Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

School of Computer Science and Statistics

Highway Merging for Vehicle Platoons in Non-Ideal Conditions

Brendan O'Connell

May 21, 2021

A Dissertation submitted in partial fulfilment
of the requirements for the degree of
MAI (Computer Engineering)

Declaration

I agree that this thesis was completed in line with the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at http://www.tcd.ie/calendar.

I have completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at http://tcd-ie.libguides.com/plagiarism/ready-steady-write.

I agree that this thesis will not be publicly available but will be available to TCD staff and students in the University's open access institutional repository on the Trinity domain only.

Brendan O'Connell

May 2021

Acknowledgements

Firstly, I would like to thank my supervisor Dr. Mélanie Bouroche for all of her support, guidance and patience. Without her constant encouragement, I would never have been able to hone in on the end goal of the project. I would also like to thank all of the contributions from Mohit Garg, without whose advice, I would not have been able to perform my experiments. For a subject matter where Google often shows no results, he assisted me many times in arriving at a solution.

I wish to thank my parents for their loving support and aid in all aspects of the project. From making cups of tea, to proof-reading the final product, you have been with me throughout. Heartfelt gratitude to Caitlin for her counsel and company, she never failed to help me in any way she could.

To all of my friends and colleagues that somehow were part of this, many thanks to all of you. Special thanks to John, Charlie, Jack, Matthew, Dave, Laura, and Aaron for your help throughout the year.

Abstract

Vehicle platooning is one of the promising solutions to improve traffic performance in terms of traffic efficiency, fuel consumption, and road safety. The main objective of a vehicle platoon is to maintain a safe inter-vehicle spacing between adjacent vehicles in such a way that collisions are avoided in the presence of different disturbances and uncertainties. Platoons can be formed and maintained using either a vehicle's on-board sensors, or a combination of sensor information and inter-vehicle communication. Finding a method of controlling a vehicle platoon that can withstand adverse conditions safely is challenging. Platoon control can be differently affected by adverse conditions.

This project investigates the design of adaptive cruise control systems for autonomous vehicles in non-ideal communication and non-ideal actuation conditions. Analysis of the behaviour of a platoon in highway merging scenarios is compared to platoon behaviour in regular highway travel using computer simulations. The string stability of a platoon (i.e. how fluctuations from the leading vehicle propagate into the platoon) is strongly linked to safe behaviour. String stability across a range of non-ideal conditions is investigated for connected and non-connected controllers. The simulation results show that the string stability of a connected controller can be significantly degraded by non-ideal communications. The string stability of a non-connected controller can deteriorate due to non-ideal actuation conditions.

These results could inform future work into the critical factors that determine how adverse conditions may influence the decisions of a platoon controller.

Contents

List of Figures

List of Tables

List of Acronyms

ACC Adaptive Cruise Control. 4, 14, 18–21, 23, 26, 28, 29, 32, 39, 41, 49, 50, 52, 62, 68, 69, 75, 81–83, 92–94, 97–102, 109

CACC Cooperative Adaptive Cruise Control. 4, 14, 18, 20, 21, 24, 26, 28–30, 32, 38–42, 49–52, 61–66, 69, 71–73, 75–77, 81, 83, 84, 89, 90, 93–95, 98–100, 103–105, 107–109

CAV Connected Autonomous Vehicle. 13, 14, 31, 42, 44, 57, 74, 83

DSRC dedicated short range communication. 21, 56

IFT information flow topology. 22, 23, 26, 41, 42, 49, 71, 72, 77, 95, 108

OMNeT++ Objective Modular Network Testbed in C++. 35, 36, 54, 55

PATH Partners for Advanced Transportation Technology. 38–41, 52, 72, 73, 75, 77, 81, 83, 84, 90, 98, 100, 104, 105, 107, 108, 110

Plexe Platooning Extension for Veins. 36–41, 49, 54–56, 71

SUMO Simulation of Urban MObility. 34–37, 54, 55

V2I Vehicle-to-infrastructure. 15, 22, 35

V2V Vehicle-to-vehicle. 14, 15, 21, 35, 40, 49, 50, 56, 61, 62, 75, 81, 83, 87, 88, 97, 107, 108

VANET Vehicular *Ad Hoc* Network. 20, 21, 35

Veins VEhicles In Network Simulation. 35, 36, 54, 55

Chapter 1

Introduction

This chapter will introduce the motivations behind this project. The potential challenges that may be faced over the course of the project will also be outlined. The approach to collecting data and arriving at conclusions will be discussed. The contributions that this project may bring to the field of Connected Autonomous Vehicle (CAV) platooning research are highlighted. Finally, a roadmap for the rest of the report is laid out.

1.1 Motivation

Along with a global trend of urbanisation [2], there has been a growing dependence on road vehicles to access urban centres [3]. Many urban centres have reached a point where roads cannot be built to handle the amount of traffic expected on them, nor can existing roads be expanded due to limited space/resources. Another path to achieving efficiency in road travel is to improve how vehicles are driven on roads. Humans, by nature, drive competitively rather than cooperatively [4]. Finding a way to drive cooperatively would increase the efficiency of road travel [5]. The most promising way forward is to allow computers to control road vehicles, so that we can ensure cooperative driving in order to give the best throughput of traffic on existing roads [6]. A significant challenge is ensuring safety in automated driving.

Safety in an automated vehicle means programming a *controller* which can make decisions to alter the vehicle's speed or path when necessary. There are controllers which allow groups of vehicles to follow behind one another in a *platoon*. An important characteristic of connected systems called *string stability* is commonly associated with safety in CAV systems. *String stable* behaviour is demonstrated by ensuring that the effect of disturbances experienced by a platoon attenuate as they propagate in an upstream direction. The difference between an autonomous vehicle and a CAV is that CAVs use Vehicle-to-vehicle (V2V) communications to assist platoon formation, where autonomous vehicles do not. A platoon of autonomous vehicles that uses CAVs is called a *connected controller*. Connected controllers can be referred to as Cooperative Adaptive Cruise Control (CACC) controllers because they use on board sensors and inter-vehicle messages to inform their platooning decisions. The predecessor to CACC controllers are Adaptive Cruise Control (ACC) controllers, which only use on board sensors to inform platooning decisions. Therefore, V2V communications are another critical factor that affect the decisions that a CAV makes. If a platooning system relies on fast communication to increase safety, the behaviour of that system in poor communication conditions should be investigated. This project will take a particular interest in the behaviour of controllers when a merging manoeuvre is being performed. A merging manoeuvre is when the controller allows a new vehicle to join an existing platoon. These manoeuvres will happen very often in real-world implementations of autonomous driving. As such, the safety of a controller must be maintained during a merging manoeuvre. The goal is to ensure that automated platooning systems can operate safely in real world conditions.

1.2 Challenges

Some of the challenges that will be faced over the course of this project are: accurately representing traffic conditions in experiments, choosing conditions that can be represented in an experiment, and understanding what constitutes either safe or unsafe behaviour to enable analysis of results.

Finding a way to accurately represent the traffic conditions that this project will investigate is a difficult challenge. There are two ways for a platooning system to be investigated, either a controller can be implemented in the real world (using real cars or small-scale replicas), or using computer simulations. If information is collected diligently, both approaches have valid but different uses.

Using computer software to simulate real-world conditions of a platooning system is usually the favoured approach, simply because of fewer associated risks and costs when compared to real-world implementations. Computer simulations come with some limitations, such as a high possibility for inaccurate replication of complex interactions. If the traffic scenario involves human drivers, then a system for emulating a human's driving behaviour would need to be created. If the traffic scenario involves V2V or Vehicle-to-infrastructure (V2I) communications, then network communication conditions would need to be created. These kinds of computer simulations can be difficult to set up and often scale poorly. What is meant by scaling poorly is that simulations that involve only a few vehicles may take seconds of computing time for an average machine, but simulations that involve complex traffic systems and many vehicles can take days of simulating time for high-end machines.

Usually creating a real-world implementation of a platooning system involves a lot of time and money, as both software and hardware need to be programmed to operate in the expected fashion. Any system that is being implemented in a real-world replication should undergo rigorous testing in simulation environments to ensure that the money being spent on the vehicles is not wasted. The advantage of creating a platoon in the real world is that there is little chance for the implementation of any condition to be wildly misleading, for example, how packet loss rate can affect communications. In a real-world implementation of a platooning system, all communication conditions would be realistic because they are being performed in the real world. However, in a computer simulation, communication conditions can only be considered estimates, as the network would be missing many possible real-world interactions. A real-world wireless network (as all V2V and V2I communications are wireless) can experience packet loss or interference very easily, but a simulated environment will behave exactly as it is programmed to behave. Obviously, this kind of control is one of the reasons that simulations are attractive, but it allows for simulations to overlook possible real-world conditions. Because of these kinds of omissions, all results that are gathered from simulations should be interpreted in the context of their simulation environments. This means that results that are collected from a simulation environment are valid when their limitations are acknowledged [7].

The traffic conditions that need to be replicated seem relatively simple; a platoon on a highway and a platoon on a highway that allows a new vehicle to join the platoon. However, creating a platoon involves creating a simulated network. The goal of this project is to examine the performance of different controllers in adverse communication and actuation conditions. Because the communication network is simulated, the adverse communication conditions must be decided upon and then encoded into the simulated network. Finding adverse communication conditions that the simulated network can imitate and also affect the performance of the platoon must be identified in order to create useful experiments. Deciding which conditions should be considered involves not only understanding what can cause non-ideal communications in a network, but also how that can be implemented in a simulation.

Learning what constitutes "safe" behaviour in a platoon of vehicles could potentially be a challenge in this project. String stability is a condition for safety that can be quantified by studying the desired controller inter-vehicle spacing in a platoon and the actual inter-vehicle spacing. However, because string stability is such a specific safety condition, it is very possible that a platoon will begin to act in an unpredictable fashion that is not intended by the controller, but still technically be string stable, or potentially for string stability to be unclear and difficult to estimate. Due to this issue, a broader definition of "safe" behaviour is required. This definition will not be as specific as the definition for string stability, as the possibility for all safety conditions cannot be theorised here. The safety of any given simulation scenario that is not purely based in the examination of string stability will be judged at the author's discretion. If any result displays unsafe behaviour that is not string instability, valid reasoning for labelling the result as unsafe will be provided e.g., a crash occurred.

1.3 Approach

As discussed above, this project will approach the problem of analysing the operation of platooning controllers in non-ideal communication and non-ideal actuation conditions using computer simulations. Utilising computer simulations in the correct way is vital to obtaining meaningful results. Specifically, the approach taken here will be to create two simulation scenarios: one scenario where a platoon is simply travelling along a highway, and another scenario where a platoon is travelling along a highway and a new vehicle attempts to join into the platoon. Once these simulation scenarios are created, they can be used to test the performance of selected platooning controllers by running simulations with each platooning controller in each non-ideal condition in both scenarios and observing the results. This approach involves repeating many steps for slightly different conditions. An advantage of many of the results being generated by changing only one or two variables is that it allows for fair comparison between the variables. A disadvantage of the results being generated in this way, is that the differences can be subtle, and the author's judgement is required to analyse the results. This could lead to the analysis of results being subjective, which is undesirable. In order to combat subjectivity, the reason for each conclusion will be stated explicitly when presented.

1.4 Contributions

The contributions of this project to the field of research surrounding autonomous vehicles are putting theoretical limits on the effect of different non-ideal conditions with respect to a platoon's ability to operate safely. The benefit of having these theoretical values that limit the operation of a platoon is that it allows for further work to be performed into these conditions. It also highlights what kinds of vulnerabilities can be intrinsic to certain platooning controllers or possibly common across a group of platooning controllers. The communication limits can be used to dictate when a real-world implementation of a connected platooning controller should fall back to a communication-less controller. These limits for communication conditions are some of the contributions that this project will make to the state-of-the-art in this field.

Both connected and unconnected controllers are considered in this project. Non-ideal actuation conditions may affect all controllers, and the extent to which they affect the safe operation of a platoon must be identified. If non-ideal actuation conditions are shown to be unsafe for any controller, then it means that for any real-world implementation, a process for identifying the actuation delay of a vehicle must be put in place and monitored constantly by the vehicle's on-board units. If a vehicle is identified to have an actuation delay that can behave unsafely, then there is the possibility that the vehicle must refrain entirely from using its ACC/CACC system. This result could be very important, because, as explained above, there is a strong possibility that all controllers could be affected to some degree by actuation delay. In short, finding possible limits on actuation conditions that affect the operation of a platoon would be useful points of reference for the autonomous driving field of research.

1.5 Roadmap of Report

The structure of this report is to first discuss all of the basic ideas and principles which are needed to understand the content of the report in Chapter 2. Then any work related to this project will be detailed in Chapter 3 so that an understanding of the state-of-the-art in the field of autonomous driving and platoon safety can be formed. After that, the methodology which will be used to collect results and analyse them will be discussed in Chapter 4. Next, the results will be analysed at a low level in Chapter 6. Finally, in Chapter 7, the trends that are revealed by the first analysis of the results will be discussed alongside any limitations of the method by which the results were collected and analysed.

Chapter 2

Background

In order to understand the content of this report, the main principles which will be discussed are laid out in this section. Firstly, the evolution of cruise control systems is discussed in Section 2.1 and Section 2.2. Secondly, the idea of platooning will be introduced in Section 2.3, as it is central to this project. This will also include learning the manner in which platoons are controlled, the controllers' characteristics. Thirdly, Section 2.4 will discuss the concept of string stability. String stability is an area of particular focus for this project, as it is a platooning characteristic that encapsulates a platoon's ability to operate safely even for large platoons of vehicles. Finally, platoon merging manoeuvres will be discussed in Section 2.5, because the behaviour of platooning systems' regular behaviour will be compared to their behaviour during merging manoeuvres.

