseline structure to find the optimal weighting, which has the response time of the baseline structure while utilizing the benefits of a bidirectional control structure.

4.5.2 $W_f = 1$ and $W_r = 0$: Unidirectional Switching Controller

The first simulation done with the switching control structure just uses the front vehicles data, as if it is a unidirectional structure.

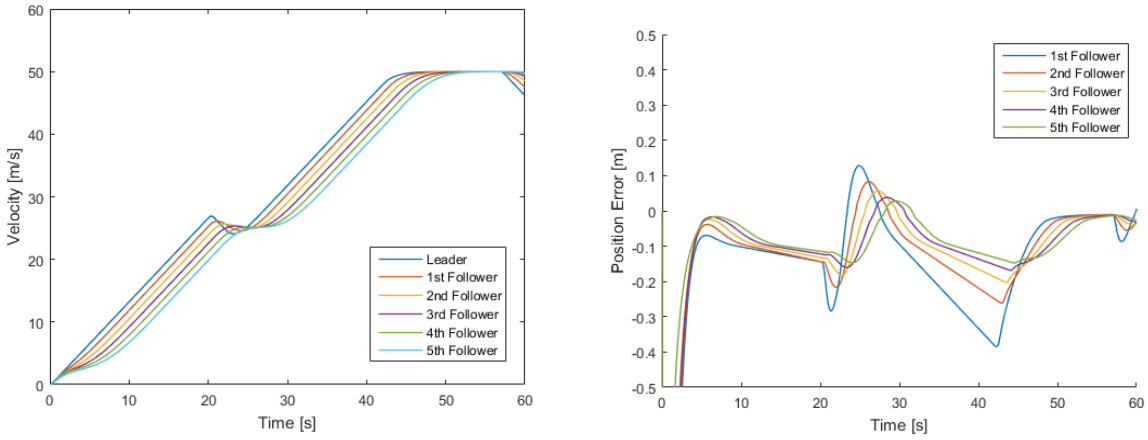


Figure 4.15: Velocity and position error plots for $W_f = 1$ and $W_r = 0$ configuration.

Compared to the baseline design the switching controller has higher spacing error during transients, but this error is attenuated as it propagates back. The steady state error is as low as that for the baseline.

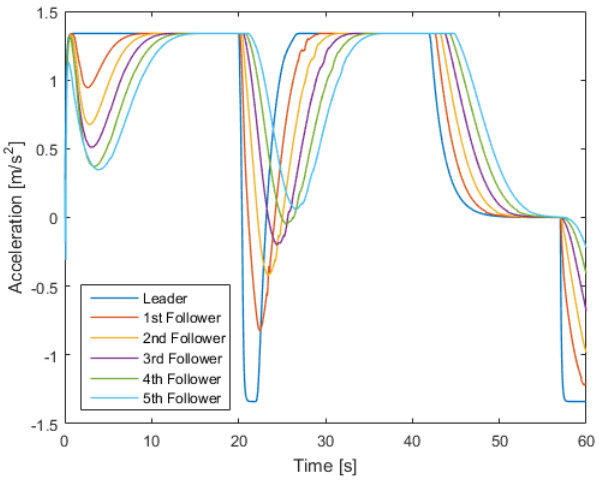


Figure 4.16: Acceleration plot for $W_f = 1$ and $W_r = 0$ configuration.

Comparing acceleration profiles it can be seen that the switching based structure maintains a higher degree of string stability than the baseline structure

4.5.3 $W_f = 1$ and $W_r = 0.3$

Next a 1-0.3 configuration is tested where the front vehicle has about 3 times more weight than the rear vehicle.

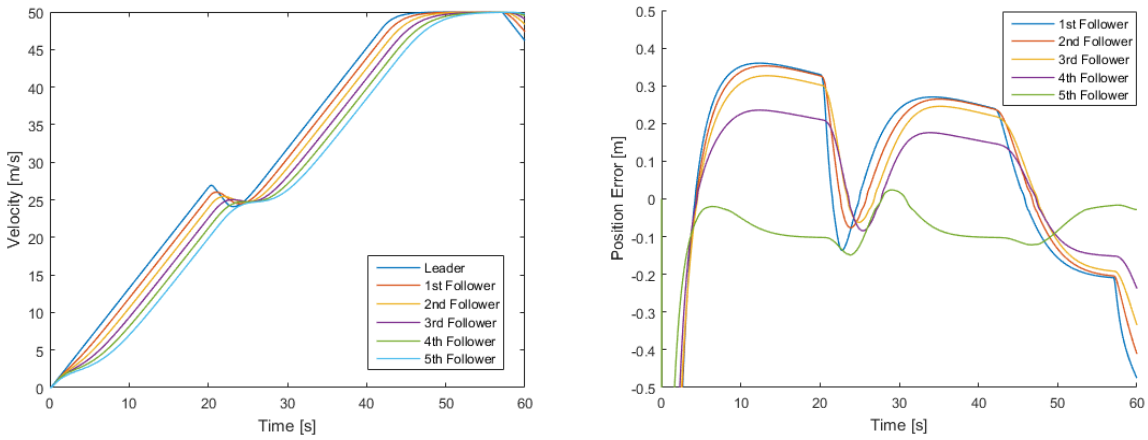


Figure 4.17: Velocity and position error plots for $W_f = 1$ and $W_r = 0.3$ configuration.

While the spacing error during the transients are the same as with the unidirectional structure, the state state error for the first few vehicles increases. Similar to the previous structure, this steady state error does not increase as it propagate backwards and is attenuated to almost zero steady state error as it reaches the last vehicle in the platoon.

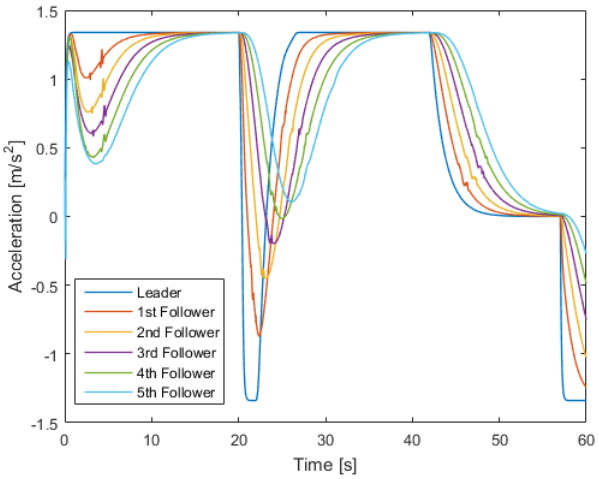


Figure 4.18: Acceleration plot for $W_f = 1$ and $W_r = 0.3$ configuration.

Looking at the acceleration profiles, string stability is stronger in this asymmetric bidirectional switching structure compared to the unidirectional structure as discussed in the previous chapter. Acceleration profile of the tail vehicle stays lower compared to the baseline structure, also response time is faster.

4.5.4 $W_f = 1$ and $W_r = 1$

Finally results from the symmetric bidirectional structure will be tested in the same scenario.

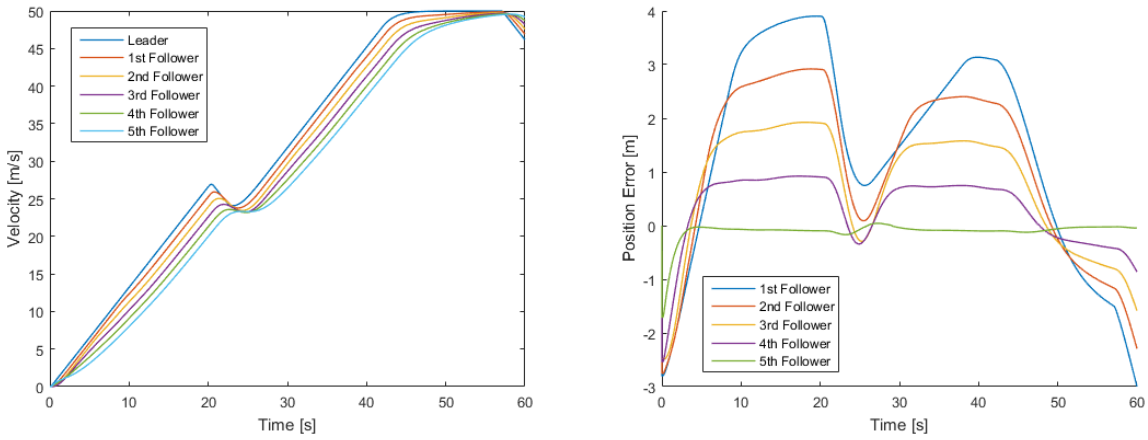


Figure 4.19: Velocity and position error plots for $W_f = 1$ and $W_r = 1$ configuration.

It is evident that both the transient and steady-state errors have increased significantly, although again these errors are attenuated as they propagate backwards. Having equal weights results in the tail vehicle of the platoon, which must use a unidirectional structure since no follower exists, dominating the platoon.

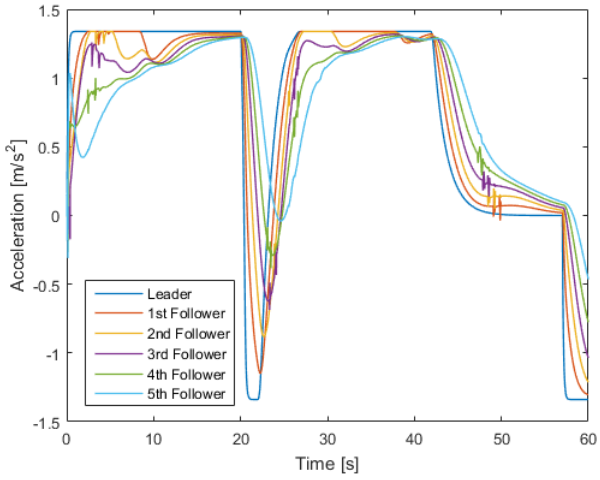


Figure 4.20: Acceleration plot for $W_f = 1$ and $W_r = 1$ configuration.

Even though the string stability is as strong as the baseline structure, 1-0.3 structure was able to maintain a stronger string-stability.

4.5.5 Simulation with High-Fidelity Vehicle Dynamics

The asymmetric bidirectional CACC structure, where vehicle weighting is $W_f = 1$ and $W_r = 0.3$, is further investigated by using a higher fidelity vehicle dynamics. Since no lateral dynamics will be used in these simulations, 2-D bicycle model is utilized for lateral dynamics. Longitudinal dynamics used for this model has realistic engine and transmission dynamics, which is connected to the chassis. Details of this model can be found in [78]. A similar six vehicle platoon configuration is maintained for this simulation, while a different velocity profile is utilized

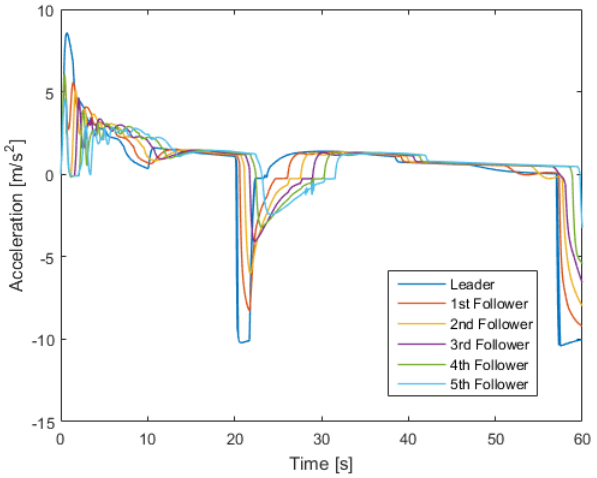


Figure 4.21: Acceleration plot for high-fidelity vehicle dynamics.

Fig. 4.21 shows the acceleration profiles of the vehicles in the platoon. By looking at the instances where the platoon starts acceleration from the start position and around the 20th second, it can be seen that platoon still maintains a string stable configuration through the simulation.

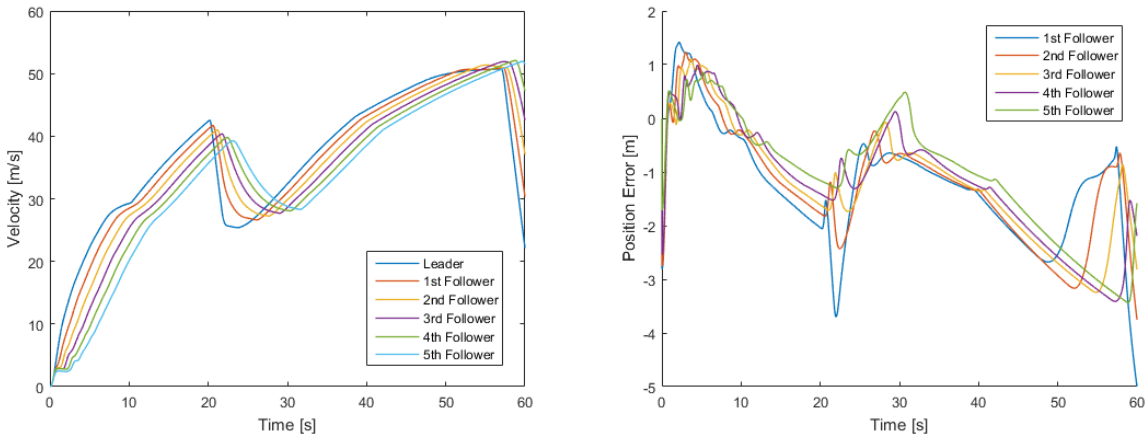


Figure 4.22: Velocity and position error plots for high-fidelity vehicle dynamics.