2.1 ACC-enabled vehicles

Adaptive Cruise Control enabled vehicles are vehicles which use sensors and a central computer to detect a vehicle's position on the road and its position relative to other vehicles on the road. This enables control of the vehicle and safe autonomous driving. ACC-enabled vehicles use any combination of radar, laser, infra-red, and camera-based sensors to collect accurate up-to-date knowledge about the environment they are travelling through. ACC is employed commercially in many forms. Vehicles fitted with ACC capabilities are able to adjust speed (and often direction) in order to drive autonomously. Some implementations are only suited to follow a car preceding them from a safe distance. It has been shown that traffic comprising a mix of manually controlled vehicles and a significant number of ACC vehicles is positively correlated to increased safety and throughput on a road [5]. There are limitations associated with using an ACC-enabled vehicle instead of a manually driven vehicle, such as, unusual scenarios that the vehicle's computer may not be able to handle well. It is extremely difficult for a fully automated system to react well to circumstances that it has not been programmed to react to. This limitation is not exclusive to ACC-enabled vehicles, it extends to all vehicles operated by autonomous controllers. As such, many implementations of ACC insist that human drivers should still pay full attention to the road even when using their automated system to ensure complete safety. All ACC systems allow for the human driver to "kill" the ACC and take over control of the vehicle in scenarios that a human interprets as dangerous, but the ACC system does not.

2.2 CACC-enabled vehicles

Cooperative Adaptive Cruise Control enabled vehicles are equipped with similar sensors to ACC vehicles, however, their biggest advantage is their ability to communicate with one another using wireless Vehicular *Ad Hoc* Networks (VANETs). This communication between vehicles allows for faster and more precise decision making than a human driver or even an ACC-enabled car would be able to perform. This means that CACC-enabled cars can predict dangerous situations and devise ways to avoid them faster than a human driver can, as demonstrated by the mixed traffic analysis performed by Yao *et al.* [8]. CACC-enabled cars are also superior to ACC-enabled cars at times when a cooperative approach is necessary for optimisation e.g. on-ramp merging [9]. Delis *et al.* also showed that mixed traffic of ACC and CACC vehicles yielded superior throughput compared to manually driven cars.

The mechanism by which CACC platoons operate is usually based on the dedicated short range communication (DSRC) protocol. This protocol has different standards in Europe [10] and the United States [11]. For both standards, the physical layer implementation is determined by some variant of the IEEE 802.11 standard. This communication allows for *ad hoc* networks to be created (this is how V2V communications work). It also permits local area networks to be set up (e.g., WiFi for the passengers in CACC-enabled vehicles), and importantly it allows for the connected vehicles to receive information about their route (e.g., points of interest or parking information on their destination).

2.3 Vehicle Platooning

Platooning is a concept which involves connecting groups of ACC/CACC-enabled cars together in a string. Platoons usually consist of a platoon leader and followers; they rarely take up more than one lane of traffic when they are not performing platoon manoeuvres. Cars in platoons are able to establish V2V communication using a wireless VANET. This communication allows for the cars to react to changes in the platoon (e.g. slowing down to avoid an obstacle) much faster than a human driver can. Faster reaction times allows the cars in the platoon to drive closer to one another than traditionally possible. This increases a vehicle's fuel efficiency because of reduced air resistance [12] and it improves overall throughput of a road [5].

Platoons must be able to perform simple manoeuvres such as merging and splitting in order to safely allow vehicles to join and leave the platoon. Platoon leaders control the actions of the platoon as a whole. If a car wishes to merge into, or split from a platoon, the command must be confirmed or denied by the platoon leader. The leader of a platoon centrally controlling the state of a platoon is an important safety feature which prevents conflicting manoeuvres being performed simultaneously, decreasing the likelihood of a disturbance in the platoon. However, a weakness to platoon manoeuvres can be exposed if the platoon leader is experiencing network communication errors. An unreliable communication network can lead to a joiner's message not being received by the leader, or possibly being received but no acknowledgement being sent back to the joiner. This means that a join manoeuvre is extremely sensitive to the communication conditions of the leader. It is worth noting that in many cases, the messages being sent between a platoon leader and a potential joiner to a platoon are the longest distance messages that a platoon would have to deal with in a simple highway scenario (excluding V2I communications).

2.3.1 Platooning Controllers

There are two common usages of the term "controller" in the context of vehicle platooning. Controller can be used to refer to the distributed controller characteristic of a platoon, but it is often used to describe the information flow topology (IFT), distributed controller, and spacing policy of a platooning system. This is because, in most cases, a distributed controller has been characterised, and ideal values for the IFT and spacing policy are known for most scenarios. The distributed controller characteristic defines how the platoon control objectives are achieved. Essentially, the distributed controller allows each vehicle to calculate its desired acceleration (and therefore, position in the platoon). It performs these calculations using the data it collects from its own sensors, and the data that are sent to it in platooning beaconing messages. This project will often be using the definition of controller that refers to all of the platooning characteristics as one. Any point where only the distributed controller is being referred to specifically will be made clear, and any point where the IFT or spacing policy of a controller are being referred to, will be clearly identified. This project deals with several different implementations of a controller, and, as suggested previously, each controller is implemented using the ideal values for IFT and spacing policy that have already been researched.

Spacing Policies There are two kinds of spacing policy that are used in platooning controllers: constant-distance spacing policy and variable-distance spacing policy. The difference between the two is how the distance that vehicles in a platoon must keep from one other is enforced.

In a constant-distance spacing policy (as the name suggests), the distance between a vehicle and its predecessor is set, at say 30 metres. This inter-vehicle distance will not change (unless the vehicles are coming to a halt). This kind of spacing policy is often used in ACC controllers because vehicles using this kind of controller are assumed to be unable to communicate. As such, if the spacing policy were to change, there is no guarantee of consensus on what that new spacing policy should apply across the entire platoon. A constant-distance spacing policy has been shown to easily lose string stability. When ACC-enabled vehicles that use this kind of spacing policy do not have a vehicle in front of them, they default to reaching a goal speed.

In a variable-distance spacing policy the inter-vehicle distance enforced by the platooning controller can change over time. The spacing policy usually changes with respect to platoon speed. This is so that the inter-vehicle distance will allow for a preceding vehicle to apply emergency braking and give the following vehicle enough to time brake safely as well. A common implementation of a variable-distance spacing policy is a constant-time headway. Headway meaning the amount of time in between adjacent vehicles in a platoon. For example, instead of saying that vehicles always need to be 30 metres apart, a constant-time headway spacing policy could dictate that vehicles should always be 0.5 seconds apart. This kind of spacing policy varies with speed and can be enforced more consistently by connected controllers because the connected controllers can propagate the variable spacing policy along the platoon quickly. These kinds of spacing policies have been shown to be better at implementing string stable behaviour than constant distance spacing policies [13].

Information Flow Topologies Platoons have several different characteristics that define the operation of that platoon, such as node dynamics, distributed controller, and spacing policy. Distributed controller and spacing policy have already been discussed. The IFT of a platoon describes how messages propagate through the string of vehicles. An example of a common IFT is predecessor following (PF) [14], whereby messages can only be transferred from a vehicle's predecessor. If the leader of a platoon using PF wishes to send a message to the final follower in the platoon, then the message will propagate through every other vehicle in the platoon.

2.4 String Stability

String stability is a performance indicator used to assess longitudinal vehicle control systems. The concept of string stability has been around for almost 50 years, the term was used first by Peppard [15] and subsequently used in vehicle platooning [16]. String stability must be considered when analysing a platooning scenario [17], as it is a vitally important factor when considering safety and efficiency of a platoon. There are many different rigorous definitions of string stability [14], though, its definition is often considered intuitive. A system that is string stable will not allow disturbances in the system to amplify as they propagate down the platoon. A system is string unstable if a disturbance is amplified (or does not attenuate) as it propagates down the platoon. String stability is best measured using distance between adjacent vehicles (inter-vehicle distance) in a platoon with respect to time. Measuring inter-vehicle distance and the ideal inter-vehicle spacing allows for the error from ideal distancing to be calculated.

From observing the inter-vehicle distance error, the attenuation or amplification of disturbances along the string can be easily observed. In the case of a string stable platoon, as the disturbance propagates down the platoon a decrease in the inter-vehicle distance error should be observed. However, research shows that inter-vehicle spacing error is not the only factor that needs to be considered in string stability when analysing heterogeneous platoons [18, 19]. The definition of heterogeneous traffic in this instance refers to traffic that is all CACC-enabled, but has differing dynamic abilities (e.g., different maximum braking capabilities). This project is solely focused on homogeneous platoons (platoons in which all of the vehicles have the same dynamics) to simplify the estimation of string stability.

An aspect of string stability that must be explored is the *boundedness* of a string stable system. Boundedness ensures that as the size of the platoon grows, no sacrifices are made with respect to string stability. That is to say that if a controller can implement a string stable and bounded platoon, then the platoon may be extended infinitely and always remain string stable. This result would be significant for a controller as it proves that safety can be maintained for extremely large platoons in ideal conditions.

2.5 Platoon Merging Manoeuvre

As mentioned previously, platoons must be able to perform manoeuvres which allow for vehicles to join or leave the platoon. A merging manoeuvre is a manoeuvre in which a vehicle joins a platoon, either by adding on to the end of the platoon or by changing lane into a safe gap within the platoon. Of the two types of merge that are possible, this project will focus solely on the scenario where a vehicle joins a platoon by joining on to the end. This project will only analyse cases where a single vehicle is merging into the multi-vehicle platoon, though there is research on multiple vehicle merges [20].

Figure 2.5.1: Example of a join-at-back manoeuvre captured from the SUMO GUI

It is very important to ensure the safety of all vehicles in the platoon at all times. Different manoeuvres introduce different risks to safety compared to simple leader-following (standard behaviour for a platoon in a highway setting), as such, controllers which purport to guarantee string stability must be rigorously tested in merging cases. Studies have investigated string stability during merging manoeuvres for identical vehicles [21] and vehicles with differing dynamics [22]. However, none of these simulations take into account the possibility of a real-world scenario where communication conditions are not ideal.

2.6 Summary

This section outlined many of the concepts that will be discussed over the course of this project. ACC and CACC controllers are the only controllers that are going to be evaluated. Platooning was explained in general, and the characteristics of platoons that are relevant to this project (distributed controller, spacing policy and IFT) were also detailed in order to facilitate robust analysis of platooning systems. String stability, a concept that is key to understanding the literature in this field of research including this project was explained insofar as is necessary. Finally, the platooning manoeuvre (joining) that will be investigated over the course of this project was outlined.

Chapter 3

Related Work

The previous chapter discussed the main concepts that need to be understood in order to be able to evaluate this project. This chapter will deal with research that has already been done which is within the scope of this project. In Section 3.1 the non-ideal communication conditions that will be investigated during the experimentation portion of this project are compared with other work that has been performed on the same conditions. Non-ideal actuation conditions will be put into the context of the state-of-the-art in this area in Section 3.2. The operation of the simulation platforms that are utilised to perform experiments is detailed in Section 3.3. Finally, the research question which drives this project is outlined in Section 3.4.

3.1 Non-ideal Communication Conditions

All real-world networks experience losses at some point. Communication loss must be taken into account for platooning scenarios, but many papers which analyse string stability of a platoon do so in ideal communication conditions [17, 23, 24, 25].

In this project, several communication variables that give rise to non-ideal communication conditions will be explored. Namely, packet loss rate of the network, transmission power of all of the vehicles in a platoon, and the background noise that is experienced across the network. Each of these communication conditions have the ability to give rise to conditions that could cause a platoon to behave in an unsafe manner. Another area of interest in the state-of-the-art is controller *degradation*. This is the idea that when a connected controller experiences significant communication loss or delay, the controller will degrade to a communication-less controller.

Harfouch *et al.* aim to show that a CACC system can adapt to its platooning environment and deploy different controllers as a result [26]. In their first experiment, a heterogeneous platoon is controlled by a simple CACC controller that cannot adapt to its situation. When the platoon is presented with a disturbance, the controller successfully avoids the disturbance, but the platoon's behaviour was not strictly string stable as the controller was expecting vehicles with identical dynamics. In their second experiment, the heterogeneous platoon was controlled by a novel controller (conceived by Harfouch *et al.*) which is able to adapt to the heterogeneity of the platoon, and it avoided the disturbance, while demonstrating string stable behaviour. In the final experiment, the platoon was exposed to non-ideal communication conditions, and a different novel controller was implemented that was able to adapt the operation of the platoon, to instead use a communication-less ACC controller in place of the CACC controller that was being used previously. This paper is successful in demonstrating that a controller can be created which adapts well to different adverse conditions that a platoon may encounter in a real world scenario.

Ploeg *et al.* introduce the idea of a CACC controller that can detect and respond to non-ideal communication conditions [27]. The paper introduces the phrase *graceful degradation*, which describes the operation of a connected platooning controller that is able to detect non-ideal communication conditions, and safely (gracefully) degrade to a platooning controller that does not require communications to hold a formation. This research was performed on a real platoon of vehicles, which were are all Toyota Prius III Executive. This leads to the graceful degradation paper assuming that the platoon being considered is homogeneous. The experiments to test the string stability of the platoon involved decelerating the leading vehicle by 5 m/s and observing the reaction of the following vehicles. The experiments show that the degraded

CACC (dCACC) system performs at least as well as the ACC system.