Position errors for this simulation, as shown in Fig. 4.22, indicates that higher spacing errors are encountered with realistic actuator dynamics. These higher errors were expected for this configuration and considering the desired spacing at those moments, position errors never exceed 10% of the desired spacing, which indicate that this controller will be well suited for real-life use.

4.5.6 Remarks

This section presented various configurations of asymmetric bidirectional switching structure, although not all possible configurations were presented. Testings done with these configurations led to the conclusion that having the weight of the front vehicle around three times the weight of the rear vehicle gives the best performance. That configuration maintains the strongest string stability and highest response time, while having least steady-state and transient position errors.

The $W_f = 1$ and $W_r = 0.3$ configuration is tested with high-fidelity vehicle dynamics and the results obtained suggests that this control structure works also well in environments with changing plant dynamics and higher actuator delays.

It can be seen that in all simulation configurations, spacing errors are highest for the vehicle directly following the platoon leader and it decreases as it propagates upstream, similar to the acceleration profile. This indicates that when this control structure is used in larger platoons, error will be always bounded.

4.6 Summary and Remarks

This chapter has presented a switching based bidirectional CACC structure as an alternative to the MPC-based bidirectional CACC structure presented in Chapter 3. The main goal in designing this controller is to have a more robust CACC structure which can compensate for faulty controller action of the follower vehicles. Also this controller structure has utilized the switching technique in order to minimize the effects of communication noise. Some of the advantages of this control structure are:

1. Easier to implement compared to MPC,
2. Stability of this switching-function is mathematically proven [76],
3. Use of the switching function allows the control structure to be robust to sensor noise.

The disadvantages of this controller are:

1. Unlike MPC, this controller cannot account for the limitations of the actuators,
2. The control structure cannot adapt to changing plant parameters.

Section 4.4.3 compared the control structure in this chapter to the MPC control structure from Chapter 3, where both controllers perform equally well. However, the switching based structure is easier to implement and requires less computational power.

The effects of asymmetric weighting have also been discussed in this chapter. It was concluded that having the weight of the front vehicle higher than that of the rear vehicle both makes the string stability stronger while lowering the steady-state and transient position errors. This structure was also compared with a highly-regarded unidirectional CACC structure from [3], in which the asymmetric bidirectional switching structure performs better overall.

Chapter 5

Adaptive Vehicle-Lane Merging with Obstacle Avoidance

In [1], two types of traffic jams were identified: Ghost traffic jams, and jams caused by inadequacy of the road or bad driving strategies. Ghost traffic jams can be solved using a string stable control structure, while the second type of traffic jams require better drivers and road structure. In this chapter and the next we address some causes and solutions to the second type of traffic jams via designing control schemes for vehicle-lane merging.

The existing body of literature on vehicle-lane merging provides different protocols for highway on-ramp merging [47, 79, 49]. However, a major issue with these studies is that they are designed specifically for the on-ramp merging problem, where the vehicles on the main carriageway approaching the on-ramp already recognize that the vehicles on the ramp will be merging with the main carriageway. This is a subset of the more general merging problem, where a random vehicle on the road needs to merge into the traffic in an adjacent lane. The major issue in a scenario with mixed driver controlled vehicles together with autonomous vehicles is that the autonomous vehicles will not recognize the merging intention unless it is communicated via turn signal or wireless communication, or the vehicle in the destination lane of the merging vehicle constantly scans its environment to asses the intentions of its immediate neighbors.

Interaction protocols for more general vehicle-lane merging tasks have gained interest in the past few years [53, 71, 73, 72]. However, these studies require extensive use of communication to inform the intentions of vehicles and to coordinate the merging.

In this chapter, we study a more general form of the lane merging problem in which only minimal communication is required between neighboring vehicles: position, velocity and

acceleration information. A merging scheme, where the vehicle in the main lane detects the intentions of neighboring vehicles based on their state parameters, is formulated. This chapter mainly focuses on detecting and reacting to a single join maneuver.

5.1 Problem Definition

A realistic driver behavior is modeled in this chapter, assuming that vehicles do not send any request messages for merging. In real life, during cases where turn signals are not working, when a vehicle starts getting close to the ego-vehicle, the ego-vehicle observes the vehicle to see if it wants to merge with the ego-vehicle's lane. If this is determined to be the case, the ego-vehicle tries to open up a gap, while maintaining safe distances with other surrounding vehicles.

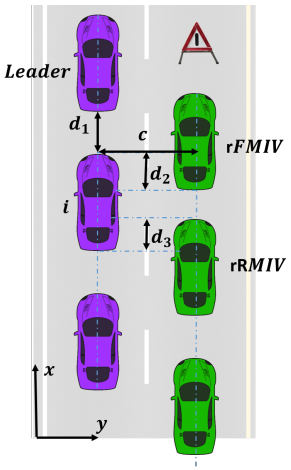


Figure 5.1: Problem definition.

The merging scenario considered is illustrated in Fig. 5.1. Vehicle i in the left platoon is the ego-vehicle. The closest right platoon vehicle in front of the ego-vehicle is referred to as the right-forward most important vehicle (rFMIV) [72]. The vehicle following rFMIV is the right-rear most important vehicle (rRMIV). The ego-vehicle will open up a gap by changing the CACC target from the platoon leader to rFMIV [67] or by adjusting the time-headway if the intention to merge is not clear .

State $X_i(t)$ for the ego-vehicle is defined as:

$$P_i(t) = \begin{bmatrix} d_{i,1} \\ d_{i,2} \\ d_{i,3} \\ c_i \end{bmatrix}, \qquad X_i(t) = \begin{bmatrix} P_i(t) \\ \dot{P}_i(t) \\ \ddot{P}_i(t) \end{bmatrix}, \tag{5.1}$$

where $d_{i,1}$ is the spacing with the preceding vehicle. $d_{i,2}$ is the longitudinal projection of bumper to bumper distance between the ego vehicle and rFMIV. $d_{i,3}$ is the bumper-to-bumper distance between the ego-vehicle and rRMIV. c_i is the lateral distance with the rFMIV.

It is assumed that the exact location of each vehicle is known through its GPS data, and that each vehicle transmits its position, velocity and acceleration to all other vehicles.

Each vehicle effectively interacts with only three other vehicles, namely: the preceding vehicle in the same lane, and the FMIV and RMIV in the other lane. It is further assumed that the road is comprised of only two lanes and only one vehicle merges at a time.

We study merging control design for two different cases, separately. Section 5.2 studies the case where each vehicle is able to communicate its intentions with the relevant vehicles. Section 5.3 investigates the case where vehicles do not communicate their intentions.

5.2 The Case with Communication

A simplified scenario is considered to test the actual merging before implementing the collision check functionality. Since a collision check is not used in this section, a merging flag is utilized. This merging flag F_i is part of the state vector $X_i(t)$.

The state $X_i(t)$ for the ego-vehicle is defined as:

$$X_i(t) = \begin{bmatrix} P_i(t) \\ \dot{P}_i(t) \\ \ddot{P}_i(t) \\ F_i(t) \end{bmatrix}. \tag{5.2}$$

5.2.1 Control Structure

The car model from the acceleration command to the actual acceleration is again defined in the form [3]:

$$\begin{aligned} G_i(s) &= \frac{K_i}{\tau_i s+1}, \\ a_i &= G_i(s)u_i, \end{aligned} \tag{5.3}$$

where τ_i represents the closed-loop bandwidth and K_i is the gain.

The relative distances measured from car i to car j are defined as

$$d_{ij}(t) = x_i(t) - x_j(t), \tag{5.4}$$

where $x_i(t)$ is the longitudinal position of the vehicle. The spacing policy is defined in terms of the desired spacing D_{ij}. A constant time-headway is used throughout this chapter, and the spacing error is defined as:

$$\begin{aligned} \delta_{ij} &= d_{ij} - D_{ij}, \\ \delta_{ij} &= d_{ij} - h_d \dot{x}_i(t). \end{aligned} \tag{5.5}$$

The string-stable CACC design previously described in Section 2.1 is used for the longitudinal controller.

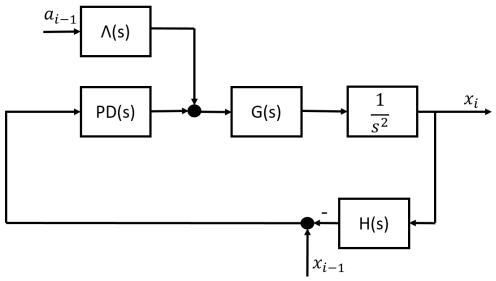


Figure 5.2: Longitudinal control structure.

Fig. 5.2 shows the design of the longitudinal controller. The acceleration of the preceding vehicle is added to the output of the controller after being fed into filter $\Lambda(s)$, which is defined as:

$$\Lambda_i(s) = (H_i(s)G_i(s))^{-1}, \tag{5.6}$$

where $H_i(s)$ is the spacing policy dynamics, assumed to be in the form:

$$H_i(s) = 1 + hs, \tag{5.7}$$

where h is the time-headway used in calculating the desired spacing. For the purpose of the lateral controller, a simple path following PD structure is used.

5.2.2 Decision Chain for the Leader of the Right Platoon

The scenario begins with the leader of the right platoon (as depicted in Fig. 5.1) detecting an obstacle in its own lane, and starts following the leader of the left platoon using virtual CACC (VCACC). After the gap with the leader of the left platoon is sufficiently large, the merging message is sent by setting state F_i to 1. Then the vehicle starts monitoring the gap between itself and the first follower of the left platoon, represented by $d_{i,3}$ in state $P_i(t)$. When the gap, $d_{i,3}$ is sufficiently large, the vehicle starts the lane change maneuver.

5.2.3 Decision Chain for the First Follower of the Left Platoon

Vehicles in the left platoon are responsible for listening to merging requests, F_i, from the right lane. When a merging request is received from rFMIV, it implies that the vehicle has already opened up a gap with the leader of the left platoon.

After receiving the merging request message, the ego-vehicle changes the target of its CACC algorithm to rFMIV by constructing a virtual copy of that vehicle in its own lane [67]. A change of target cannot be done instantaneously, and so in order to get smooth control outputs, a gradual target change is proposed. Fig. 5.3 shows the gradual change of the target over a period of two seconds. The duration of the gradual change can be adjusted to the needs of the specific scenario.

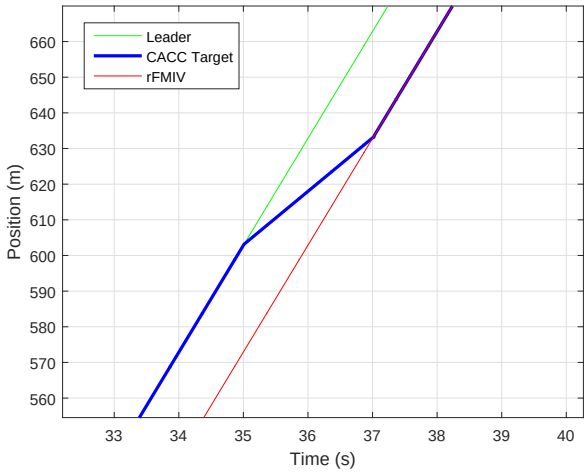


Figure 5.3: Change of moving target.

5.3 The Case without Communication

This section addresses a similar problem described in the previous section, but here vehicles do not explicitly communicate their intentions. For this scenario we propose a collision check technique, which is used in situations where cars are not explicitly communicating their intentions to merge but instead slowly start drifting into the lane they want to merge with.

5.3.1 Collision Check

The ego-vehicle constantly maintains a check for the possibility of a collision with neighboring vehicles from an adjacent lane. Two indices are calculated to decide what action the ego-vehicle will take. The first index is the time-to-collision index (t_{col}), viz., the time for a possible collision with a vehicle in the adjacent lane, which is calculated dynamically as the solution to the equation:

$$t_{col} = |\frac{c}{\dot{c}}|. \tag{5.8}$$

The second index is called the ego-vehicle-collision check (γ_c) and checks the relative

speed of the vehicles to determine if a potential collision can be avoided by the ego-vehicle opening up a gap. γ_c has a value between zero and one, and is calculated as follows:

if center of rFMIV will be in zone S after t_{col} seconds

$$\gamma_c = max(0, min(1, t_{col}^{-1}))$$

else

$$\gamma_c = 0$$

where zone S denotes the region between the rear bumper of the ego-vehicle to the front bumper of its leader. Fig. 5.4 shows this region.