Harfouch *et al.* [26] and Ploeg *et al.* [27] are able to demonstrate that a controller can be created which is able to respond to its environment in due time, and make adjustments to the control structure of the platoon that do not affect the string stability of the system. The only issue is that each of them theorise the boundary of operation. They believe that non-ideal communication conditions can be easily identified using communication latency. This is not always the case, for example, during a jamming attack, communications may degrade much faster than the calculations based on latency would be able to detect, causing a platoon to continue using a connected controller despite adverse conditions.

Teo *et al.* describe the operation of any mobile vehicle platoon that experiences losses [28]. Teo *et al.* did not specify that the platoon in question had to be road vehicles, in fact it was suggested that the idea could be used for aeroplanes or road vehicles. Due to this generalisation of "lossy" behaviour, some of the concepts in this paper may need to be adapted to work for platoons of road vehicles. The paper concludes that if a network experiences intermittent losses, string stable behaviour can be maintained if the following vehicles are able to accurately estimate the velocity of the vehicles preceding them, as long as that velocity does not exceed bounds that are detailed in the paper. This is the point where generalising the concept to all moving vehicle platoons benefits road vehicle platoons. All CACC-enabled vehicles have the capability to measure the distance between themselves and their predecessor by using their on-board sensors. Therefore the concepts developed in this paper can be well suited to road vehicle platoons for networks that can experience intermittent losses. An assumption was made over the course of the paper that the leading vehicle's velocity would not alter beyond certain bounds. So it is worth noting that the concepts developed in this paper may be well suited to a platoon that experiences small disturbances, but if a platoon experiences a large disturbance (e.g., the leader suddenly braking to a full stop), these concepts may not hold. This knowledge could possibly be used to create a controller that does not instantly degrade to a string unstable ACC controller if the leading vehicle's position is measured not to change during periods of high packet loss in the network.

Alipour *et al.* discuss the impact of jamming attacks on CACC systems [29]. Unlike most other research performed into non-ideal communication conditions, this paper does not discuss factors that may mitigate the effect of poor communications on a platoon. Instead this paper analyses the position of a jamming attacker relative to a platoon, and which position affects the platoon the most. The jamming attack, in this case, manifests as a large increase in background noise for a localised area in the platoon. The increased noise affects transmissions in a fashion proportional to the distance from the jamming source to the transmission nodes. The paper concludes that when a jamming source is placed very close to the leading vehicle in a platoon, the effects are worse than other positions. This shows that background noise can be artificially increased in order to attack a platoon. Therefore, measures must be put in place to prevent the possibility that a sudden increase in background noise can result in a disturbance causing a crash in a platooning vehicle.

Table 3.1: Analysis of related work that considers non-ideal communication conditions

Paper	**String stability**	**Packet loss rate**	**Background noise**	**Transmission power**	**Degradation**
Ploeg 2014 [27]	y	n	n	n	y
Harfouch 2018 [26]	y	n	n	n	y
Teo 2003 [28]	y	y	n	n	n
Alipour 2017 [29]	y	y	y	n	n

Table 3.1 cross-references each of the topics of research that are discussed in this paper with the current state-of-the-art.

3.1.1 Communication Condition Justification

Each of the conditions were chosen for investigation for a reason. This section will discuss the reasoning behind the choices of: packet loss rate, background noise, and transmission power as the network variables investigated in this project.

Packet loss rate was chosen as a non-ideal network condition worthy of investigation because it is a common manifestation of many different kinds of unreliable communication channels. Packet loss rate can be manifested in a network by many different mechanisms e.g. network congestion, bugs in software, problems with hardware, security threats. Packet loss rate is a powerful network condition to investigate, because if a controller is found to be robust in its handling of high rates of packet loss, then it will be robust for many adverse communication conditions, no matter how they are manifested.

Background noise was chosen as a non-ideal communication condition because its effect on platoon behaviour is not well researched at this point in time. It is a communication condition whose potential impact will only grow as the number of implementations of CAV platoons in the real world increases. It has been investigated in the sense of a jamming attack before. However, if connected platoon controllers become commonplace in the future, the levels of background noise that cause adverse effects must be well understood. This is a reason for investigating i in this project.

It is noteworthy that no papers have investigated the effects of transmission power on platoon communications. This is likely because the effects of low transmission power can be similarly manifested using other variables like background noise. However, due to the fact that it has not been thoroughly investigated before, it should be inspected to either confirm or reject the assumption that it behaves similarly to other communication variables whose effects are proportional to the distance of transmission.

Packet delay is another valid communication condition that could be investigated. However, connected controller degradation has already been calculated using packet delay by Ploeg *et al.* [27].

3.2 Non-ideal Actuation Conditions

The quality of a vehicle is never guaranteed, engine conditions can easily degrade over time if they are not maintained. If actuation conditions can drastically affect the performance of a platoon, then there are two possible approaches to mitigating the effects of actuator delay on a platooning system. Firstly, a controller which is robust enough to withstand large amounts of actuator delay could be developed. Secondly, actuator delay could be monitored and maintained below some ideal value in platooning situations. A lot of research performed into connected platoons assumes the actuation delay of a vehicle to be some small constant or negligible value, so the delay is often omitted from calculations [24, 23, 30]. In reality, pneumatic braking systems involve actuation delay because it is an unavoidable physical phenomenon [31]. Therefore actuation delay should be accounted for as a variable across platooning systems. It is possible that vehicles are adapted to suit autonomous driving have a weakness that causes their actuation delay to degrade over time. This degradation of many ACC/CACC enabled vehicles could lead to a situation whereby an entire platoon of vehicles experience actuation delay much greater than their (negligible) expected value.

Di Bernardo *et al.* propose a consensus approach to platooning that involves allowing for actuation delay (it is referred to as actuation lag in the paper) [32]. Actuation delay is considered in the case that the vehicle is being driven by a human, but the controller still wishes to be able to alert the human to possible dangers if they are driving too close to a vehicle in front of them. In order to calculate a safe driving distance, the actuation delay of the vehicle and the human reaction time is taken into account. An example more relevant to the work of this project is shown later in the paper, where the actuation delay of a vehicle is used in calculating the ideal inter-vehicle spacing inside a heterogeneous platoon.

Morbidi *et al.* [21] approach the string stability of a platoon in an analytical manner. In doing so, several equations are derived so that sufficient conditions for string stability can be defined. The value for actuation delay is often ignored in these analytical approaches, however, Morbidi *et al.* allow for actuation delay in their approach, and the value for actuation delay is included as a variable in their sufficient conditions for string stability. As mentioned, that work differs from this project, as it was an analytical approach rather than an experimental approach, so no estimates of what range of values for actuation delay may cause string unstable behaviour were included.

In Valente's PhD thesis [33], computer simulations are used to evaluate the string stability of different systems. Actuation delay is included as a factor that can effect string stability. Using empirical methods, Valente was able to estimate a function to describe actuation delay of a given engine type at a known speed. This was used as part of a calculation to estimate the overall lag experienced by a real-world vehicle due to mechanical processes. As such, a range of values for actuation delay that may affect string stability were included.

Table 3.2: Analysis of related work that considers non-ideal actuation conditions

Paper	**String stability**	**Analytic approach to actuation delay**	**Experimental approach to actuation delay**
Di Bernardo 2015 [32]	y	y	n
Morbidi 2013 [21]	y	y	n
Valente 2015 [33]	y	n	y

As demonstrated in Table 3.2, most of the work that has been performed to date investigating actuation delay as a factor that can affect the string stability of a platoon is performed using analytical methods. This paper will take a different approach to Valente, but will still use experimental methods to investigate the effects of actuation delay on string stability.

3.3 Simulation Platforms

An integral part of this project is the experimental evaluation of different platoon controllers for different communication conditions. As this project does not have the available resources to implement the platoon controllers in a real-world scenario, computer simulations will be used to implement the platoon controllers. There is a lot of flexibility associated with using computer simulations in this manner, as in real-world implementations, the cost of a failure of the system would be large, but for a computer simulation, the failure of an implementation is easily handled.

SUMO

There are several different simulators that are commonly used to reproduce traffic conditions, for example AnyLogic's proprietary "Road Traffic Simulation Software", SimWalk's proprietary "Road Traffic Simulator", and the open-source Eclipse Simulation of Urban MObility (SUMO) project. For this project, a traffic simulator which is able to accurately represent homogeneous traffic travelling along a straight highway is required. The proprietary platforms do not have as many extensions freely available as SUMO does. Because SUMO is open-source, many more programs are able to interact with it, compared to the proprietary programs. This made SUMO an obvious choice for the traffic simulator.

SUMO is designed to be able to handle large networks. SUMO can be used on its own to simulate a range of traffic scenarios where communications are not needed, or where the simulation parameters do not need to vary too widely. For example, if a simulation involved a section of an urban environment with pedestrians and vehicles which are all humanly controlled, then SUMO could be used to simulate it, without the need for any other assistance from other simulation programs. SUMO has a built-in GUI that displays the traffic scenario that is being considered. However, the simulations for this project require a simulated network in order to facilitate platooning messages. Therefore, a network simulator also needs to be identified.

OMNeT++

Objective Modular Network Testbed in C++ (OMNeT++) is a discrete event simulator that is widely used to simulate networks and network messages. It can work alongside the SUMO simulator in order to provide V2V or V2I communications to a traffic simulation. It is perfectly suited to create a VANET for this project, as it allows for non-ideal communication conditions to be realised over the course of a simulation. OMNeT++ also comes with a built-in GUI. A difference between SUMO and OMNeT++ is that a SUMO simulation can be started from the OMNeT++ GUI but an OMNeT++ simulation cannot be started from the SUMO GUI. When a SUMO simulation is run using the OMNeT++ GUI, every message that is being sent over the vehicular network can be observed. This is a powerful tool that can be used to discern information flow topologies of controller implementations if they are unknown, it can also be used to read the content of any given message over the course of a simulation.

Veins

The VEhicles In Network Simulation (Veins) environment requires both SUMO and OMNeT++. This is because the Veins simulating environment gives a platform upon which SUMO and OMNeT++ can be integrated together easily and coherently. Veins allows for easy simulations which require vehicles/infrastructure to be able to communicate.

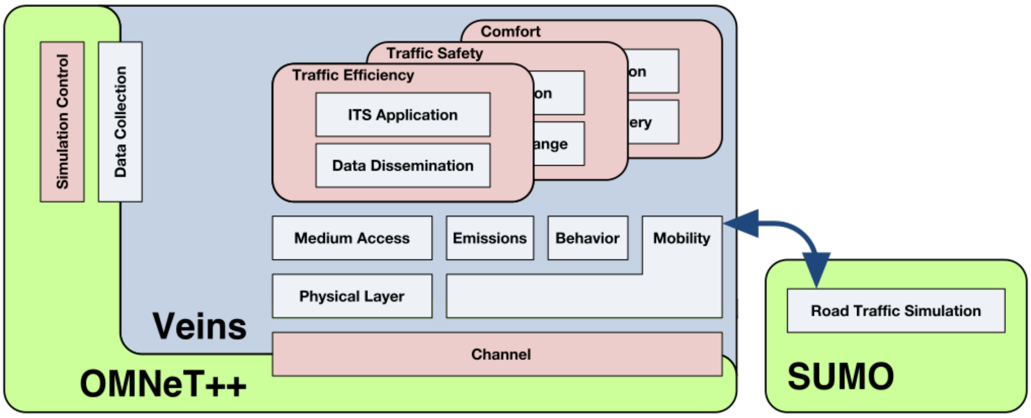

Figure 3.3.1: Veins architecture. Source: [1]

Veins operates as an interface between SUMO and OMNeT++ by acting as an extension to OMNeT++, as demonstrated in Figure 3.3.1, that can perform all of the functionalities of the network simulator that are necessary to implement vehicular networks. Veins communicates with SUMO through sharing a port on the localhost server. The port is determined at runtime and both platforms can read and write commands to one another over the shared port in order to facilitate an accurate simulation experience.

Plexe

Platooning Extension for Veins (Plexe) is an extension of the Veins simulator which provides the ability to form platoons, create platooning messages, and perform platoon manoeuvres. This interface will be used extensively throughout this project to simulate platooning scenarios. It is important to note that simulations which are run using this platform are deterministic. As such, simulating a single scenario multiple times does not offer new information and therefore, simulations will only be run once only (no repeats).

Plexe uses C++ to perform the business logic of each of its features, and a configuration file to initialise all of the conditions that can change from simulation to simulation. When Plexe creates a simulation scenario, each of the conditions that are initialised in the configuration file are used by a C++

class to implement the conditions in the scenario.

The base classes that Plexe simulations require are: a position helper, a platooning application, a platooning protocol, a traffic manager, and a scenario. The position helper contains the definition of methods that are used to assist the platooning application in discerning things like the position of a particular vehicle in a platoon or the ID of the leader of a platoon to allow vehicles to send messages to the leader. The platooning application dictates the handling of platooning messages, for example, a platooning application would define how platoon beaconing messages or join manoeuvre messages should be interpreted. The platooning protocol defines the operation platoon beaconing messages. The traffic manager uses SUMO's traffic control interface (TraCI) to change variables that SUMO needs to be informed about such as, the number of vehicles in a simulation, the controller that each vehicle should use, etc. The scenario defines the behaviour of a given simulation, for example, a scenario could be used to apply a disturbance in a simulation, or to permit a new vehicle to join the platoon.

To implement each of the scenarios in the experiments of this project, becoming familiar with this structure of C++ classes was vital. Creating a new scenario which implements a disturbance (either sinusoidal or full-braking) and a join manoeuvre would not have been possible without an understanding of this structure. New scenario, and platooning application classes were needed to implement that kind of behaviour.

The new scenario class that was created to allow for join manoeuvres with disturbances was based on a simple join-at-back merging scenario, but it had to be able to accept new arguments from the Plexe configuration file to allow for disturbances to be added. To facilitate sinusoidal disturbances, a boolean option was added to the configuration file. Along with this boolean option, other options which controlled the amplitude, frequency, and duration of the sinusoidal disturbance were added. A different boolean option was added to allow for a full-braking disturbance. In the same fashion as the sinusoidal disturbance, other options were added to control the maximum rate of braking, and the simulation time at which the disturbance occurred.

The new platooning application class that was added, assisted the new scenario class in achieving the goal of creating a merging manoeuvre with disturbances, by allowing the platoon controller's speed to be altered by the previously mentioned scenario options. Another vital piece of code that was added to the platooning application was implementing the correct controller in the new joining vehicle. Plexe is a relatively new software, and some of the CACC controllers have only been added to the library in the last 6 months. As such, some of the behaviours are hard-coded to operate for only the PATH CACC controller (the first connected controller implemented). This meant that other options were added to the configuration file, to inform the new platooning application of the desired platooning controller for each simulation, so that the joining vehicle could be given the correct controller once it successfully joins the platoon. The application handled this controller, and passed it to the traffic manager, which implemented the change in controller.

Plexe allows for the simulation results to be recorded at a chosen interval. The default value for data collection frequency is 100 Hz, that means that 100 times per second over the course of a simulation, Plexe writes the value of each vehicle's distance from its predecessor, speed, acceleration, controller acceleration, and relative velocity to a vector (.vec) file. At the end of each simulation, the vector file is converted into an R data (.Rdata) file which allows for a plotting program to display the results graphically. The graphs that are produced by this data collection are the results that will be analysed.

Table 3.3: Summary of simulation platforms' abilities

Platform	**Vehicle dynamics**	**Network conditions**	**Platooning**
SUMO	y	n	n
OMNeT++	n	y	n
Veins	y	y	n
Plexe	y	y	y

Table 3.3 summarises the abilities of each of the simulation platforms that were considered for this project. Plexe is the simulation platform that will be used to create the experiments for the project because it is the only platform

that can simulate platoons which experience non-ideal communication and non-ideal actuation conditions.

Implementations of Platooning Controllers in Plexe

There are 3 different controllers discussed over the course of this project. Two of these controllers are connected controllers, and the other is a controller that does not require inter-vehicle communications to function. The term controller here refers to the distributed controller, the spacing policy, and the information flow topology characteristics of a platoon. The reason that these three controllers were chosen as the controllers in this project is because the two connected controllers (Ploeg and PATH) are state-of-the-art, and have been implemented in real-world platoons. This makes them perfect for testing against potential real-world scenarios as they claim to be fit for the purpose. It is also convenient that they have been implemented in Plexe. The ACC controller is a state-of-the-art implementation of an ACC controller. It is a perfect example of a communication-less controller that can be compared against the two selected CACC controllers.

ACC The ACC controller implemented in Plexe [34] is described as a simple interaction between the cruise control of the vehicle (acceleration calculated using the desired cruising speed) and the sensor-informed controller (desired acceleration calculated using sensor information). The acceleration at any given moment is calculated as the minimum of these two values. This feature ensures that when a vehicle is below its desired cruising speed and the sensors inform the controller that there is room in front of the vehicle to accelerate, then it will do so. If the sensors inform the controller that there is no space to accelerate to its desired speed, then the controller calculates a new acceleration (or deceleration) to ensure the vehicle is safe. Another example of the ACC controller in action would be that if a vehicle in front of the ACC-enabled vehicle accelerates beyond the desired cruising speed, then the controller will not exceed the desired cruising speed.

This controller uses a constant-time gap spacing policy that is set at 0.4 seconds. There is no information flow topology associated with this controller, as the platoon does not require constant beaconing to remain in formation.

PATH CACC The California Partners for Advanced Transportation Technology (PATH) project developed an implementation of a CACC controller that optimises road throughput and fuel efficiency by forming strings of vehicles on highways using relatively small inter-vehicle spacing [35]. The project claims the ability to yield a 25% reduction in energy consumption compared to normal highway driving. This project is mainly focused on truck platooning, but the principles that were discussed have been extrapolated to suit any kind of vehicle that Plexe can simulate. It is also worth noting that the California PATH project did not implement any functionality with respect to longitudinal control. The project was entirely focused on latitudinal control, and theorised that drivers would still have to perform the steering of vehicles which use this controller mechanism. Due to the project only defining behaviour of latitudinal control, the formation of vehicles is explicitly referred to as a string. They do not use the word platoon to describe a formation where only the speed is under automated control.

The project also theorised the existence of a "CACC truck-only" lane on highways. The project believed that if trucks which are using a small constant time gap spacing policy are deployed on highways with manual drivers, then the manual drivers could become intimidated by the presence of large vehicles that other traffic cannot merge into (forming a kind of wall of trucks). The leading vehicle in the string of trucks is made the leader of the string much like traditional platooning.

The project states that the leader must have the ability to alert its followers to conditions that would cause the followers to have to steer manually. For example, changing lane or avoiding a road hazard. The PATH project also discusses the theoretical existence of what they call a "kill switch". The kill switch alerts the driver that the conditions of the string have become unsafe, and the driver must take over complete manual control of the vehicle. This concept is similar to the idea in this project, that once a V2V communication network becomes unsafe, then connected controllers must use their own kill switch to fall back to a communication-less controller. However, the PATH project explicitly stated that the use of the kill switch was purely for use in research only due to concerns about how the kill switch being activated would affect current implementations of CACC controllers. What is meant by this, is that hitting the kill switch implies that a vehicle can no longer be part of the string, so as soon as the switch is flipped, all vehicles in the string must be made aware of this, and stop trusting all controller commands that they receive. It is the mechanism of how a vehicle in a string reacts to these untrustworthy CACC messages that renders the kill switch only acceptable for use in research.

This controller uses a constant-time gap spacing policy of 0.2 seconds. It also makes use of a leader-predecessor following IFT. A leader-predecessor following IFT is characterised by each vehicle having to receive platooning messages from both the leader of the platoon, and its direct predecessor.

Ploeg CACC The Ploeg CACC controller [30] is so named because of the main author of the paper who laid out its functionality. Ploeg is a prolific researcher in the field of autonomous driving, and in particular string stability in autonomous platooning systems [13, 36, 37, 38, 39]. The paper on which the Plexe implementation of the Ploeg CACC controller is based, compares and contrasts the operation of an ACC-enabled platoon of vehicles to the operation of a CACC-enabled platoon of vehicles. Ploeg points out that much of the research performed on the control design of CACC systems is theoretical analysis, not practical implementations and the evaluation thereof. Ploeg suggests that a constant time gap spacing policy of 1 second or less is required to ensure that traffic throughput and fuel efficiency are maximised in a platooning scenario, and that using connected controllers is the best strategy to achieve these time headways. Ploeg experimented on real vehicles in this paper [30], and as such the results can be interpreted as extremely successful. Implementing a controller in a real-world platoon is a huge leap forward in the development of a connected controller. The vehicles that were used (Toyota Prius III Executive) required only a few components to be added in order for them to function well as a platoon. This is a promising result for CAV implementations in the future. Not having to completely retrofit a vehicle to the purpose of driving autonomously, and using some of the existing components as part of the final product shows that CACC implementations may be commonplace in the near future.

This controller uses a constant-time gap spacing policy of 0.6 seconds. It also makes use of a predecessor-following IFT.

3.4 Research Question

The question that this research project poses is: do non-ideal communication and non-ideal actuation conditions have any significant effect on the behaviour (particularly the string stable behaviour) of different platoon controllers? If they have a significant effect, when do these effects render a particular platooning controller unsafe?

The first question has been asked by several other researchers, but mostly analytic methods were used to answer them. Using an analytic approach, it is difficult to discern what real values may cause the adverse effects that are observed. This leads to the second question; because experimental methods are used in this paper, real values will be possible to find. These real values have not been calculated before, and answering this question could give significant results.

3.5 Summary

In conclusion, a lot of analytical research has been performed in the field of autonomous platooning with respect to both non-ideal communication conditions and non-ideal actuation conditions. Work has already been done in identifying how controllers that are affected by non-ideal conditions can degrade to controllers that are better suited to new conditions. However, very little work has been done to identify the kinds of conditions that may cause a platoon controller to have to degrade to a safer alternative. The objective of this project is to use experimental methods to discern the limits of communication and actuation conditions that may cause a platooning controller to degrade.

Chapter 4

Method

The goal of this project is to investigate the design of merging behaviour for CAVs in non-ideal communication and non-ideal actuation conditions. As such, simulations using a suite of vehicle simulating tools will be run. The results from these simulations will be used to investigate the effects (paying particular attention to string stability) that non-ideal conditions have on a platoon that is not performing a manoeuvre. These effects will be compared to the effects measured when a platoon performing a manoeuvre is simulated.

The simplest way to investigate string stability is to add a disturbance to the system. As mentioned in Section 2.4, the definition for a string stable platoon is defined as a platoon where disturbances which propagate down the string of vehicles do not amplify. This effect can be observed in homogeneous platoons by analysing the inter-vehicle spacing with respect to time, paying particular attention to the times when the platoon leader experiences a disturbance. It is worth noting that the platoon leader is not necessarily the first vehicle to experience a disturbance in a platoon, but for a highway scenario, the platoon leader is far more likely to experience a disturbance than any other vehicle in the platoon.

This section will outline the scenarios that will be simulated for different platooning controllers across different non-ideal conditions so that results can be collected and analysed. This section will also explain how these scenarios are used and why they are a valid approach to collecting the data needed for accurate analysis.

4.1 Sinusoidal Braking Scenario

This simulation scenario is a great test for string stability as it will constantly test the controller's ability to make the correct decisions for the vehicles in the platoon in order to keep them all at the ideal distance from the car in front of them. As non-ideal communication conditions are introduced into the scenario, their effect on the string stability of the platoon should be apparent when analysing the graph which plots inter-vehicle distance with respect to time.

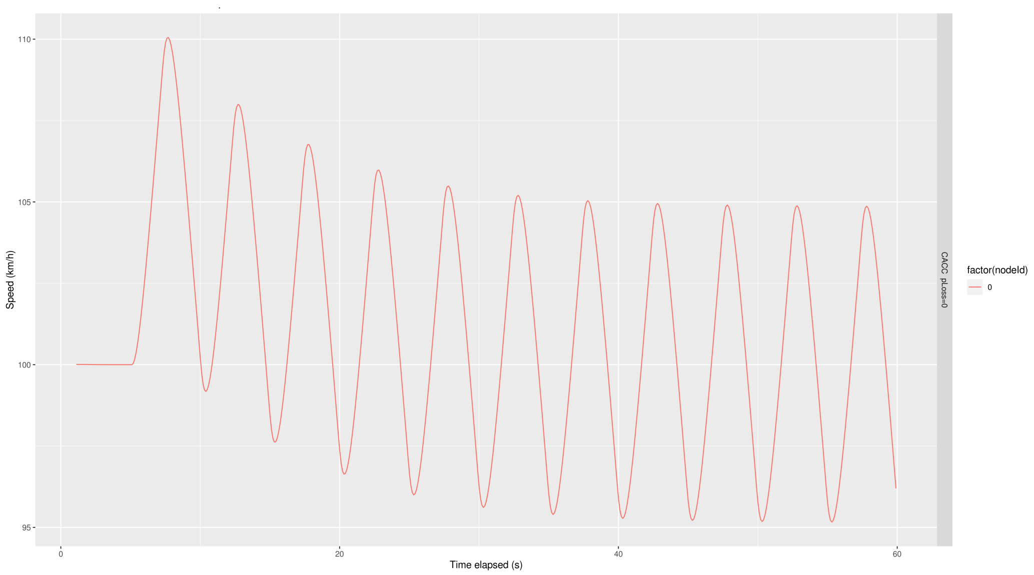

Figure 4.1.1: Sinusoidal disturbance demonstration: vehicle speed w.r.t. time

This scenario involves a platoon of vehicles which has size 8 (meaning there are 7 vehicles following a leader). The platoon leader and all of its followers begin the simulation with a speed of 100 km/h and then after 5 seconds it will start oscillating its speed (accelerating and decelerating) sinusoidally with a frequency of 0.2 Hz, meaning it will accelerate for 5 seconds and then decelerate for 5 seconds repeatedly until the simulation ends. An example of what a sinusoidal disturbance looks likes from the point of view of only the leader is demonstrated in Figure 4.1.1.

4.2 Full Braking Scenario

This simulation scenario is a good test for string stability of a platoon for extreme braking conditions. This is because the controller should always be able to have the followers come to a full stop behind the leader. If behaviour different from a safe full stop is observed, then it can be implied that some real-world disturbance has affected the operation of the controller, and as such, a method to mitigate the disturbance should be investigated. An example of what a full-braking disturbance looks likes from the point of view of only the leader is demonstrated in Figure 4.2.1.