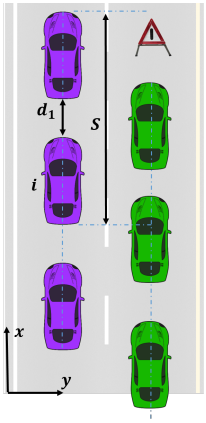


Figure 5.4: Merging region for the ego-vehicle.

The Fig. 5.5 shows the state transition for a left lane vehicle:

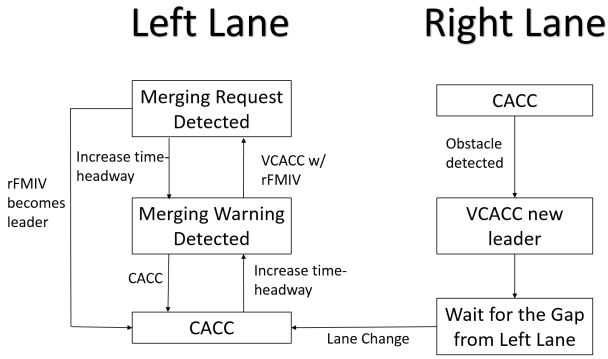


Figure 5.5: State transition for left lane vehicle.

Transitions between states are done by checking the values of c and γ_c.

if $c \geq c_{threshold}$ & $\gamma_c \geq 0$
    Merging Warning Detected:
    Change the time-headway $h_d = 1 + \bar{h}_d * \gamma_c$

else if $c \leq c_{threshold}$
    Merging Request Detected:
    Start VCACC with rFMIV

else
    CACC

where $c_{threshold}$ is the critical point at which rFMIV starts getting too close to the ego-vehicle and $\bar{h}_d$ is the constant that defines the weight of γ_c on the time-headway.

5.4 Simulations

Simulations are done with four vehicles in the left lane and one in the right lane. During these simulations all vehicles are taken as point particles and are initially spread out with one meter between each other in both lanes. Since the leader of the right platoon has the intention of merging behind the leader of the left platoon, initially the leader of this right platoon is located on same longitudinal position as the first follower of the left platoon. The leader of the left platoon is controlled manually (as in, it has a predefined velocity profile). Only the leader of the right platoon and the first follower of the left platoon are equipped with merging controllers.

All vehicles are assumed to be homogeneous. GPS positions correspond to the center of each vehicle and are assumed to be known. The desired spacing between vehicles is defined with a one second time-headway.

5.4.1 The Case with Communication

These results simulate an environment where the vehicles have the proper channels to communicate their merging requests.

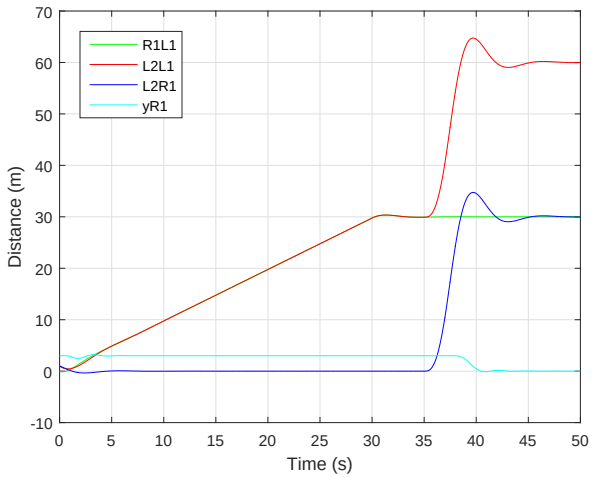


Figure 5.6: Relative spacing between vehicles.

Fig. 5.6 shows the spacings between the leader, rFMIV ($R1L2$) and first follower ($L1L2$), first follower and rFMIV ($L2R1$) and the lateral position of rFMIV ($yR1$). At t=35s rFMIV sends the merging flag and the ego-vehicle gradually changes its target to rFMIV to open up a safe gap. When there is enough gap between the leader, rFMIV and ego-vehicle, rFMIV starts the lateral maneuver. It takes around 3.5 seconds for the ego-vehicle to open up the gap and 1.5 seconds for rFMIV to do the actual merging maneuver. Fig. 5.7 shows a close-up of this fragment.

Fig. 5.8 shows the velocity profile of all the vehicles in this scenario. First follower starts slowing down to open up a gap with respect to the merging vehicle around t=35s. Looking at the velocity profile of the second and third followers vehicle at this moment, it can be concluded that the platoon maintains string stability, due to the fact that their responses do not exceed the one of the leader.

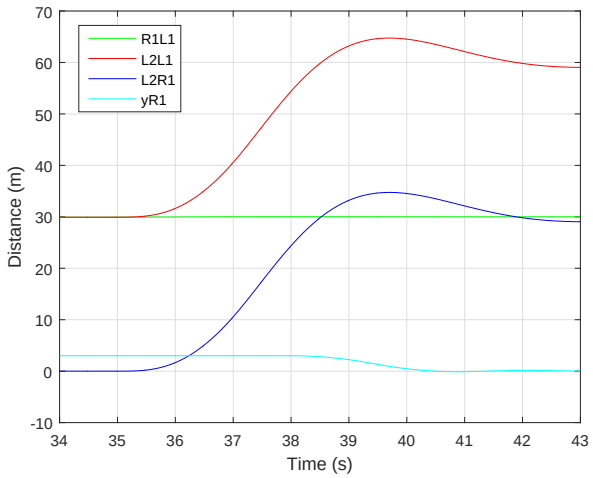


Figure 5.7: Relative spacing between vehicles during merging.

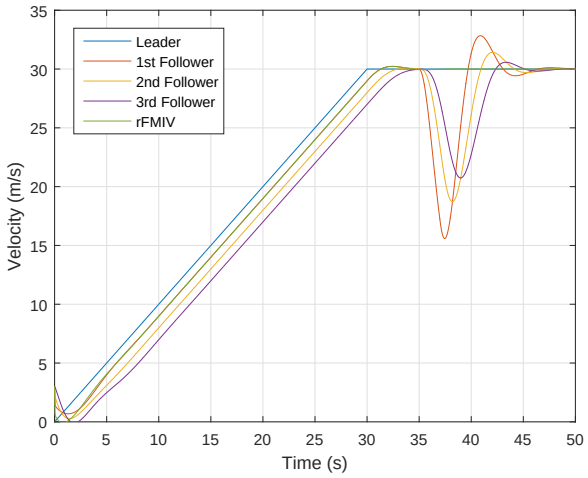


Figure 5.8: Vehicle speeds.

5.4.2 The Case without Communication

Decision chains and control structures in this scenario are similar to the previous one, except the merging flag is not used. Instead, at t=35s the vehicle in the right platoon starts getting closer to the ego-vehicle by a small amount to initiate the gap opening.

rFMIV monitors the gap between both the leader and the ego-vehicle to make sure not to cause a collision. When there is enough gap between both vehicles, rFMIV completes merging with the lane. While rFMIV gets close to the left lane, the ego-vehicle uses collision check functionality to navigate between states.

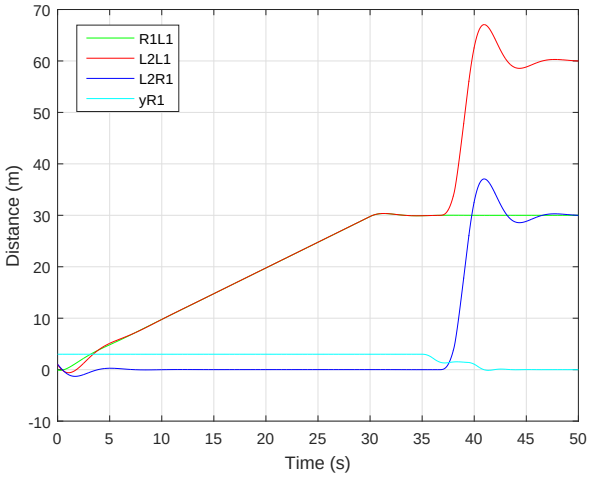


Figure 5.9: Relative spacing between vehicles.

Fig. 5.9 shows the spacings between the leader, rFMIV and ego-vehicle along with the lateral position of rFMIV. At t=35s rFMIV starts getting close to the ego-vehicle to prepare for a merging. After enough gap is opened between vehicles, rFMIV finishes the merging. It takes 6 seconds in total for the rFMIV to merge with the left platoon, as opposed to 5 seconds in the case where merging flag is used. Fig. 5.10 shows a close-up of this fragment.

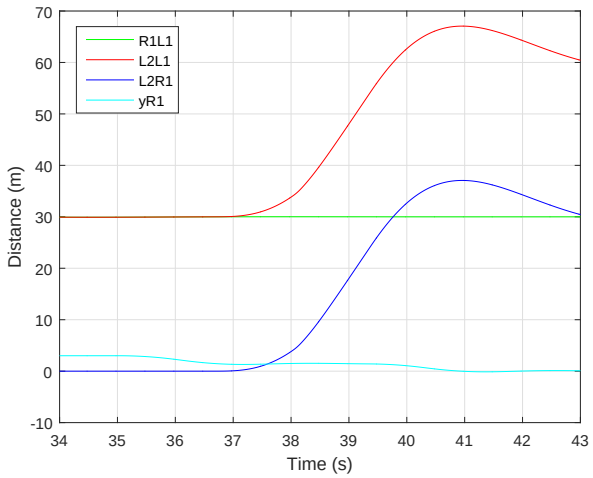


Figure 5.10: Relative spacing between vehicles during merging.

Fig. 5.11 shows the velocity profile for all the vehicles in this scenario. String-stability is still maintained in this scenario, and minimum speeds of the vehicles are similar to the first scenario.

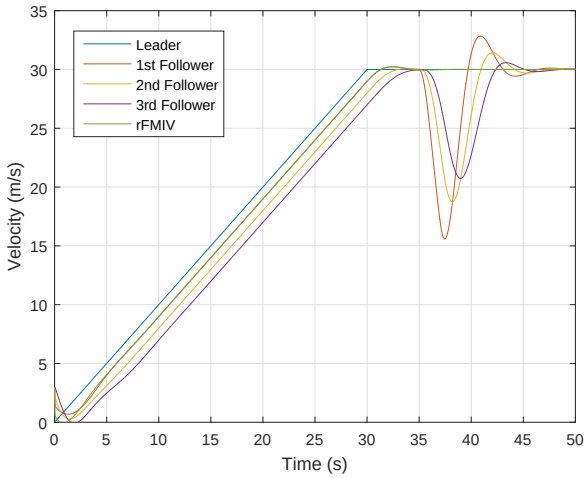


Figure 5.11: Vehicle speeds.

5.5 Summary

In this chapter, a generalized merging detection protocol where the merging vehicle does not send any merging requests to the main lane vehicles has been presented. Simulation tests were successful and string-stability was maintained for the whole simulation time for each case.

The structure presented in this chapter is designed for environments where both fully and semi-autonomous vehicles exist together. It is mainly designed to avoid collisions and dangerous situations rather than increase traffic throughput and negative effects of merging. This work will be used as a basis of the next chapter where a multi vehicle merging scheme is proposed for fully-autonomous vehicles.

Chapter 6

Graph Theory Based Merging Scheme

Chapter 5 explored the problem where a vehicle decides to change lane with and without the use of inter-vehicle communication. The problem definition presented in Chapter 5 only considers merging of a single vehicle. Although Chapter 5 is used as the basis of the work introduced in this chapter, the solution of a single vehicle merge cannot directly be applied to platoon merging.

This chapter studies a more general form of the lane merging problem where only minimal communication is required between neighboring vehicles. No roadside unit (RSU) will be used. The platoon leaders have no specific functions in this merging algorithm - all the consensus is done regardless of the class of the vehicle, i.e. follower or platoon leader.

A detailed review of published work on platoon merging is explained in detail in Section 2.5.3. As was noted there, the focus of most research in the literature is either modeling or control of highway on-ramp merging. Ploeg et al. recently outlined a merging protocol for lane closures, where the protocol suggested by the organizers was implemented as a part of GCDC 2016 by more than ten teams using different vehicles. At the end of Section 2.5.3, this protocol was explained in detail along with its shortcomings.