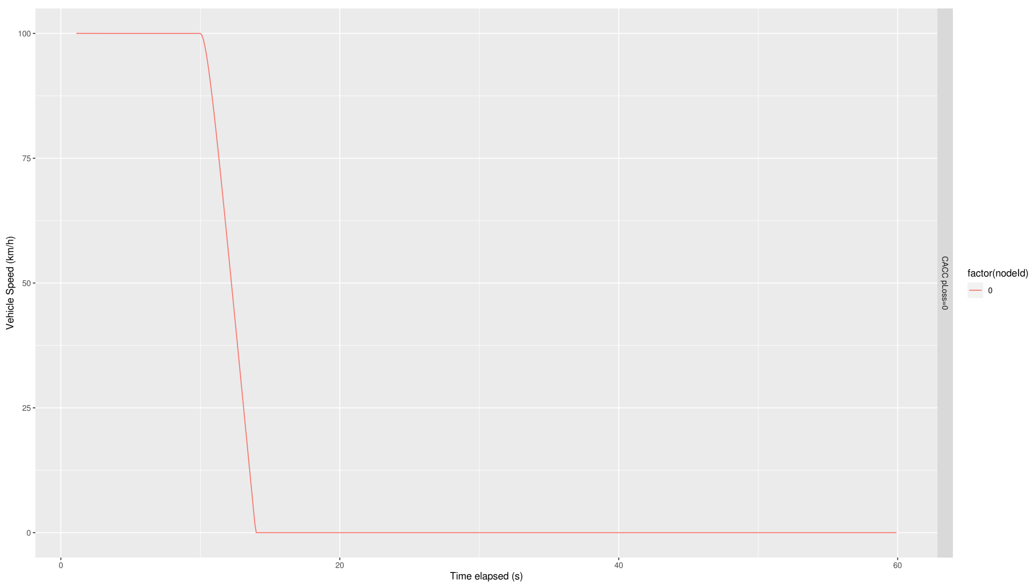

Figure 4.2.1: Full-braking disturbance demonstration: vehicle speed w.r.t. time

This scenario involves a platoon of vehicles which has size 8. The platoon leader and all of its followers begin the simulation with a given speed (100 km/h) and after 5 seconds, the platoon leader applies the brakes and causes the vehicle to decelerate quickly. The followers should come to a full stop behind the leader.

There is a characteristic of this scenario that is observed for some controllers, whereby after the lead vehicle has come to a full stop, some of the followers will creep closer to their predecessor. This is caused by the controller's implementation of a constant-time headway. When a vehicle's predecessor is completely stopped, the controller still tries to decrease the inter-vehicle distance. The effect that this has on the analysis of results is that the small changes after the initial stop can readily be ignored. This would not be the case if the scenario includes a vehicle crashing, because as soon as vehicles crash into one another, the simulation cannot continue. This explains, why for some of the graphs, the results seem to just stop at a certain point (this will be noted in the explanation of the results).

4.3 Join Manoeuvre Scenario with Sinusoidal Disturbance

For this scenario, the disturbance could be introduced at any time, but in order to isolate the time that the controller has to handle the joining vehicle as well as all of its followers, the disturbance is only added when the joining vehicle is close. The disturbance will cause the platoon leader to accelerate and decelerate sinusoidally in the same fashion as the scenario without a merging manoeuvre. The disturbance can be stopped at any time (i.e. it does not have to last for the entire simulation to create meaningful results). However, once the controller gives the joining vehicle the same spacing policy as the other followers and the desired inter-vehicle spacing is the same as the other platoon members, the behaviour of the platoon should be the same as the scenario without a merging manoeuvre. Therefore, the disturbance can be stopped shortly after the joining vehicle has become a platoon follower.

This scenario uses a platoon of size 8 with another vehicle that joins at the back of the platoon to make a platoon of size 9. The platoon leader and its initial 7 following vehicles begin the simulation in a fully formed platoon going at 100 km/h. After 6 seconds, a new vehicle spawns roughly 50 metres from the end of the platoon travelling at 100 km/h. After a further 10 seconds, the new vehicle begins it merging manoeuvre, whereby it requests permission from the platoon leader to join the platoon. After the platoon leader acknowledges the joiner's request and gives permission to begin the manoeuvre, the joining vehicle moves into the same lane as the rest of the platoon and begins to accelerate towards the back of the platoon. After the joining vehicle gets close to the platoon it slows down to match the speed of the platoon, in order to facilitate a safe joining manoeuvre. It is at this point, when the joining vehicle is very close to the platoon that the disturbance will be introduced into the platoon.

4.4 Join Manoeuvre Scenario with Full Braking Disturbance

This scenario uses the same formation geometry as the scenario above i.e., a platoon of size 8 is travelling along a highway, and another vehicle joins the platoon from the back. In the previous scenario, however, the disturbance that was introduced to test the string stability of the controller existed for an extended period of time, and then it could be turned off. This scenario involves having the leader brake suddenly and come to a full halt. This scenario may show different results, as the increased rate of deceleration may cause the controller to expose unsafe behaviour.

The reason that the results may vary depending on the distance between the platoon and the joiner when the disturbance begins is because of the differing communication conditions for different inter-vehicle distances. If the joining vehicle is close to its predecessor, the controller must send urgent updates to the joiner to tell it to slow down. These urgent updates must be received by the joiner, i.e. any significant packet loss experienced by those urgent updates could have catastrophic effects. Possibly even resulting in a crash. But if the joiner is far away, the controller's updates are less likely to cause a crash if they are lost across an unreliable network. However, the further away the joiner is from the leader and/or its predecessor (depending on the IFT), the more likely it is to experience the adverse effects of the unreliable network.

4.5 Controller Communication Error-handling

The goal of many of these simulations is to investigate the behaviour of controllers when non-ideal communication conditions are present. When doing this investigation, it is vital to understand the behaviour of the controllers when a platooning message is not received. In a real-world scenario, the vehicle may be required to fall back to a mode of operation that does not require any V2V communications (e.g. ACC) in order to ensure the safety of the vehicle and the platoon as a whole. It is important to ensure that the behaviour of a given platoon controller is isolated to the expected behaviour of that controller (e.g. a connected controller should be unable to function normally in 100% packet loss rate). Therefore the behaviour of Plexe's controllers in the presence of 100% packet loss (i.e., no communications) will be investigated. Because of this, some simple control simulations will be run to investigate the behaviour of Plexe's connected controllers when it does not receive a message.

The first scenario used to investigate the behaviour of the controllers involves using the full braking scenario (Section 4.2) and the packet loss rate of the V2V network is set at 100% loss. This will test the behaviour of a controller as no messages will travel from vehicle to vehicle to warn them of the braking leading car. If the CACC controller used by the simulator manages to stop behind the leading car rather than crashing into it, then there is sufficient proof that the CACC controller falls back to a controller which does not use V2V communications. However, if the platoon immediately crashes into the back of the leading vehicle (without slowing down), then there is sufficient evidence that when the CACC controller does not receive any communications, it has no other mode of operation (i.e. the vehicle that is being controlled will do whatever it was last told to do *ad infinitum*).

The simulator's ACC controller will be used as a baseline comparison to the CACC controller, as the ACC controller never requires V2V communications to control the behaviour of the platoon (unless a new vehicle attempts to join the platoon).

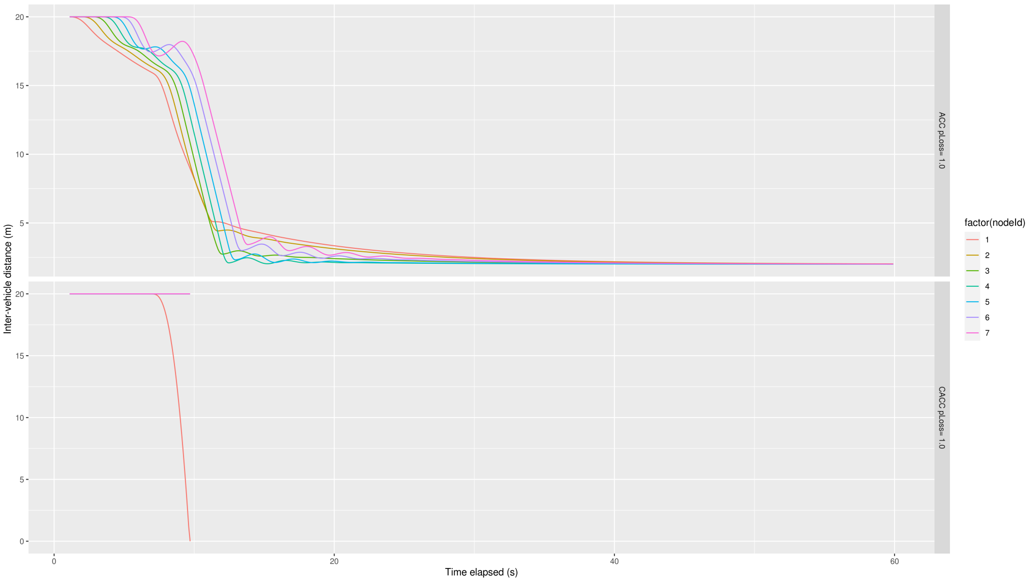

Figure 4.5.1: Packet loss rate 100% control experiment: inter-vehicle distance w.r.t. time. a) ACC b) CACC

Figure 4.5.1 shows the inter-vehicle distance of the vehicles in the platoon with respect to time. The ACC baseline shows the expected behaviour i.e., that a lack of all V2V communications did not affect the controller in its ability to stop the followers from hitting the leader. The CACC model ends early as a result of a crash between the platoon leader and its first follower. This implies that the CACC controller has no fall-back for when communications are unavailable. This observation can be confirmed if the acceleration which was calculated by the controller for the vehicle which crashed showed no change until the crash occurred. If the controller acceleration showed any change before the crash, then there must be a fall-back of some description, as the controller would have no other way of informing the first follower to brake other than V2V messages which have been eliminated.

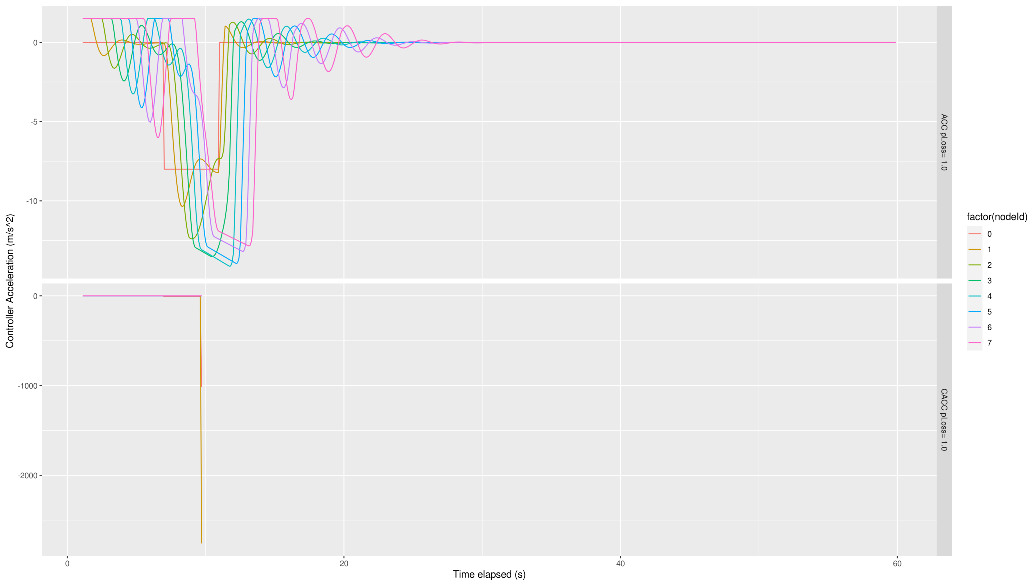

Figure 4.5.2: Packet loss rate 100% control experiment: controller acceleration w.r.t. time. a) ACC b) CACC

Figure 4.5.2 shows that the controller did not calculate a change in the acceleration of the vehicle that crashed. The sharp slope downwards in the graph was only created at the point of impact. Therefore, it can be safely concluded that when no communications are present, the CACC controller that is used by this simulation environment has no fall-back controller. This is an important result to understand, as it means that any behaviour that is observed over the course of the experiments are purely a result of the controller being tested and there is no other controller influencing the operation of the platoon.

4.6 Experimental Procedure

The experimental procedure that will be followed for the collection of the results during the simulations is broken down into several repeated steps. Firstly, the scenarios which do not involve merging manoeuvres will run. For each of these scenarios there are 3 non-ideal communication conditions (packet loss rate, background noise, and transmission power) which were explained in Section 3.1.1 and simple non-ideal actuation conditions explained in Section 3.2. Each of the three platooning controllers - ACC, PATH CACC, and Ploeg CACC - will be investigated using each of the non-ideal conditions. The justification behind the selection of each of these controllers is outlined in the preamble for Section 3.3. The investigation of each non-ideal condition will begin by varying the condition over a wide range in order to locate the range of values that may cause the non-ideal conditions to have an effect on the platooning controllers. Observing the results of the wide range of conditions will allow for a smaller range of values for the conditions to be investigated. The boundary after which each non-ideal condition may cause unsafe behaviour in each platooning controller can be located by interpreting the results of the smaller range of values for the non-ideal condition being investigated.

This complicated flow has been concisely explained using pseudocode in Algorithm 1.

4.7 Summary

In conclusion, this chapter has detailed the 4 scenarios that will be used to test the string stability of a platoon (sinusoidal disturbance for non-merging and merging, and full-braking disturbance for non-merging and merging), why those results are valid for comparison, and how the results will be collected using the techniques already outlined.

Algorithm 1 Experimental Method

```
for scenario of: non-merging scenario, merging scenario do
    for condition of: packet loss rate, background noise, transmission power, actuation delay do
        for controller of: ACC, PATH CACC, Ploeg CACC do
            Predict possible outcomes of wide range
            for wide range of values for condition do
                Initialise simulation using scenario, controller, value of condition
                Run simulation
                Collect simulation results
                Convert results into R data format
            end for
            Run plotting program to create graph of collected results
            Compare predictions to results
            Deduce small range from wide range of results
            for small range of values for condition do
                Initialise simulation using scenario, controller, value of condition
                Run simulation
                Collect simulation results
                Convert results into R data format
            end for
            Run plotting program to create graph of collected results
            Discuss results at a low level
        end for
    end for
end for
```

Chapter 5

Implementation

This chapter will explain some of the issues that were encountered over the course of this project and how they were overcome. Section 5.1 will discuss some of the difficulties encountered with setting up the simulation platforms that were required to run experiments. Section 5.2 will explain the matter of how some communication conditions had to be discarded from consideration in this project. Section 5.3 will lay out the difficulties associated with becoming familiar with all of the relevant nomenclature of a field of research that someone has not encountered before. Section 5.4 will explain the issues associated with interpreting the results of the experiments.

5.1 Simulation Platforms

In order to run the simulations for this project integrating the behaviour of several state-of-the-art simulators was necessary. The platforms that were needed were a basic traffic simulator (SUMO), a communication network simulator (OMNeT++), a platform to allow the traffic and networks to operate with one another (Veins), and a platform to allow platooning operations to be carried out (Plexe). As each of these simulation platforms are mostly used for specialised research, they are complex and have many available features, not all of which are well documented. The main simulation platform used in this project, Plexe, operates by using different features from each of the other platforms and combining them in a complex manner. The field of platooning research is relatively small when compared to all of the fields that make use of the other platforms, as such, Plexe is somewhat underdeveloped simply because it is not widely used. This combination of the fact that it is a highly specialised platform, and it is not used by many researchers means that it is poorly documented. This poor documentation is the cause of the following issues.

The operation of OMNeT++, Veins, and SUMO are well established for Windows operating systems. I was able to run every example scenario provided with each of those platforms on my own Windows machine. Installing Plexe was not so easy. An error in communication between the SUMO app and the Veins simulation caused all examples to crash immediately. I spent several weeks investigating this problem. I still believe that the problem may have been solvable, and probably related to different versions of Veins existing on the same machine, or different versions of Python existing on the same machine. The error message that was displayed said that "readChar()" could not be defined. "ReadChar()" probably referred to a Python package utilised by Veins to communicate with the SUMO app over local ports. I did not figure out this issue.

A solution to this problem was presented to me: install a virtual machine which uses the same operating system that someone who has experience with Plexe is using. I attempted to follow this direction, but unfortunately due to a break down in communications, I installed a different version of Ubuntu (20.04) than had been recommended to me (Ubuntu 16.04). A similar problem persisted with the new virtual machine; OMNeT++, SUMO, and Veins all worked fine, but Plexe caused issues. I found a source online that purported to have tested Plexe using Ubuntu versions 16.04, 18.04, and 20.04. Each of these versions worked for that source, so I continued attempting to solve the problem by changing the project dependencies to include more packages. After much frustration, I finally decided to delete the Ubuntu 20.04 virtual machine that I had, and start from scratch with Ubuntu 16.04. After I installed a virtual machine that ran on Ubuntu 16.04, I had Plexe working in a matter of minutes. This was a satisfying resolution.

Plexe has documentation [40], but the documentation is not comprehensive, or even close to it. This means that many fringe issues require seeking assistance in other places than the official documentation. It was a frustratingly common occurrence that Googling an issue would result in no search results. Not having exhaustive documentation allowed for many minor delays in the development of the simulations while I attempted to solve the issues myself. The poor documentation was mitigated by reading the comments in the source code, and asking for help from people more experienced with Plexe than me.

5.2 Communication Conditions

For this project I investigated the effects packet loss rate, transmission power, and background noise on several platoon controllers. Each of these conditions causes a slightly different effect on the platoon, as discussed in Section 3.1. One of the challenges of this project was finding variables that I could control in the simulation environment that have an effect on communications and that could conceivably happen. A control variable that I investigated for a while was beaconing interval. This was an attempt to simulate packet loss and packet delay simultaneously, but unfortunately the operation of beaconing interval does not accurately represent how message delay would occur. Beaconing interval is also a variable for the communication protocol that on any network that makes use of beaconing. So if it can be varied, one might think that it would be a valid variable to investigate. However, the protocol that is used for all state-of-the-art V2V communications, DSRC, and all protocols that make use of this have a beaconing interval of exactly 10 Hz [41]. This means that varying the beaconing interval would not be a worthwhile area of study for the time being. If new protocols arise that make use of different beaconing intervals, then it would be worth investigating the effect that beaconing interval may have on communication quality. There is emerging research that a beaconing interval of 10 Hz could lead to network congestion in high traffic-density areas [42]. A new form of beaconing called *jerk beaconing* has been proposed by Segata *et. al* which suggests beaconing that only operates when necessary rather than constant polling.

5.3 Jargon

One of the most difficult aspects of this project was learning to become comfortable using the jargon in the Connected Autonomous Vehicle area of research. There are many terms that have specific meanings in CAV research that may mean completely different things in other contexts. One example of this are the terms homogeneous and heterogeneous. These words are often meant to mean things that are well mixed and consistent (homogeneous) and things that are not well mixed, and consist of different parts (heterogeneous). Specifically when it comes to CAVs heterogeneous can refer to two definitions. The first definition of heterogeneous traffic is traffic that have different modes of control. An example of heterogeneous traffic in this sense would be traffic where there are some manually-driven vehicles and some (connected) autonomous vehicles. The second definition of heterogeneous traffic is traffic that have differing dynamics. Differing dynamics means different vehicle movement characteristics e.g., top speed, acceleration, braking. Homogeneous means the opposite of both of these respective definitions in the same way. These two phrases are just some of the examples of jargon that must be learned well in order to be able to understand that state-of-the-art literature.

This kind of jargon can lead to ambiguity when understanding research. Another example of when a word has multiple meanings that are similar in some ways but vitally different in others is the word "controller". Controller can simply be an implementation of a spacing policy that takes beaconing messages into account. But a controller can also be the entire control infrastructure for a platoon. This definition not only includes the implementation of the spacing policy, but also the implementation of the information flow topology.

5.4 Interpreting Results

There was no way to collect the ideal inter-vehicle distance from the simulator during the simulations. As such, the usual method for calculating the presence of string stability, using inter-vehicle spacing error (the absolute value of desired spacing minus actual spacing) was not possible over the course of this project. This meant that a simple script for calculating the presence of string stability was not possible. In place of using spacing error, a graphical approach was taken. This graphical approach involved representing the inter-vehicle spacing of each predecessor-vehicle pair on a plot, and manually examining the changes in spacing over the course of the simulation. Lacking an objective method of calculating string stability means that despite my best efforts to assess the safety conditions of any given simulated scenario, some subjectivity is associated with the interpretation of the results. In order to avoid this, any point where string unstabel behaviour, or unsafe behaviour was observed, then valid reasoning was given as to why the behaviour was unsafe.

5.5 Summary

This chapter laid out many of the issues that were encountered over the course of the project, and how they were avoided or mitigated.

Chapter 6

Results

This chapter will display and demonstrate the results of the experiments conducted over the course of this project. Section 6.2 will explain how the result plots are structured, and how to read the plots. Section 6.2 will detail the results for each non-ideal conditions in the non-merging scenarios. Section 6.3 will outline the results collected for each non-ideal condition for each of the merging scenarios.

6.1 Reading result plots

The figures used to represent the results collected from the simulations can be intimidating to look at for the first time. However, they all share common features that are discussed throughout this chapter.

Figure 6.1.1 demonstrates a typical looking figure that will be seen over the course of this chapter, but the key features have been numbered. The following list explains what each of the numbers represent:

1 Denotes the y-axis label. This will often be representing inter-vehicle distance in metres.

2 Denotes the x-axis label. This will always represent the amount of simulation time elapsed in seconds.

3 Denotes the first oscillation. The first oscillation will be referred to in this chapter. For a sinusoidal disturbance, this means the trough of the first sinusoidal acceleration that the leader undergoes.

4 Represents the controller and condition value pair. The name of the controller being used, the non-ideal condition being tested, its value, and units all appear here. These are often difficult to read, so the value pairs are included in the caption of the figure.

5 The legend. The different colours in this box represent the position of the vehicles in the platoon, and the colours of the lines that represent their behaviour on the plot. It should be noted that the leader (ID 0) does not appear on any of the inter-vehicle distance plots because it does not have a predecessor.

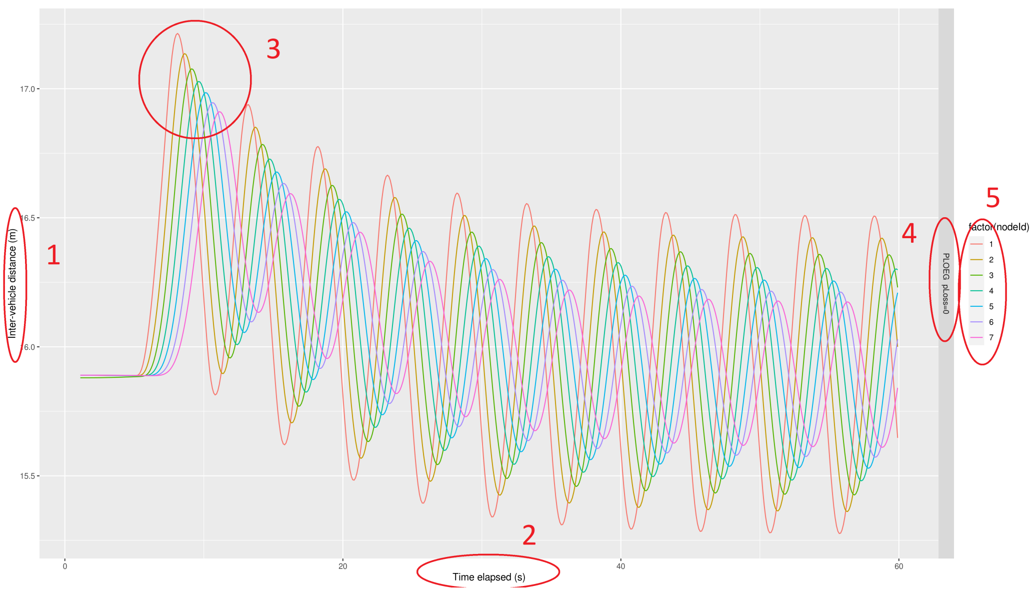

Figure 6.1.1: An example plot whose key features are annotated

6.2 Scenarios Without Merging Manoeuvres

The next section concerns non-ideal communication conditions and non-ideal actuation conditions for scenarios that without merging manoeuvres. For each of the non-ideal communication simulations, a boundary value should be found, after which the string stability or safety of the platoon is compromised as a result of the poor communications.

6.2.1 Packet Loss Rate

There are several ways to simulate non-ideal communication conditions over a network, but one of the most simple and effective is to vary the packet loss rate. As with almost all digital communications (that are not streams of information), data must be divided into packets when being sent over a network. An unreliable network can arise for many different reasons, but the end effect will almost always appear as some form of packet loss, or maybe packet delay. This is why varying the packet loss rate simulates non-ideal communication conditions so well, it simulates many different kinds of unreliable channels at once.

V2V communications for a platoon of vehicles can be vital for the safety of all vehicles in the platoon. As shown by an earlier demonstration, if the CACC controller being used in these simulations does not receive communications, it will simply carry on doing what it was last told to do. This means that any adverse effects on performance observed during this experiment could be very harmful if the same controller were used in real life without a fall-back.

Control Experiment As a control for this experiment, the scenario will be implemented using a controller which does not utilise V2V communications, and therefore, should not behave differently for each of the different simulated values for packet loss.