One of the biggest issues with platoon merging is the possibility of main lane traffic coming to a halt if the entire platoon attempts merging simultaneously. In the protocol presented in this chapter, a limit is placed on the number of cars that can merge simultaneously, and the effect of the value of this limit will be analyzed in the simulations section. This limit is represented with the parameter N. This parameter is set prior to the implementation, so it is a design constant rather than a value set by the leader. Pairing,

gap opening and merging is only done between the first N vehicles, and as the merging is completed this window of actively merging vehicles propagates to the back of the platoon. This allows only the first N vehicles to pair up and the others to wait, since preemptive pairing might result in rogue pairs.

A merging scheme is formulated where the vehicle on the main lane detects the intentions of neighboring vehicles based on their state parameters. This work will focus on join maneuver detection and gap opening, excluding the details of the lateral controller design for full implementation of the maneuver.

6.1 Problem Definition

The scenario investigated in this chapter consists of a two-lane highway, where one of the lanes is closing further down the road due to an accident or roadwork. The leader of the gap opening platoon, or the target lane, is denoted by i and the leader of the merging platoon (as in, the vehicle closest to the lane closure point), is identified by $i-1$. Fig. 6.1 shows a representation of a possible vehicle configuration near the lane closure.

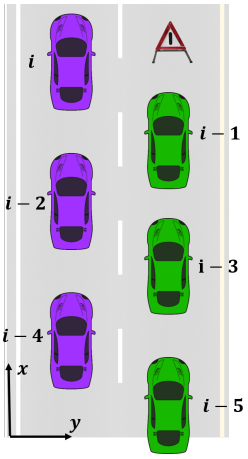


Figure 6.1: Vehicle configuration.

It is assumed that all vehicles within a certain vicinity of the lane closure adopt the same speed to make merging possible. A merging protocol should be available for all vehicles, such that a zipper merge can be implemented by the vehicles using only external sensors and communication data. This means that vehicle $i-1$ merges between vehicle

i and $i-2$, vehicle $i-3$ merges between vehicles $i-2$ and $i-4$, etc. Every vehicle is responsible for pairing with the correct vehicle, where each pair is defined as the vehicle for whom a gap will be opened to, i.e. vehicle $i-1$ should pair up with vehicle i, then open up the respective gap required for the merging to be actualized. This merging protocol should be paired with a set of information to be communicated between vehicles such that it should be able to reach a consensus for all probable cases.

A common problem faced during a multi-vehicle merging scenario, as described above, is vehicles towards the end (or tail) of the platoon slowing down too much, even stopping in certain cases. In (present day) real world traffic, where drivers do not have the opportunity to communicate a merging scheme, merging is done on a one-by-one basis. While this may not lead to the tail vehicle slowing down to a standstill, it will take a long time to complete the merge.

In this chapter, a complete longitudinal merging scheme for a versatile platoon merging scenario is proposed. This structure comprises of the design of the interaction protocol, longitudinal control agents and integration of these components.

The merging scheme presented in this chapter is designed in a way to address certain issues missed by other studies in this field. The proposed communication structure and message set aims to keep the number of messages sent between vehicles to a bare minimum, while assuring that it is enough to reach a consensus with this set of messages. Another issue that is addressed in this chapter is the use of a dedicated platoon leader to initiate and assign roles to other participants; the protocol should be completely decentralized. Another goal we had for his work is that it should be scalable, so that the same protocol could be used for various merging scenarios or for multi-lane roads. Finally, a desirable feature would be that no vehicles in either platoon should come to a complete stop due to the merging protocol.

Performance matrices used to evaluate the effectiveness of the controller are:

1. Speed of the tail vehicle,
2. Time (or distance) required to complete the merge.

6.2 Communication Structure

Marjovi [80] introduced the concept of using a dynamic graph based approach to control multi-lane vehicular convoys. Similar to [80], a dynamic graph is used in this chapter to represent interactions of the ego-vehicle with its neighboring vehicles.

Each vehicle communicates with a set of vehicles that is within its wireless communication range, but only a small subset of these vehicles are relevant to the control and decision making strategy of that particular vehicle. The group of vehicles whose information is pertinent to an ego vehicle is represented by a local dynamic graph specific to that ego-vehicle.

This graph is denoted with $G_r = (V, E)$, where V represents the vertices, i.e. vehicles that have communication links with the ego-vehicle (including the ego-vehicle itself) and the edge set E represents the wireless communication links between the vehicles. The graph G_r is undirected, that is, communication links between vehicles are bidirectional.

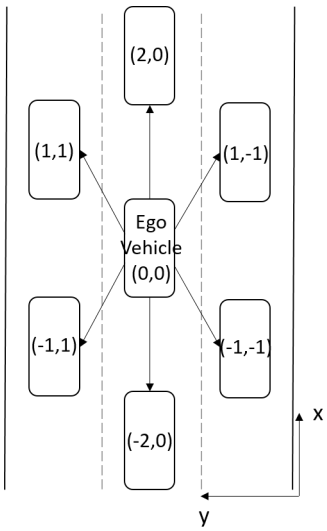


Figure 6.2: Connectivity map for ego vehicle.

For each vehicle, this local graph includes only the vehicles that are in the immediate neighborhood of the vehicle. The graph for a vehicle in the middle lane of a three lane road is shown in Fig. 6.2. Each vehicle within its immediate neighborhood is assigned a local set of coordinates, which will be the local-identifier for that certain vehicle. These local identifiers are shown in Fig. 6.2. A local coordinate system is defined, where the ego vehicle is the origin, and vehicles move on the positive x direction. In this local coordinate system, the ego-vehicle is at coordinate $(0, 0)$. The preceding vehicle is at coordinate $(2, 0)$, and a vehicle driving in the lane to the left and immediately ahead of the ego-vehicle is represented at coordinate $(1, 1)$. These connectivity graphs are created locally by each vehicle, and are not shared through communication. These graphs are dynamic and updated with each time step.

Each vehicle is not expected to be fully-connected at each instance. An isolated incidence matrix is used to illustrate existing connections. The isolated incidence matrix for a fully connected ego-vehicle is:

$$I_{(0,0)} = [111111]. \tag{6.1}$$

Universal ID's of vehicles existing in the vertexes V of the graph G_r are stored in the connectivity vector CV. The connectivity vector for the fully connected ego-vehicle is:

$$CV_{(0,0)} = \begin{bmatrix} ID_{(2,0)} \\ ID_{(1,1)} \\ ID_{(-1,1)} \\ ID_{(-2,0)} \\ ID_{(-1,-1)} \\ ID_{(1,-1)} \end{bmatrix}. \tag{6.2}$$

Each vehicle will wirelessly communicate certain states to all its neighbors. The state vector is defined in a way that includes all the necessary information to perform a successful platoon merge, but kept as compact as possible to make communication faster and more effective. The state vector is formed as follows:

$$X_i = \begin{bmatrix} ID_i \\ x_i \\ y_i \\ Heading \\ v_i \\ a_i \\ MergeRight_i \\ MergeLeft_i \\ RoadClosureCounter_i \\ MIO_i \end{bmatrix}. \tag{6.3}$$

where $MergeRight$ and $MergeLeft$ are two flags used to notify the surrounding vehicles of the ego-vehicle's intentions. $RoadClosureCounter$ is used to communicate whether the merge is an isolated incident or is a lane closing. For an isolated merge this field is set to 1 or to a positive number N for a lane closure. The value of N will be discussed in subsequent sections. If there is a vehicle in front of the ego-vehicle in an idle state, then the $RoadClosureCounter$ is inherited from the preceding vehicle and lowered by one, while

never going below 0. MIO is an acronym for the Most Important Object, which is the object that the ego-vehicle is temporarily following in the *target change* state.

6.3 Hierarchical Architecture

This section gives an overview of the components used to implement/simulate this merging structure. Each vehicle contains three main structures:

1. Vehicle,
2. Communication Structure,
3. Control structure.

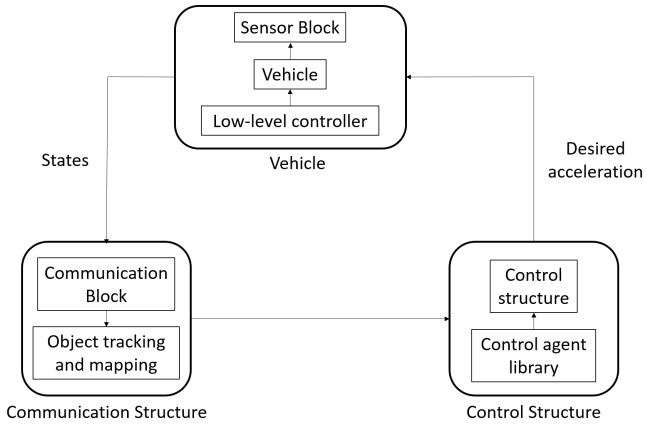


Figure 6.3: Hierarchical architecture.

The Vehicle gets the desired acceleration from the control structure, which is fed into the low-level throttle-brake controller. The vehicle simulator block sends an updated state matrix to the communication block.

The Communication Structure gets the states of the ego-vehicle and the communication information from the surrounding vehicles, and creates the connectivity vector CV.

The Control Agent Library gets the connectivity vector, states of the vehicle and all the states of vehicles connected to the ego-vehicle, and activates the appropriate control structure.

Each vehicle is assigned a local identifier as shown in Fig.6.2. Local identifiers are assigned as in [80] and paired with the actual vehicle ID's. Usage of local identifiers ensures that the control agent library used by each agent will be consistent. The weighting matrix and the local identifier information is used only within the vehicle and not shared via V2V communication.

Since we are only focusing on the longitudinal aspects of platoon merging, only the longitudinal agents will be discussed. These are:

1. Cruise Control,
2. Unidirectional-CACC,
3. Bidirectional-CACC,
4. Virtual-CACC,
5. Blind-Spot Avoidance.

In the control structure section, the functionalities of these agents will be explained.

6.4 Merging Algorithm for Platoon Merging

The merging algorithm for merging and gap opening vehicles will be explained in this section. Vehicles will be further sub-categorized depending on whether they are the platoon leader or not. Approaching the merging zone all vehicles set their cruise speed to a previously defined value. This phase ensures that a pairing between vehicles will not lead to rogue paring. This phase is activated when a vehicle gets within a certain distance of the lane closure.

6.4.1 Merging vehicle

If there are any lane closures or accidents that can affect other vehicles in that lane, the leader is responsible for alerting other vehicles in its own lane. It is not efficient for all

vehicles to start changing lane at the same time, since this will make the traffic come to a halt if the total number of merging vehicles is high. Instead the leader will alert the N closest vehicles in its platoon. This is done by setting the *RoadClosureCounter* to N. If this counter is N for the ego-vehicle, then vehicle $(-2, 0)$ will set its*RoadClosureCounter* to $N-1$. This propagation continues until the counter hits zero, and a new virtual platoon leader is created within the platoon. Once vehicle $(-2, 0)$ becomes the new leader of the platoon and detects the road closure it will update its counter as N. In this way all the vehicles in the platoon will have a positive number as their *RoadClosureCounter* at one point.

Having a positive number for *RoadClosureCounter* indicates that the vehicle in question has the intention to make a lane change. The default value for this state is 0, which indicates that the vehicle has no intention to change lane at that moment. While this counter indicates the intention to merge, the direction of the merge is indicated with $MergeRight$ and $MergeLeft$ flags.

Once the intention flags are set, the merging vehicle will choose the most important object (MIO) for the merging action. Every time *RoadClosureCounter* is larger than zero and MIO is set to zero, the ego-vehicle will try to find a MIO. There are three cases to be considered:

1. if (1,-1) does not exist for the ego-vehicle, then MIO will be kept as zero,
2. if $x_{(1,-1)}$ precedes $x_{(2,0)}$, then (1,-1) will be the MIO if and only if vehicle (2,0) does not have this vehicle as its MIO,
3. if the X position of (1,-1) is between the ego-vehicle and (2,0), then (1,-1) is assigned as the MIO.

After the MIO assignment is complete, the next stage is opening the required gaps.

1. If MIO is selected, then an appropriate gap will be opened up with respect to the MIO. Once the required gaps are opened up, the vehicle goes into the merging state,
2. If the *RoadClosureCounter* is N while MIO is still not set, then vehicle (-1,-1) should start opening up the required gap for the ego-vehicle. Once the ego-vehicle detects that the gap is large enough, merging occurs.

MIO and *RoadClosureCounter* should be reset once the merge is completed. The state transition for the merging vehicle is shown in Fig.6.4.

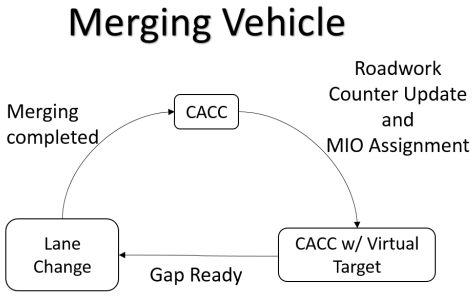


Figure 6.4: State transition for merging vehicle.

6.4.2 Gap Opening vehicle

Vehicles constantly monitor the *RoadClosureCounter* of all the surrounding vehicles. If at any point in time one of them has a positive value, the gap opening logic is activated. Priority is given to the vehicle with $RoadClosureCounter = N$, since that vehicle is closest to the road closure.

If one of the neighboring vehicles with $RoadClosureCounter = N$;

1. has an MIO and this MIO is in the (2,0) position, then a gap will be opened with respect to the vehicle with $RoadClosureCounter = N$ and it will be set as the MIO,
2. has no MIO selected and is in the (1,-1) or (1,1) position, then a gap will be opened with respect to this vehicle and set as the MIO.

For all other neighboring vehicles with a positive *RoadClosureCounter*, the ego-vehicle will check if any of the surrounding vehicles' MIO is in the (2,0) position. If it is, then that vehicle will be set as the MIO and a gap will be opened.

Once the merging of the MIO is completed, the MIO state will be reset and the vehicle will go back into regular CACC mode.

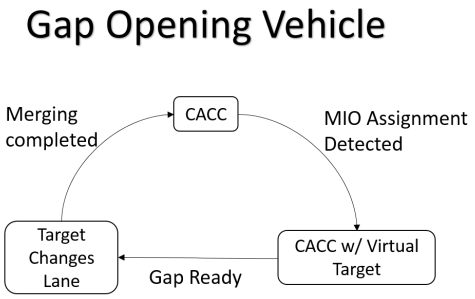


Figure 6.5: State transition for gap opening vehicle.

6.5 Vehicle Model and Control Structures

6.5.1 Vehicle Dynamics

Due to the expansive number of vehicles required to simulate the merging scenario, a simple yet effective vehicle model as described in Section 2.1, is used. The acceleration-based plan model is given as follows:

$$G_i(s) = \frac{K_i}{\tau_i s + 1}, \tag{6.4}$$

where τ_i and K_i are parameters dependent on vehicle. Input to the plant is an acceleration command u and the output is actual acceleration a.

The distance between the ego-vehicle in question and it's leader is given by:

$$d_{i,i-1} = x_i - x_{i-1}. \tag{6.5}$$

The distance between the ego-vehicle in question and it's follower is given by:

$$d_{i,i+1} = x_i - x_{i+1}. \tag{6.6}$$

The control inputs for the controllers used in this section are deviations from the desired positions. Here the desired relative distance is represented by D_i, which is defined as:

$$\begin{aligned} \delta_{i,i-1} &= d_{i,i-1} + D_i, \\ \delta_{i,i+1} &= d_{i,i+1} - D_i. \end{aligned} \tag{6.7}$$

A constant time headway spacing policy is used in our model. The formula for D_i is defined as:

$$D_i = D_0 + h_i v_i, \tag{6.8}$$

where h_i is referred to as the desired headway time. D_0, the constant spacing, is assumed to be zero in this chapter.

6.5.2 Cruise Control

The control structure is used by the platoon leader in the case where no MIO is present. For this purpose a simple PD-based control structure, as shown in Fig. 6.6, is used.

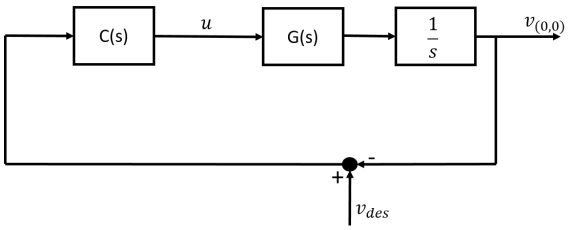


Figure 6.6: Control structure for cruise control function.

The controller, $C(s)$ is defined as:

$$C(s) = k_p + k_d s. \tag{6.9}$$

6.5.3 Bidirectional-CACC

During platooning with vehicles within the ego-vehicle's own lane, this bidirectional CACC structure is employed. The structure explained here was previously introduced in Section 4.4.2. The organization of this control structure is shown in Fig. 6.7.

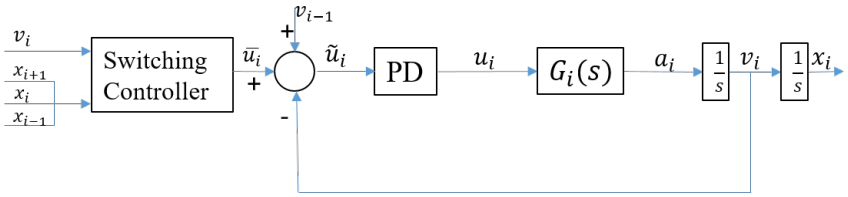


Figure 6.7: Controller design for acceleration controlled vehicles with variable spacing.

The switching control functions output takes the following form:

$$\bar{u}_i = -\sigma(\delta_{i,i-1}) - 0.3\sigma(\delta_{i,i+1}), \tag{6.10}$$

The switching function $\hat{\sigma}$ is defined, as in Section 4.4.2:

$$\sigma(\delta) = \begin{cases} 0 & if \quad |\delta| \leq \bar{\eta_m} \\ \bar{k}\frac{\delta - \bar{\eta_m}}{\bar{\delta}}\delta & if \quad \bar{\eta_m} < |\delta| \leq \bar{\eta_m} + \bar{\delta} \\ \bar{k}\delta & if \quad |\delta| > \bar{\eta_m} + \bar{\delta} \end{cases} \tag{6.11}$$

where $\bar{\eta_m}$ represents the upper limit on the sensor noise and $\bar{\delta}$ is a design constant.

This output is combined with the feed-forward component and the negative of the current velocity to create the velocity error.

$$\tilde{u}_i = \bar{u} + v_{i-1}. \tag{6.12}$$

The deviation from the controlled velocity is fed into the PD controller to get the acceleration input.

6.5.4 Unidirectional-CACC

Vehicles immediately behind or in front of a vehicle taking active part in a merging switch to unidirectional CACC. The Control structure used for this purpose uses the same architecture as that previously presented in Section 6.5.3.

While the same switching function is used for the unidirectional CACC, the error with the following vehicle is not included in $\bar{u}_i$. The control output before the feed-forward component ($\bar{u}_i$) is defined as:

$$\bar{u}_i = -\hat{\sigma}(\delta_{i,i-1}). \tag{6.13}$$

6.5.5 Virtual-CACC

Virtual CACC is activated when a vehicle, that is not the platoon leader, needs to open up a gap with another vehicle from the neighboring lane while making sure that minimum spacing rules with its own predecessor are not violated. Virtual-CACC refers to the combination of two separate unidirectional-CACC with a fuzzy switching function. The virtual-CACC structure is shown in Fig. 6.8

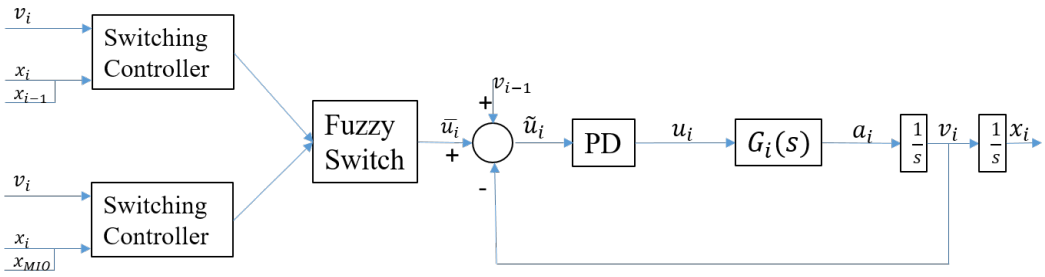


Figure 6.8: Virtual-CACC.

The control structure described in Section 6.5.4 is run on both the MIO and preceding vehicle $(2, 0)$. The control outputs coming from two structures are added after going through the fuzzy switching structure. The switching function is based on δ_t:

$$\delta_t = d_f - d_{MIO}. \tag{6.14}$$

The switching function is shown in the figure below.

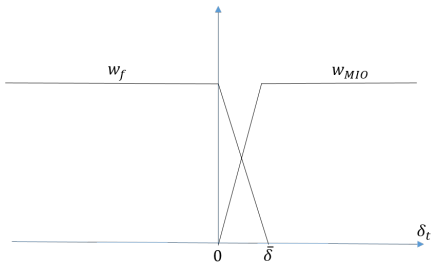


Figure 6.9: Switching function for virtual-CACC.

The purpose of the fuzzy switching function shown in Fig. 6.9 is to have a smooth transition between the vehicles two targets. The switching does not center around $\delta_t = 0$,

instead the ego-vehicle starts switching to its leader when $\delta_t = \hat{\delta}$. This earlier gradual switching is used to assure that at $\delta_t = 0$, ego-vehicle will be only following its leader.

6.5.6 Blind-Spot Avoidance

A human driver tends to avoid staying in neighboring vehicles' blind spots; different works in this area tried to mimic this behavior by forcing vehicles to stay in a slot-based chessboard formation [15, 61]. In this chapter we will use a similar action so that vehicles in neighboring lanes are never perfectly in line. This will ensure that pairing for lane change will be unique at each time. This controller will be active at all times, not just when the vehicle lane change mode. The approach proposed in this section is less aggressive compared to the ones presented in [15, 61] in order to make the implementation feasible in environments where vehicles of different sizes coexist.

For the purpose of blind-spot avoidance, a potential function similar to the one presented in [81] is used. The potential function in [81] is used instead for gap opening to the target vehicle, whereas in this thesis a similar function is proposed for the steady state use to eliminate vehicles cruising in each others blind spots.

Each vehicle only considers it's two closest neighbors in the left lane. Vehicles in the left-most lane do not activate this controller.

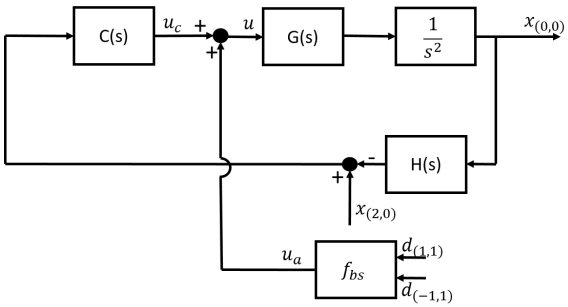


Figure 6.10: Block diagram for blind spot avoidance with basic CACC.

The input to the blind-spot avoidance function is the spacing with the two closest vehicles in the left lane. The function is defined as following:

$$f_{bs}(t) = -be^{-d^2_{(-1,1)}/(2a^2)} - be^{-d^2_{(1,1)}/(2a^2)}, \tag{6.15}$$

where a and b are design constants. The blind-spot avoidance function presented in this section is not a stand alone structure, instead it is used in addition to other platooning structures. The blind-spot avoidance function is only activated in platooning mode and automatically deactivated if any of the surrounding vehicles are playing an active role in merging.

6.6 Simulations

Two sets of simulations were performed to show how the merging scheme and the control structures presented in this chapter work. PreScan was used to perform all the simulations presented in this chapter. The details of each simulation are explained in the relevant section.

6.6.1 Effects of Blind-Spot Avoidance on String Stability

The blind-spot avoidance proposed in this chapter plays an important role in the effectiveness of the proposed merging protocol. In order to test the effects of this blind-spot avoidance function on the string-stability of the platoon, a simulation scenario is created. A platoon of six vehicles, of which the leader' velocity profile is predefined, is placed in the left lane and a single vehicle on the right lane. All left lane follower vehicles are equipped with the bidirectional CACC. The right lane vehicle is controlled with an erratic velocity profile that makes this vehicle drift around the first three followers of the left platoon. Fig. 6.11 shows the x-positions of the followers of the left platoon and the right vehicle.

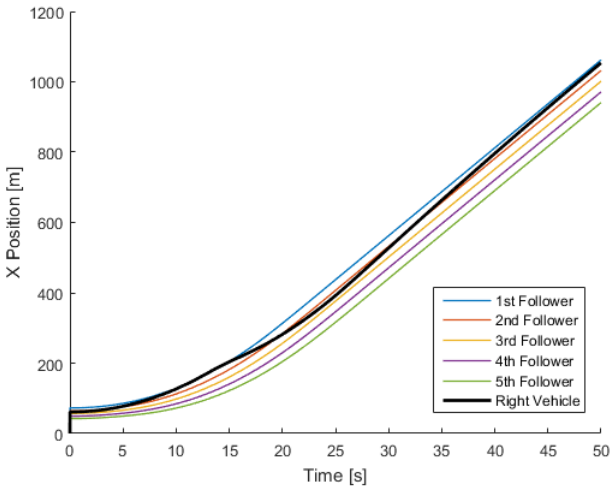


Figure 6.11: X-positions of all vehicles.

Fig. 6.12 shows the acceleration profiles of the followers. It can be seen that the string stability of the platoon is maintained at all times.