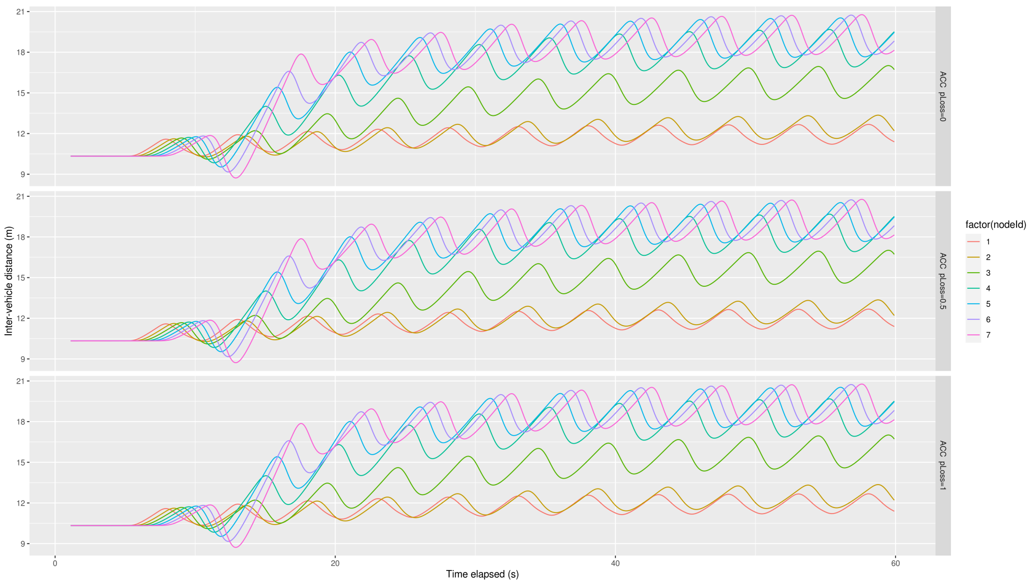

Figure 6.2.1: Varying packet loss rate for ACC controller: inter-vehicle distance w.r.t. time. a) pLoss=0 b) pLoss=0.5 c) pLoss=1

As demonstrated by Figure 6.2.1, the ACC controller neither improves nor dis-improves as the amount of packet loss experienced by the platoon's V2V network varies. This shows that the ACC controller can be thought of as a valid fall-back solution if the network becomes unreliable. This figure also demonstrates that inter-vehicle distance grows as the simulation time increases, and the inter-vehicle distance for vehicles far back in the platoon experience greater amplification than vehicles close to the leader. These are sufficient conditions to conclude that the ACC controller is not string stable. Having a fall-back controller for situations where the network fails is vital, but using that fall-back for other situations is undesirable if the fall-back is not string stable. This stems from the fact that the ACC controller must use a constant time headway as there is no communication to facilitate varying the headway for all vehicles as would be needed to implement a variable time headway.

CACC Controller This scenario can be used to evaluate the response of the CACC controller to variable packet loss. A broad range of values for packet loss will be used initially. The range of packet loss being investigated will narrow as the results are interpreted. The ideal value for packet loss rate (zero) is displayed in the wider range of results.

Figure A.1.1 shows that for ideal communications, the platoon behaves predictably in an organised fashion. At 15% packet loss the platoon begins to behave erratically, but the plot shows that the controller is still able to behave in a string stable fashion because the values for inter-vehicle distance trend downwards along the platoon. There are some exceptions, namely the 4th follower in the platoon begins to separate from the vehicles in front of it from roughly 45 seconds in the simulation. At 30% packet loss the platoon moves quite erratically. The plot shows that the first follower comes very close to colliding into the platoon leader. The disturbances do not appear to amplify as they propagate down the platoon, but the behaviour is clearly not safe because of the near-miss between the leader and the first follower. At 45% packet loss, the simulation ended early as a result of the first follower crashing into the back of the platoon leader. This behaviour is extremely unsafe. It is worth noting that although the controller was not able to prevent a crash in this situation, the platoon still appears to most;y behave in a string stable manner. The inter-vehicle distance does keep trending downwards (attenuating) along the platoon, even in the most erratic cases. This demonstrates that it is not only string stability that can play a role in the safety of a platoon.

From this, it can be deduced that if the packet loss rate of a system using the CACC controller tested here exceeds 30% it can be thought of as an unsafe system. If the packet loss rate exceeds 45% then the system is extremely unsafe, and a communication-less fall-back should be implemented. However, the behaviour at 15% is not unsafe enough to deem it completely necessary to use the fall-back system. As such, further investigation will be done on a range of values from 15% to 30%.

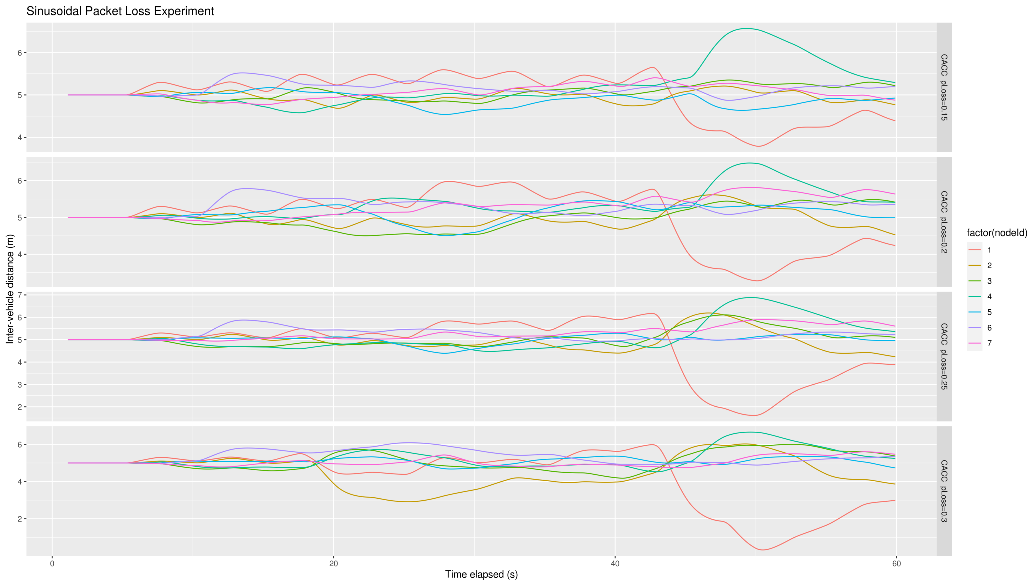

Figure 6.2.2: Varying packet loss rate for CACC controller over a small range: inter-vehicle distance w.r.t. time. a) pLoss=0.15 b) pLoss=0.2 c) pLoss=0.25 d) pLoss=0.3

From Figure 6.2.2 it is clear to see that the changes in behaviour between each packet loss rate is quite small, but the biggest contrast between each scenario is the minimum distance between the platoon leader and the first follower. For a 30% packet loss rate, the first follower gets within 50 centimetres of the platoon leader, but for 20% and 25% the first follower only gets within 3 metres and 2.5 metres (respectively) of the platoon leader. This shows that the system, when experiencing 25% packet loss is much more safe than the system when it is experiencing 30% packet loss. It is worth noting again that the system never shows strong evidence of a consistent lack of string stability, but it simply behaves erratically and in an unsafe manner. In conclusion, if the network that this system uses ever experiences a packet loss rate greater than 25%, then a communication-less fall-back should be used.

Ploeg Controller Here, the Ploeg CACC controller will be tested against a wide range of values for packet loss rate to determine when the controller begins to act in an unsafe manner.

Figure A.1.2 demonstrates the performance of the Ploeg CACC controller across a wide range of values for packet loss rate. For no packet loss (0% packet loss), the controller demonstrates very clearly that it can behave in a string stable fashion for ideal conditions. It is clear to see that the inter-vehicle distance of each vehicle decreases along the platoon for every oscillation of the disturbance. It is a very good plot to visually demonstrate the definition of string stability. At 25% packet loss rate, the controller demonstrates very similar string stable behaviour when compared to the ideal condition, however, some erratic movement is also present. This is not enough to consider to the controller completely unsafe, but it must be noted. At 50% packet loss, the controller begins to behave in a much less string stable manner. This can be observed at the peak of the first oscillation in its plot. Each of the vehicles in the platoon's inter-vehicle distance reach almost exactly the same amplitude. This demonstrates that the disturbance is no longer strictly being attenuated. Instead, it seems to even out. This may be considered the edge of where string stability is possible for the controller, but further investigation is required. At 75%, the controller behaves both erratically and in a string unstable manner. For such a high rate of packet loss, it is expected that the operation begins to break down at this point. The operation of the platoon at 100% packet loss is also shown here. This is to demonstrate (*again*) that without a fall-back controller, failures in the network can cause a complete breakdown in the operation of the platoon.

Figure 6.2.3: Varying packet loss rate for Ploeg controller over a small range: inter-vehicle distance w.r.t. time. a) pLoss=0.45 b) pLoss=0.5 c) pLoss=0.55 d) pLoss=0.6

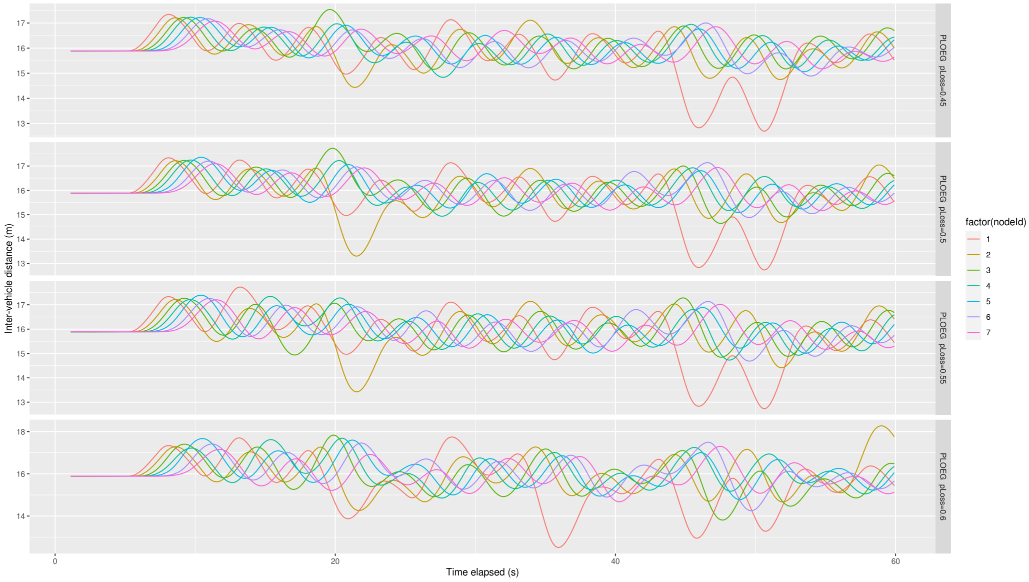

Figure 6.2.3 demonstrates the boundary of where the Ploeg CACC controller becomes string unstable. It can easily be observed that for 45% packet loss, at the peak of the first oscillation in this plot, the first follower's inter-vehicle distance is the greatest. The distances attenuate along the platoon, but only slightly. This shows that the platoon behaves in a string stable fashion for this amount of packet loss, but it will not remain string stable much longer. The performance of the controller at 50% was discussed above. At 55% the first oscillation shows that the distances neither attenuate nor amplify along the platoon, they stay constant. This behaviour is not repeated for the later oscillations. For some of the later oscillations, the platoon does appear to behave in a string stable fashion, whereby the inter-vehicle distances attenuate (e.g., the oscillation @ ˜12 seconds in), but for other oscillations, the distances can greatly amplify (e.g., @ ˜42 seconds). At 60% packet loss, the system begins to behave quite erratically. But because the margin for safety applied in this implementation of the Ploeg controller is much higher than the margin of safety implemented in the previous CACC

controller, the erratic behaviours do not result in a crash. Increasing the margin of safety decreases the throughput of the controller, as the vehicles travel farther apart from one another but it increases the robustness of the controller with regards to network losses.

In conclusion, this controller is able to reliably implement string stable behaviour in its platoon until a packet loss rate of roughly 50%. Any packet loss rate greater than 50% is likely cause string unstable behaviour; erratic behaviour can be expected as well. However, at 50% packet loss rate, the erratic behaviour which is observed is unlikely to raise any major safety concerns.

6.2.2 Background Noise

A factor which may have to be taken into account for real-world scenarios is the background noise of the network environment. If a wireless message is being sent across a noisy environment, the data in the message can easily become garbled or possibly even attenuated completely before the receiver reads the message. This could be a real-world cause of packet loss. It is important to note that in order for a message to be lost over the distances concerned for this scenario (propagation distances in the order of tens of metres), the background noise may have to be significant. Investigating the amount of background noise needed to cause packet loss to occur may yield results that show that the background noise needed is much higher than is possible to come across in real-world scenarios. However, there is the possibility that the background noise could be intentionally raised in an attempt to jam platooning signals as part of a network attack on the platoon.

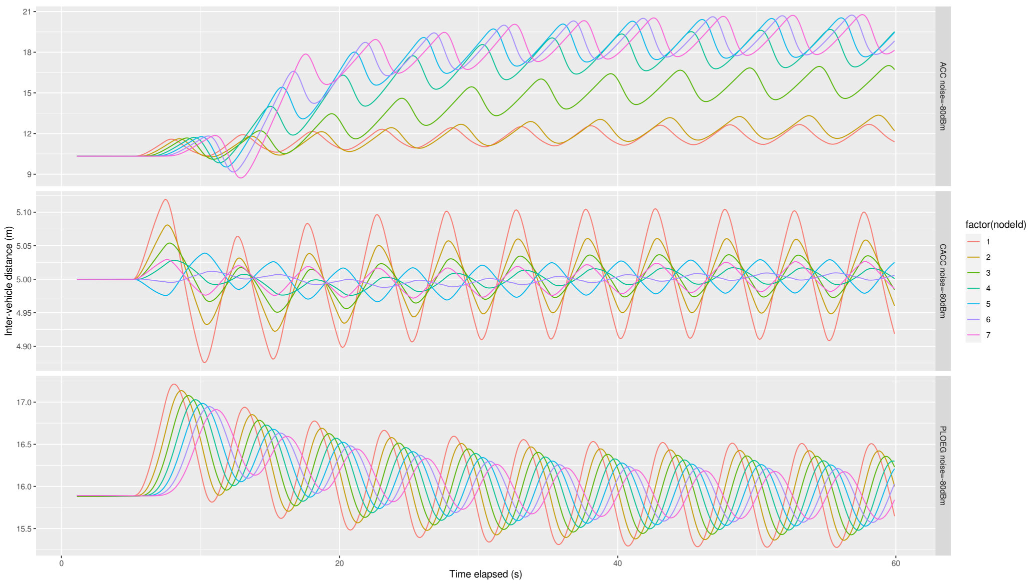

Figure 6.2.4: Control background noise of -95 dBm: inter-vehicle distance w.r.t. time. a) ACC b) PATH c) Ploeg

Figure 6.2.4 demonstrates the behaviour of each of the chosen controllers for the ideal condition when the background noise power is -95 dBm.