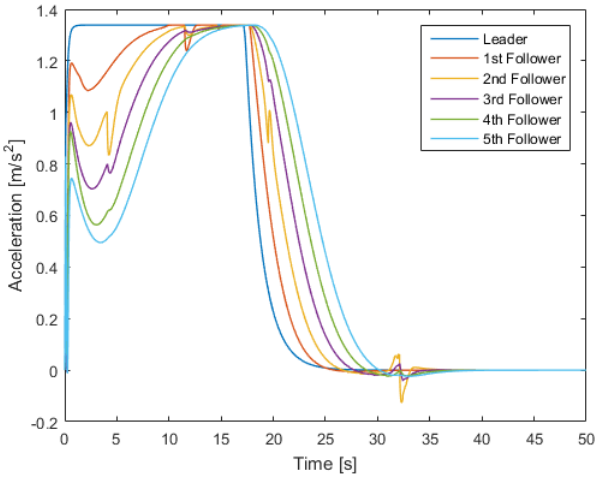


Figure 6.12: Acceleration plots of left-lane vehicles.

Where effects of the blind-spot avoidance function are present, it can be seen that the spikes caused by this function are attenuated as they propagate upstream.

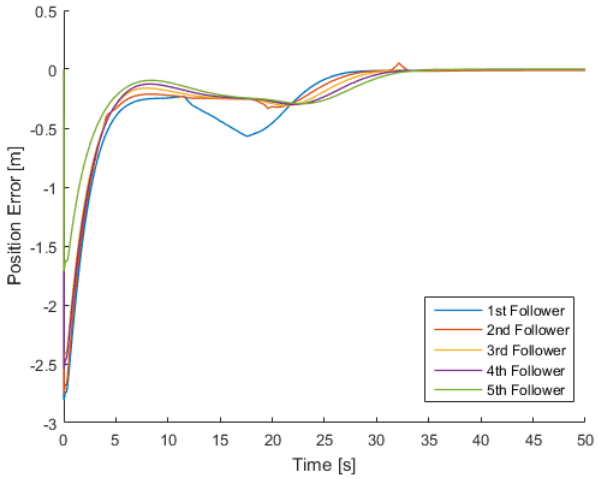


Figure 6.13: Deviation from desired spacing for left lane follower.

Fig. 6.13 demonstrates how the blind-spot avoidance function impacts deviation from desired spacing. The blind-spot avoidance function causes the vehicles in the left platoon to deviate at most 0.5 meters from their desired spacing, which is quite insignificant compared to the value of the desired spacing. Also spacing errors are not propagated upstream since string stability is not affected by the addition of this function.

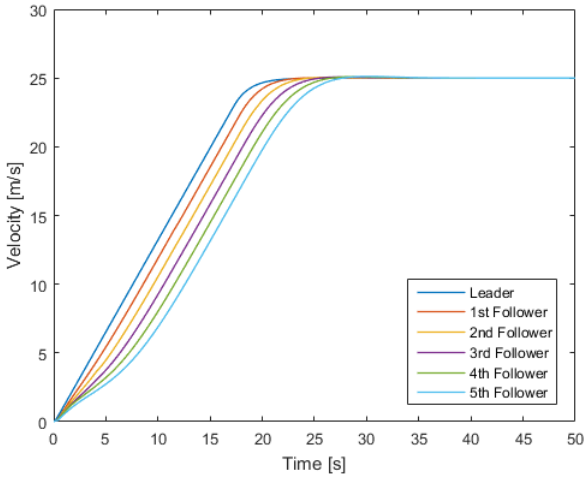


Figure 6.14: Velocity plots of left-lane vehicles.

6.6.2 Effects of Number of Simultaneously Merging Vehicles

A two lane straight road with eighteen vehicles was used to simulate the merging scheme proposed. Ten of the eighteen vehicles were on the gap opening lane where the rest was on the merging lane. Simulation was designed to start with two platoons moving with equal speed to avoid including the speed matching phase. All the vehicles used in the simulation were equipped with the same control structure, vehicle dynamics and decision making scheme.

No road side unit is used in the simulation other than the signal emitted by the lane closure sign. This signal only includes message type and position of the relevant activity.

The proposed merging scheme is tested with different values of N to see the effect of N on the merging. Figures 6.15 and 6.16 show the time it takes to complete and the slowest speed of the tail vehicle versus N.

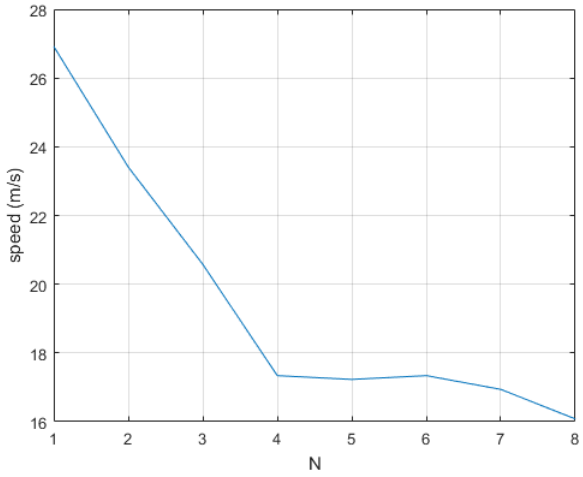


Figure 6.15: Effect of N on platoon speed.

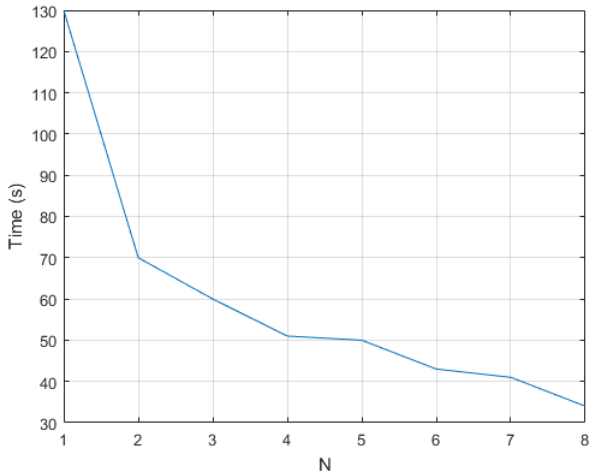


Figure 6.16: Effect of N on total merging time.

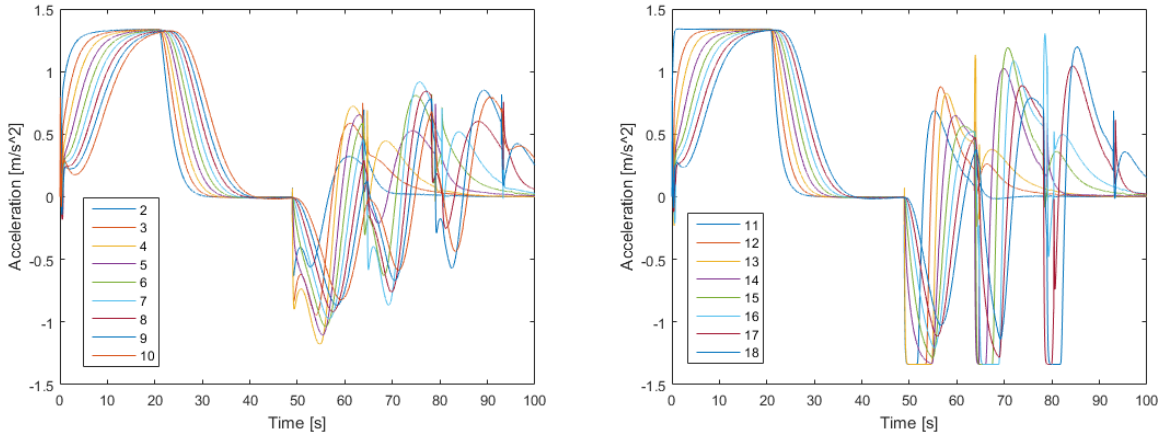
Based on Fig. 6.15 it can be seen that speed of the tail vehicle decreases rapidly until $N = 4$, after which the change in speed is significantly reduced. Similarly, looking at 6.16 it can be seen that time required to complete merging decreases significantly, after which change starts to become insignificant.

It can be seen from this simulation that the time it takes to complete the merging depends on the percentage of the platoon simultaneously merging. For this simulation $N = 4$ represents half of the platoon merging simultaneously. When half the platoon tries to merge simultaneously speed of the tail vehicle drops from 25m/s to 17/ms. Where as if only one third of the platoon merges simultaneously, case with $N = 3$, then the minimum speed increases from 17m/s to 20 m/s while the total time required for merging only increases from 50 seconds to 60 seconds.

6.6.3 Simulation Results for $N = 3$

This section presents the results of a eight vehicle platoon merging with a ten vehicle platoon. The merge is simulated with $N = 3$, which means that only three vehicles can actively merge at a time. Vehicle IDs 1-to-10 belong to the gap opening lane, where the ones with the ID 11-to-18 belong to the merging lane.

Simulations show the vehicles start accelerating from their initial stop positions until they reach their desired cruising speed. Once vehicles reach the vicinity of the merging point, merging is initiated.



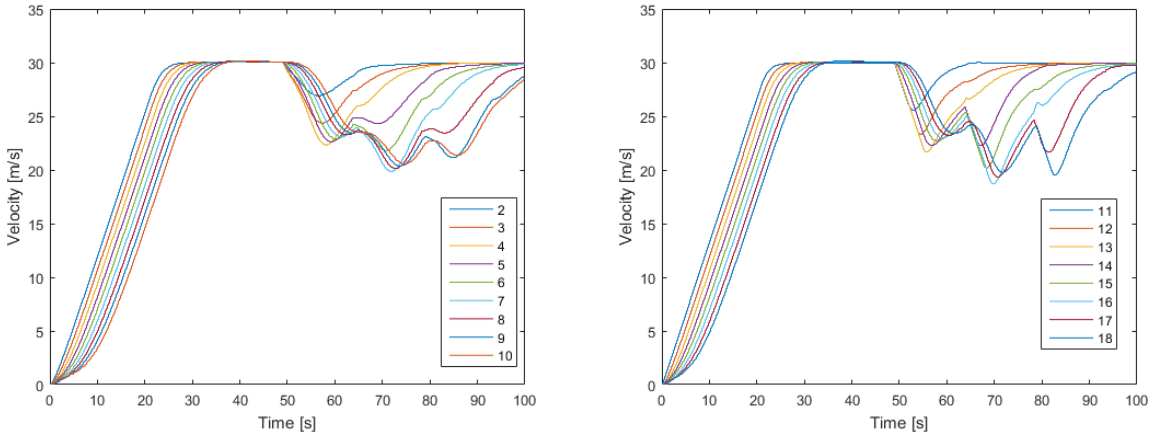
(a) Gap opening lane.

(b) Merging lane.

Figure 6.17: Acceleration plots of all vehicles.

Fig. 6.17 shows the acceleration plots of both lanes. Acceleration plots are presented to demonstrate the string-stability of the formation during the merge. Investigating the string stability during the merge, only the following vehicles should be taken into consideration, e.g. if vehicles 2-to-4 are opening up gaps, then acceleration profiles of vehicles 5-to-10 should be considered when looking at the string stability. Between t=50s to t=65s, gap opening of the first three vehicles can be visualized. Looking at the acceleration profiles of their followers, it can be concluded that the platoon maintains string stability.

A similar check can be applied to the second plot in Fig. 6.17, where merging vehicles open up their respective gaps between t=50s to t=65s. During this time interval, it can be seen that vehicles with ID's 11-to-13 are opening up gaps. Looking at the acceleration profiles of their followers, vehicles with IDs 14-18, maintain string stability.



(a) Gap opening lane.

(b) Merging lane.

Figure 6.18: Velocity plots of all vehicles.

Velocity profiles of the merging vehicles are shown in Fig. 6.18. Plots presented in this figure show the three-by-three merge of the vehicles. Instead of vehicles slowing down all at once to create all 8 of the gaps, gaps are opened three-by-three, which in turn allows the vehicles who complete their merge first to accelerate to their original speed immediately.

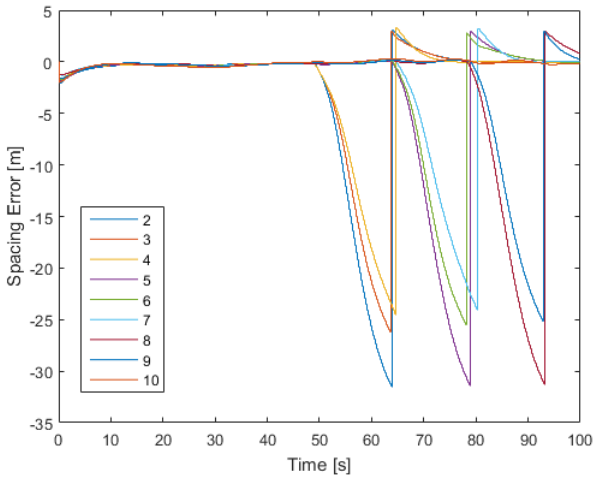


Figure 6.19: Distance errors for the gap opening lane.

Fig. 6.19 visualized the spacing errors for the gap opening vehicles. The spacing errors

are given with respect to the leaders of each vehicle. While opening up the gap required for merging, vehicles follow their virtual targets, which result in the spacing error plot with their respective leader larger. This plot shows vehicles recognizing their new leaders, once the merging vehicles complete their merge. This change of leader can be seen around t=67s, where vehicles with IDs 2-to-4 choose the newly merged vehicles as their leaders.

6.7 Summary

This chapter has integrates all the concept used throughout the thesis to create a unified merging scheme. The chapter started with the proposal of a new method to sort the incoming communications. The graph based structure has allowed the creation of a protocol that uses the graph instead of having a special protocol for each individual case.

Some of the existing merging schemes have two distinct stages where, in the first stage all vehicles on one of the platoons open up the required gaps, then the rest is done sequentially. This type of two stage gap opening works in cases where the total number of vehicles in the merging platoon is relatively small, but when two really big platoons need to merge with each other this structure can cause the traffic to come to a full stop. The merging protocol proposed in this chapter parametrizes the number of vehicles that can merge simultaneously, which is represented with N. Effects of different values of N was also investigated in this chapter.

Also a blind-spot avoidance function has been proposed to avoid any two vehicles cruising at the same longitudinal position. This function was used to decrease any potential conflicts that might occur due to faulty pairing.

Chapter 7

Summary and Future Directions

This thesis has studied the control and applications of platoon formations by presenting new control structure designs, and protocols where these platoon control algorithms can be used to alleviate traffic congestions. An in-depth analysis of the literature on platoon control structures and merging applications has been presented. *It has been noted that there still is no consensus on an optimal cooperative adaptive cruise control structure for platooning purposes.* Most notably, the recently emerging field of bidirectional control structures in platoon formations is left mostly untouched.

Two distinct bidirectional control structures have been proposed in this thesis. Chapter 3 presented a novel adaptive MPC based control structure for bidirectional CACC. MPC is selected for this purpose because it has the ability to work within the boundaries of the vehicle actuator. This control structure is designed to work with a parameter estimator so that the control output is not affected from plant parameter variations. Even though this structure gives promising results, high computational cost is a put off for most car manufacturers. An alternative bidirectional control structure is proposed in Chapter 4. The control structure presented in this chapter is based on PD control and switching, which is used increase the robustness of the controller to sensor noise. Effects of asymmetric vehicle weighting are also discussed in this chapter. *Our findings conclude that having the weight of the front vehicle higher than the rear vehicle strengthens the string-stability while lowering both the steady-state and transient position errors.* Simulation results have shown that the proposed asymmetric bidirectional control structure can maintain a stronger string stability than other highly regarded unidirectional CACC structures.

Additionally, the literature survey revealed another area that has been overlooked. Modeling of lane changes and lateral control structures for lane change maneuvers have

been popular for a while due to immediate applications, while the interest in the area of merging strategies have been left mostly untouched, except for on-ramp merging. Merging strategies for on-ramp merging started gaining interest in the mid-1990s and still is an area of interest. *It has been discovered that the area of more general platoon merging applications received very little interest until recent years.*

Chapter 5 presented a single vehicle-lane merging structure where vehicles do not have the proper channels of communication. This work has addressed an important need of semi-autonomous vehicles, as the early steps of developing autonomous highways. The main goal of this merging structure is to avoid any possible collision that might arise due to miscommunication, rather than optimizing traffic efficiency.

The single vehicle merging scheme proposed in Chapter 5 has been used as a template in creating a protocol in Chapter 6 for a platoon merging scheme where two platoons need to completely merge due to a lane closure. A graph-based communication structure model is proposed to represent the interactions of vehicles. The use of graph-based connectivity map gives the ability to form a protocol without presetting roles within the platoon. The proposed structure is designed to work without any additional road side units, which decreases the cost of implementation. This chapter has also proposed a blind spot avoidance function, which is used to avoid any two vehicles to be in a side-by-side configuration in their steady state. Simulations done with this structure demonstrate that the blind-spot avoidance function has no adverse effects to the string-stability of the platoon structure.

This thesis proposed two distinct bidirectional CACC structures along with a unified merging scheme to facilitate platoon merging. Some of the potential future works tat can be studied as continuation of this thesis are as follows:

i The parameter estimator proposed in Chapter 3 gave promising results in all the simulations, however it is not tested on real-life vehicles or with high-fidelity car models. Having performance results with a higher-fidelity model is required before actual testing of this estimator.

ii For both MPC schemes presented in Chapter 3, fixed prediction horizon is used. Even though this had yielded to good simulation results, having a variable prediction horizon, which depends on the speed of ego-vehicle, can be used. The horizon length can be decreased to lower the computational demands of the MPC during lower speeds, and increased for higher speeds.

iii The switching based control scheme presented in Chapter 4 showed good simulation results for the high-fidelity vehicle model and was able to maintain string stability for

all simulation cases. A formal string-stability analysis of the proposed control scheme is currently being studied.

iv The control designs presented in Chapter 6 can be expanded to designs in environments with three or more lanes. The presented form of the control scheme already enables the transition from two lane to three lane easier, utilizing the presented three lane connectivity graph.

v The blind-spot avoidance function proposed in Chapter 6 is designed to work for multi-lane roads, but is only tested for two lane roads. How string stability is affected by the use of blind-spot avoidance on roads with three or more lanes should be investigated.

vi The road closure counter introduced in Chapter 6 parametrizes the number of vehicles that can merge simultaneously. Simulation results presented in this thesis give some insight to how this number effects the minimum speed of the platoon along with the time required to complete the merge, however more tests need to be run on platoons with different sizes and configurations in order to reach more conclusive results. Another option is to make the road closure counter and active parameter, which can be set by the platoon leader on-the-go, instead of having its maximum value set prior to implementation. This configuration is expected to yield to better performances if the total size of the merging platoon is known by the platoon leader, which is assumed not to be known in this work.

Bibliography

[1] T. H. A. Van den Broek, J. Ploeg and B. Netten, "Advisory and Autonomous Cooperative Driving Systems", *In Proc. IEEE International Conference on Consumer Electronics*, pp. 279-280, 9-12 Jan. 2011.

[2] B. Van Arem, H. Driever, P. Feenstra, J. Ploeg, G. Klunder, I. Wilmink, A. Zoutendijk, Z. Papp and B. Netten, "Design and Evaluation of an Integrated Full-Range Speed Assitant", Technical Report, TNO, the Netherlands, 2007.

[3] G. J. L. Naus, R. P. A. Vugts, J. Ploeg, M. J. G. van de Molengraft and M. Steinbuch, "String-Stable CACC Design and Experimental Validation: A Frequency-Domain Approach", *IEEE Transactions on Vehicular Technology*, vol. 59, no. 9, pp. 4268-4279, Nov. 2010.

[4] R. Kianfar, B. Augusto, A. Ebadighajari, U. Hakeem, J. Nilsson, A. Raza, R. S. Tabar et al, "Design and Experimental Validation of a Cooperative Driving System in the Grand Cooperative Driving Challenge", *IEEE Intelligent Transportation Systems*, vol.13, no.3, pp. 994-1007, Sept. 2012.

[5] R. Scarinci and B. Heydecker, "Control Concepts for Facilitating Motorway On-Ramp Merging Using Intelligent Vehicles", *Transport Reviews*, vol.34, no.6, pp. 775-797, 2014.

[6] C. Englund, L. Chen, J. Ploeg, E. Semsar-Kazerooni, A. Voronov, H. H. Bengtsson and J. Didoff., "The Grand Cooperative Driving Challenge 2016: Boosting the Introduction of Cooperative Automated Vehicles", *IEEE Wireless Communications*, vol. 23, no. 4, pp. 146-152, August 2016.

[7] F. E. Sancar, B. Fidan, J. P. Huissoon and S. L. Waslander, "MPC Based Collaborative Adaptive Cruise Control with Rear End Collision Avoidance", *In Proc. IEEE Intelligent Vehicle Symposium*, pp. 516-521, 8-11 June 2014.

[8] F. E. Sancar, B. Fidan and J. P. Huissoon, "Deadzone Switching Based Cooperative Adaptive Cruise Control with Rear-End Collision Check", *In Proc. IEEE International Conference on Advanced Robotics*, pp. 283-287, 2015.

[9] R. Pueboobpaphan and B. Van Arem, "Driver and Vehicle Characteristics and Platoon and Traffic Flow Stability", *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2189, pp. 89-97, 2010.

[10] E. van Nunen, M. R. J. A. E. Kwakkernaat, J. Ploeg and B. D. Netten, "Cooperative Competition for Future Mobility", *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1018-1025, Sept. 2012.

[11] R. Kianfar, P. Falcone and J. Fredriksson, "A Receding Horizon Approach to String Stable Cooperative Adaptive Cruise Control", *In Proc. 14th International IEEE Conference on Intelligent Transportation Systems*, pp. 734-739, 2011.

[12] S. P. Sahiyan, S. S. Kumar and A. I. Selvakumar, "A Comprehensive Review on Cruise Control for Intelligent Vehicles" *International Journal of Innovative Technology and Exploring Engineering*, vol. 2, no. 5, April 2013.

[13] C. Bergenhem, S. Shladover, E. Coelingh, C. Englund and S. Tsugawa, "Overview of Platooning Systems", *In Proc. of the 19th ITS World Congress*, 22-26 Oct. 2012.

[14] E. Chan, Eric, P. Gilhead, P. Jelinek, P. Krejci and T. Robinson, "Cooperative Control of SARTRE Automated Platoon Vehicles", *In Proc. of the 19th ITS World Congress*, 22-26 Oct. 2012.

[15] S. Kato, S. Tsugawa, K. Tokuda, T. Matsui and H. Fujii, "Vehicle Control Algorithms for Cooperative Driving with Automated Vehicles and Intervehicle Communications", *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 3, pp. 155-161, Sep. 2002.

[16] Q. T. Luong, J. Weber, D. Koller and J. Malik, "An Integrated Stereo-Based Approach to Automatic Vehicle Guidance", *In Proc. IEEE International Conference on Computer Vision*, pp. 52-57, 1995.

[17] W. Knight, "Driverless Cars Are Further Away Than You Think", *MIT Technology Review*, 22 Oct. 2013.

[18] G. Naus, R. Vugts, J. Ploeg, R. van de Molengraft and M. Steinbuch, "Cooperative Adaptive Cruise Control, Design and Experiments", *In Proc. American Control Conference*, pp. 6145-6150, 2010.

[19] J. E. Naranjo, C. Gonzalez, J. Reviejo, R. Garcia and T. de Pedro, "Adaptive Fuzzy Control for Inter-Vehicle Gap Keeping", *IEEE Transactions on Intelligent Transportation Systems*, vol. 4, no. 3, pp. 132-142, Sept. 2003.

[20] D. Swaroop, J. K. Hedrick and S. B. Choi, "Direct Adaptive Longitudinal Control of Vehicle Platoons", *IEEE Transactions on Vehicular Technology*, vol. 50, no. 1, pp. 150-161, Jan 2001.

[21] W. Batayneh, O. Al-Araidah, K. Bataineh and A. Al-Ghasem, "Fuzzy-Based Adaptive Cruise Controller with Collision Avoidance and Warning System", *Mechanical Engineering Research*, vol. 3, no. 1, pp. 143 , 2013.

[22] M. A. Olivares-Mendez, P. Campoy, I. Mellado-Bataller, I. Mondragon, C. Martinez and J. L. Sanchez-Lopez, "Autonomous Guided Car Using a Fuzzy Controller", *Recent Advances in Robotics and Automation*, Springer Berlin Heidelberg, pp. 37-55, 2013.

[23] M. Senturk, Ismail Meri Can Uygan and L. Guvenc, "Mixed Cooperative Adaptive Cruise Control for light commercial vehicles", *In Proc. IEEE International Conference on Systems, Man and Cybernetics*, pp. 1506-1511, 2010.

[24] J. Ploeg, B. T. M. Scheepers, E. van Nunen, N. van de Wouw and H. Nijmeijer, "Design and Experimental Evaluation of Cooperative Adaptive Cruise Control", *In Proc. IEEE Conference on Intelligent Transportation Systems*, pp. 260-265, 2011.

[25] K. Lidstrom, K. Sjoberg, U. Holmberg, J. Andersson, F. Bergh, M. Bjade and S. Mark, "A Modular CACC System Integration and Design", *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1050-1061, Sept. 2012.

[26] L. Guvenc, I. M. C. Uygan, K. Kahraman et al., "Cooperative Adaptive Cruise Control Implementation of Team Mekar at the Grand Cooperative Driving Challenge", *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1062-1074, Sept. 2012.

[27] M. H. Lee, H. G. Park, S. H. Lee, K. S. Yoon and K. S. Lee, "An Adaptive Cruise Control System for Autonomous Vehicles", *International Journal of Precision Engineering and Manufacturing*, vol. 14, no. 3, pp. 373-380, 2013.

[28] I. Bayezit, T. Veldhuizen, B. Fidan, J. P. Huissoon and H. Lupker, "Design of String Stable Adaptive Cruise Controllers for Highway and Urban Missions", *In Proc. 50th Annual Allerton Conference on Communication, Control and Computing*, pp. 106-113, 2012.

[29] V. L. Bageshwar, W. L. Garrard and R. Rajamani, "Model Predictive Control of Transitional Maneuvers for Adaptive Cruise Control Vehicles", *IEEE Vehicular Technology*, vol.53, no.5, pp. 1573-1585, Sept. 2004.

[30] D. Corona and B. De Schutter, "Adaptive Cruise Control for a SMART Car: A Comparison Benchmark for MPC-PWA Control Methods", *IEEE Control Systems Technology*, vol.16, no.2, pp. 365-372, March 2008.

[31] G. Naus, J. Ploeg, R. van de Molengraft and M. Steinbuch, "Explicit MPC Design and Performance-Based Tuning of an Adaptive Cruise Control Stop-&-Go", *In Proc. IEEE Intelligent Vehicles Symposium*, pp. 434-439, 4-6 June 2008.

[32] F. Bu; H. S. Tan and J. Huang, "Design and Field Testing of a Cooperative Adaptive Cruise Control System", *In Proc. American Control Conference*, pp. 4616-4621, 2010.

[33] D. Swaroop, J. K. Hedrick, C. C. Chien and P. Ioannou, "A Comparision of Spacing and Headway Control Laws for Automatically Controlled Vehicles", *Vehicle System Dynamics*, vol. 23, no. 1, pp. 597-625, 1994.

[34] S. Klinge and R. H. Middleton, "Time Headway Requirements for String Stability of Homogeneous Linear Unidirectionally Connected Systems", *In Proc. 48h IEEE Conference on Decision and Control*, pp. 1992-1997, 2009.

[35] M. R. I. Nieuwenhuijze, "String Stability Analysis of Bidirectional Adaptive Cruise Control", *Eindhoven University of Technology*, 2010.

[36] V. Butakov and P. Ioannou, "Personalized Driver/Vehicle Lane Change Models for ADAS", *IEEE Transactions on in Vehicular Technology*, vol.64, no.10, pp. 4422-4431, Oct. 2015.

[37] A. Kesting, M. Treiber and D. Helbing, "MOBIL: General Lane Changing Model for Car-Following Models", *In Proc. Transportation Research Board Annual Meeting*, pp. 86-94, 1999.

[38] D. Bevly, X. Cao, M. Gordon et al., "Lane Change and Merge Maneuvers for Connected and Automated Vehicles: A Survey", *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 105-120, March 2016.

[39] C. Yang, M. Milacic and K. Kurami, "A Longitudinal Control Concept For Merging Of Automated Vehicles", *In Proc. Intelligent Vehicles Symposium*, pp. 408-413, 1993.

[40] P. Kachroo and Z. Li, "Vehicle Merging Control Design for an Automated Highway System", *In Proc. IEEE Intelligent Transportation Systems*, pp. 224-229, 1997.

[41] W. Cao, M. Mukai, T. Kawabe, H. Nishira and N. Fujiki, "Cooperative Vehicle Path Generation During Merging Using Model Predictive Control with Real-Time Optimization", *Control Engineering Practice*, vol. 34, pp. 98-105, 2015.

[42] H. C. H. Hsu and A. Liu, "Platoon Lane Change Maneuvers for Automated Highway Systems", *In Proc. IEEE Conference on Robotics, Automation and Mechatronics*, pp. 780-785, vol.2, 2004.

[43] M. Goli and A. Eskandarian, "Evaluation of a Multi-Vehicle Merging Strategy Under Different Lateral Maneuvers in the Presence of Sudden Braking", *In Proc. ASME Dynamics Systems and Control Conference*, Oct. 2015.

[44] S. Halle and B. Chaib-Draa, "A Collaborative Driving System Based on Multiagent Modelling and Simulations", *Transportation Research Part C: Emerging Technologies*, vol. 13, no. 4, pp. 320345, Aug. 2005.

[45] B. Van Arem, C. J. G. van Driel and R. Visser, "The Impact of Cooperative Adaptive Cruise Control on Traffic-Flow Characteristics", *Intelligent Transportation Systems*, vol. 7, no. 4, pp. 429-436, Dec. 2006.

[46] L. Davis, "Effect of Adaptive Cruise Control Systems on Mixed Traffic Flow Near an On-Ramp", *Physica A: Statistical Mechanics and its Applications*, vol. 379, no. 1, pp. 274290, June 2007.

[47] Z. Wang, L. Kulik and K. Ramamohanarao, "Proactive Traffic Merging Strategies for Sensor-Enabled Cars", *In Proc. ACM VANET 07*, pp. 3948, 2007.

[48] A. Kanavalli, L. V. Udaya Ranga, A. G. Sathish, P. Deepa Shenoy, K. R. Venugopal and L. M. Patnaik, "Proactive Sliding-Window Strategy for Merging Sensor-Enabled Cars", *In Proc. IEEE Conference on Networks*, pp. 1-5, 2008.

[49] B. Choi, S. Lin and E. S. Peters, "Extended Driver-Assisted Merging Protocol".

[50] R. Pueboobpaphan, F. Liu and B. Van Arem, "The Impacts of a Communication Based Merging Assistant on Traffic Flows of Manual and Equipped Vehicles at an On-Ramp Using Traffic Flow Simulation", *In Proc. IEEE Intelligent Transportation Systems*, pp. 1468-1473, 2010.

[51] R. Scarinci, B. Heydecker and A. Hegyi, "Analysis of Traffic Performance of a Ramp Metering Strategy Using Cooperative Vehicles", *In Proc. IEEE Intelligent Transportation Systems*, pp. 324-329, 2013.

[52] S. Sivaraman, M. M. Trivedi, M. Tippelhofer and T. Shannon, "Merge Recommendations for Driver assistance: A Cross-Modal, Cost-Sensitive Approach", *In Proc. IEEE Intelligent Vehicles Symposium*, pp. 411-416, 2013.

[53] M. Segata, B. Bloessl, S. Joerer, F. Dressler and R. L. Cigno, "Supporting Platooning Maneuvers Through IVC: An Initial Protocol Analysis for the JOIN Maneuver", *In Proc. IEEE Wireless On-demand Network Systems and Services*, pp. 130-137, 2014.

[54] R. Dang, J. Wang, S. E. Li and K. Li, "Coordinated Adaptive Cruise Control System With Lane-Change Assistance", *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2373-2383, Oct. 2015.

[55] M. Antoniotti, A. Desphande and A. Girault, "Microsimulation Analysis of Multiple Merge Junctions Under Autonomous AHS Operation", *In Proc. IEEE Intelligent Transportation System*, pp. 147152, 1997.

[56] B. Ran, S. Leight and B. Chang, "Microscopic Simulation Analysis for Automated Highway System Merging Process", *Transportation Research Record*,vol. 1651, pp. 98-106, 1998.

[57] B. Ran, S. Leight and B. Change, "A Microscopic Simulation Model for Merging Control on a Dedicated-Lane Automated Highway System", *Transportation Research Part C: Emerging Technologies*, vol. 7, no. 6, pp. 369-388, 1999.

[58] X.-Y. Lu, H.-S. Tan, S. E. Shladover and J. K. Hedrick, "Automated Vehicle Merging Maneuver Implementation for AHS", *Vehicle System Dynamics*, vol. 41, no. 2, pp. 85107, Feb. 2004.

[59] Y. Wang, E. Wenjuan, W. Tang, D. Tian, G. Lu and G. Yu, "Automated On-Ramp Merging Control Algorithm Based on Internet-Connected Vehicles", *IET Intelligent Transport Systems*, vol. 7, no. 4, pp. 371-379, 2013.

[60] V. Milanes, J. Godoy, J. Villagra and J. Perez, "Automated On-Ramp Merging System for Congested Traffic Situations", *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 500-508, June 2011.

[61] D. Marinescu, J. urn, M. Bouroche and V. Cahill, "On-Ramp Traffic Merging Using Cooperative Intelligent Vehicles: A Slot-Based Approach", *In Proc. IEEE Intelligent Transportation Systems*, pp. 900-906, 2012.

[62] T. Awal, L. Kulik and K. Ramamohanrao, "Optimal Traffic Merging Strategy for Communication and Sensor-Enabled Vehicles", *In Proc. IEEE Intelligent Transportation Systems*, pp. 1468-1474, 2013.

[63] J. Wei, J. M. Dolan and B. Litkouhi, "Autonomous Vehicle Social Behavior for Highway Entrance Ramp Management", *In Proc. IEEE Intelligent Vehicles Symposium*, pp. 201-207, 2013.

[64] G. Raravi, V. Shingde and K. Ramamritham, "Automatic Merging of Vehicles: Design, Algorithms, Performance", *Cooperative Robots and Sensor Networks 2015*, Springer International Publishing, pp. 231-255, 2015.

[65] P. Fernandes and U. Nunes, "Multiplatooning Leaders Positioning and Cooperative Behavior Algorithms of Communicant Automated Vehicles for High Traffic Capacity", *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 3, pp. 1172-1187, June 2015.

[66] T. Sakaguchi, A. Uno and S. Tsugawa, "Inter-Vehicle Communications for Merging Control", *In Proc. IEEE International Vehicle Electronics Conference*, pp. 365-370, 1999.

[67] A. Uno, T. Sakaguchi and S. Tsugawa, "A Merging Control Algorithm Based on Inter-Vehicle Communication", *In Proc. IEEE International Conference on Intelligent Transportation Systems*, pp. 783-787, 1999.

[68] D. Baselt, F. Knorr, B. Scheuermann, M. Schreckenberg and M. Mauve, "Merging lanes-fairness through communication", *Vehicular Communications*, vol. 1, no. 2, pp. 97104, 2014.

[69] R. Horowitz, C. W. Tan and X. Sun, "An Efficient Lane Change Maneuver for Platoons of Vehicles in an Automated Highway System", *California Partners for Advanced Transit and Highways (PATH)*, May 2004.

[70] S. Lam and J. Katupitiya, "Cooperative Autonomous Platoon Maneuvers on Highways", *In Proc. IEEE International Conference on Advanced Intelligent Mechatronics*, pp. 1152-1157, 2013.

[71] M. Goli and A. Eskandarian, "A Systematic Multi-Vehicle Platooning and Platoon Merging: Strategy, Control and Trajectory Generation", *In Proc. ASME Dynamic Systems and Control Conference*, Oct. 2014.

[72] E. S. Kazerooni and J. Ploeg, "Interaction Protocols for Cooperative Merging and Lane Reduction Scenarios", *In Proc. IEEE Intelligent Transportation Systems*, pp. 1964-1970, 2015.

[73] H. H. Bengtsson, L. Chen, A. Voronov and C. Englund, "Interaction Protocol for Highway Platoon Merge",*In Proc. IEEE Intelligent Transportation Systems*, pp. 1971-1976, 2015.

[74] A. Bemporad, M. Morari and N. Lawrence Ricker, *Model Predictive Control Toolbox User's Guide*, 2013.

[75] P. Ioannou and B. Fidan, *Adaptive Control Tutorial*, Philadelphia, PA: SIAM, pp. 25-82, 2006.

[76] B. Fidan and B. D. O. Anderson, "Switching Control for Robust Autonomous Robot and Vehicle Platoon Formation Maintenance", *In Proc. 15th Mediterranean Conference on Control & Automation*, pp. 1-6, 27-29 June 2007.

[77] I. Herman, D. Martinec, Z. Hurk and M. Sebek, "Harmonic Instability of Asymmetric Bidirectional Control of a Vehicular Platoon", *In Proc. American Control Conference*, pp. 5396-5401, 2014.

[78] Tass International, *PreScan Manual*, Ver. 7.6.0

[79] L. Wu and X. Chen, "The Automated Vehicle Merging Based on Virtual Platoon", *In Proc. IEEE Conference on Automation and Logistics*, pp. 1938-1941, 2008.

[80] A. Marjovi, M. Vasic, J. Lemaitre and A. Martinoli, "Distributed Graph-Based Convoy Control for Networked Intelligent Vehicles", *In Proc. IEEE Intelligent Vehicles Symposium*, pp. 138-143, 2015.

[81] A. M. Medina and E. S. Kazerooni, "DEL150318 i-GAME D2.2 Generic real-time control system FINAL", *Interoperable GCDC AutoMation Experience*, 2015.