ACC - Control Experiment For the control of this experiment, the scenario is implemented using the ACC controller, because this controller should not be affected by poor communications.

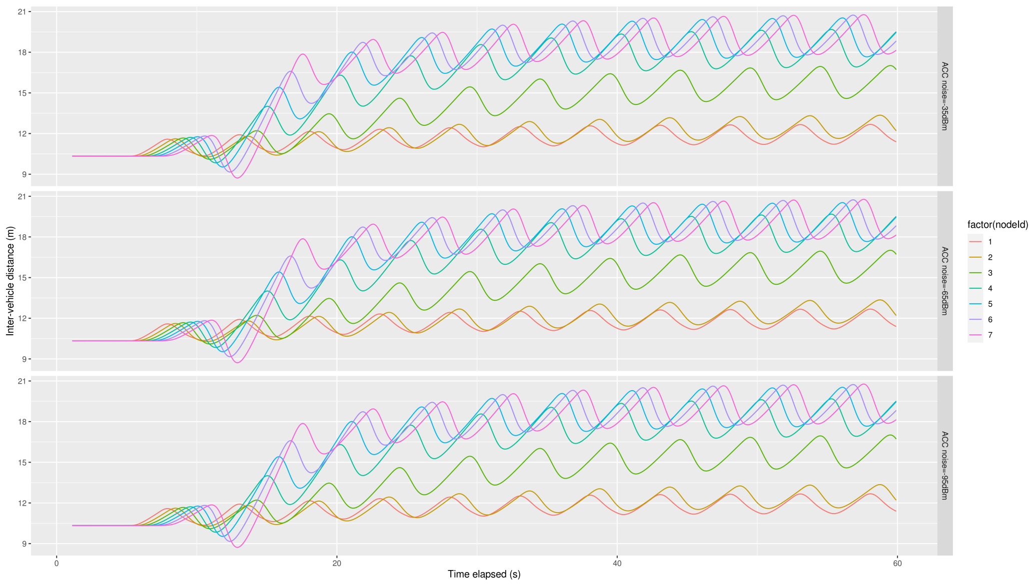

Figure 6.2.5: Varying background noise: inter-vehicle distance w.r.t. time. a) noise=-35 dBm b) noise=-65 dBm c) noise=-95 dBm

As expected, Figure 6.2.5 shows that the background noise of the scenario does not affect the performance of the ACC controller.

CACC controller The CACC controller has a small margin of safety, so it uses a relatively small inter-vehicle distance for the spacing policy. The effect of background noise on a transmission is proportional to the distance the transmission travels across the network. This suggests that the controller will stay string stable for most scenarios, and it will likely behave dangerously erratic before it behaves in a string unstable way.

Figure A.1.3 shows that there is no change in the operation of the platoon using the CACC controller for values of background noise power between -95 dBm and -80 dBm. This is likely because background noise power is not a linear scale. The difference between an amount of background noise that has no impact on communications and background noise that has a large impact on communications could be tiny. This is could explain why for the plot of background noise power at -65 dBm suddenly the platoon is entirely unable to function. The two intervals between the plots are identical, but the changes in result are massive. This suggests that finding the boundary of background noise that causes the platoon to become string unstable may be difficult.

Figure 6.2.6: Varying background noise for the CACC controller: inter-vehicle distance w.r.t. time. a) noise=-69.5 dBm b) noise=-70 dBm c) noise=-71 dBm

Figure 6.2.6 confirms the prediction that finding the range over which the platoon begins to behave unreliably is difficult. The plot which demonstrates the smallest amount of background noise power is the plot of -71 dBm. This plot demonstrates that most of the platoon behaves completely normally for the entire duration of the simulation, but the final vehicle in the platoon strayed a small bit farther from the rest of the platoon at roughly 45 seconds into the simulation. This one case where the vehicle strays too far from the platoon to reliably receive platooning communications most likely occurred because of a lost message (or messages) due to the background noise. The plot which demonstrates the behaviour of the platoon at -70 dBm shows very similar features to the scenario with slightly less noise. This is because, all of the platoon behaves normally, apart from the final vehicle in the platoon which splits from the rest of the platoon (because a message was lost to the background noise), and then can never recover to get back to the group. The rest of the platoon behaves completely normally. For the next scenario, the amount of background noise power was only reduced to -69.5 dBm. This is a very small adjustment to the amount of noise in the network, but the effect it has on the performance of the platoon is massive. The final vehicle in the platoon splits off from the rest of the platoon earlier than in any of the other scenarios. The vehicle which splits off (the "splitter") and the platoon leader both still believe that the splitter is part of the platoon, but in fact it is not receiving correct platoon information due to packet loss caused by the background noise. This means that the leader does not adjust the platooning messages which it is sending to the splitter. The splitter continues to attempt to implement the acceleration that is given to it by the controller. This results in the splitter performing whichever accelerations 'get through' to the vehicle, and nothing else. The splitter gets within 2 metres of its preceding vehicle which is extremely unsafe behaviour.

These plots demonstrate another example whereby the safety of the platoon is not being called into question by a lack of string stability, but simply by erratic movements. There was no evidence that there exists a point where the simulation worked without crashing **and** the platoon was behaving in a string unstable manner. Another platooning characteristic which comes into play to explain the result that is being observed here is the IFT that is being used by the Plexe CACC controller. This controller uses the leader-predecessor following IFT. This suggests that the final vehicle in the platoon was the only vehicle struggling from the communication conditions because it was simply too far from the platoon leader to reliably receive the platooning information that it needed in order to function normally. This exposes a flaw in the way that the information flow was implemented for this controller. The controller should be able to identify that messages from the leader to the vehicle are being lost, and therefore the vehicle's predecessor should be used to transmit platooning commands to the vehicle which is experiencing the message loss.

Ploeg Controller The Ploeg CACC controller uses a mid-range sized inter-vehicle distance for its spacing policy, it uses a predecessor-following IFT which means that the issue observed with the previous CACC controller (whereby the communications from the leader were lost) should not be repeated. This suggests that the controller should be quite sturdy across a wide range of values for background noise power. This controller should present itself as being able to remain string stable for a long time. It should not appear to behave erratically until it has already appeared to behave in a string unstable manner.

Figure A.1.4 demonstrates that the Ploeg CACC controller is more robust than the PATH CACC controller. It offers no evidence as to what the boundary of operation for this controller is. It shows that the controller performs exactly the same for all values of background noise power which were tested. Its performance is also noteworthy for its complete string stability throughout the simulation.

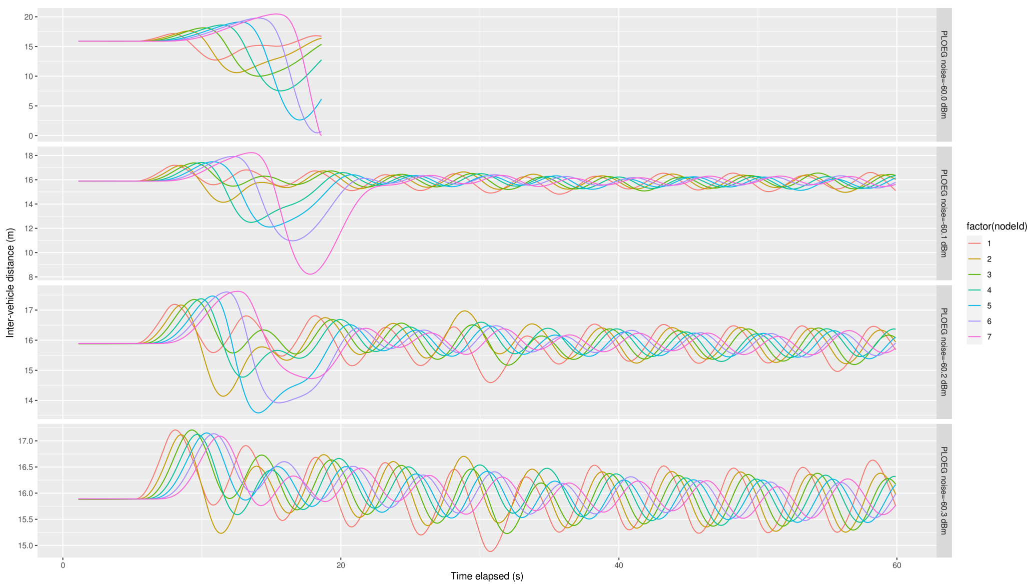

Figure 6.2.7: Varying background noise for the Ploeg controller: inter-vehicle distance w.r.t. time. a) noise=-60 dBm b) noise=-60.1 dBm c) noise=-60.2 dBm d) noise=-60.3 dBm

Figure 6.2.7 shows similar behaviour to the PATH CACC controller with respect to the fact that very small changes in the background noise power can have massive effects on the operation of the controller. The range tested in this figure is less than 0.5 dBm. The huge difference in behaviours across these small changes in the noise power shows how dangerous the background noise can be if it is never detected. The plot which demonstrates the operation of the platoon at -60 dBm finishes early because the simulation resulted in a crash. This crash was between the final vehicle in the platoon and its predecessor. Before the crash occurred there was only one oscillation, and it demonstrated good string instability. It can easily be seen from the first oscillation that the inter-vehicle distance amplified along the platoon. The trough following the crest which demonstrated string instability resulted in a crash because of further string instability. The disturbance was consistently amplified. At -60.1 dBm of background noise power, the platoon showed very similar behaviour to the first plot, but it narrowly avoided the crash between the final vehicle in the platoon and its predecessor. After it was able to avoid the crash, it settled back to a state where it didn't experience that kind of behaviour again. When the platoon settled into the state, it behaved in a somewhat string stable manner. It cannot be considered completely string stable because there were points in the simulation (e.g., @ 55 seconds) where vehicles far back in the platoon were experiencing oscillation amplitudes greater than the first follower. However, on average, the inter vehicle distance trended slightly downward along the platoon for each oscillation. Those two reasons are why, at -60.1 dBm, it cannot be considered completely string stable but it is somewhat string stable. At -60.2 dBm, the first oscillation showed string unstable behaviour, and up until roughly 35 seconds in the simulation, the platoon behaved very erratically. But after 35 seconds, the platoon behaved in a string stable manner. The fact that this scenario demonstrated both string stability and string instability shows that this is roughly the boundary of where the platoon can behave with string stable characteristics when exposed to high amounts of background noise. The plot which describes the platoon at -60.3 dBm shows very similar behaviour to the previous simulation, except for one aspect. The two simulations differ in their performance at the first oscillation, the previous simulation showed amplification in the inter-vehicle distances, but this simulation shows attenuation at the first oscillation. It behaves somewhat erratically, but mostly string stable until roughly 35 seconds into the simulation, after which it behaves entirely string stable for the remainder of the simulation.

6.2.3 Transmission Power

In real-world situations, the quality of hardware can degrade over time, or even be intentionally tampered with. One possible manifestation of this degradation in quality could be a large decrease in the amount of power that the communication unit on board the vehicle is able to produce. A drop in transmission power could have a large effect on a CAV's ability to remain part of a platoon.

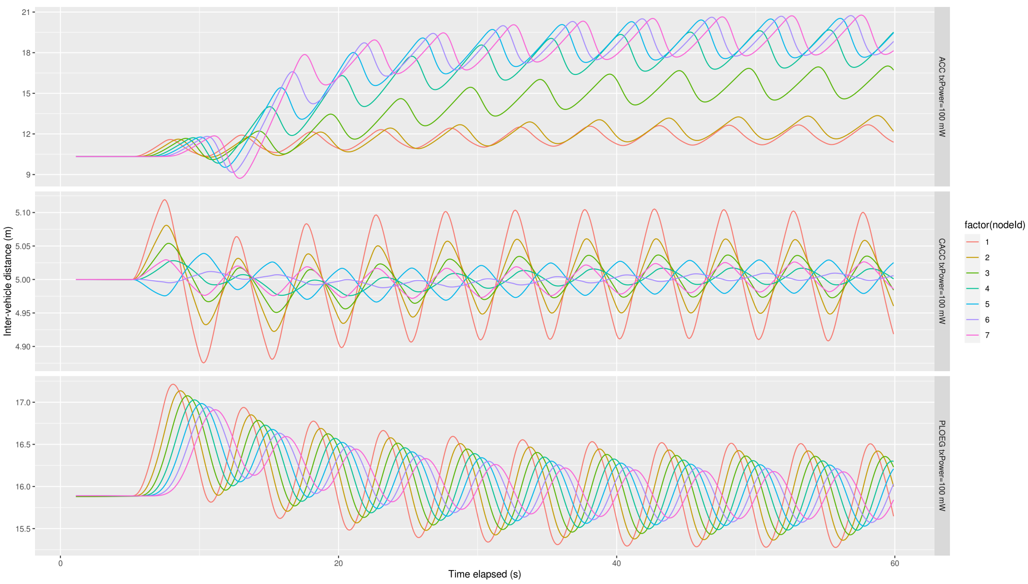

Figure 6.2.8: Varying transmission power: inter-vehicle distance w.r.t. time. a) ACC b) CACC c) Ploeg

Figure 6.2.8 demonstrates the operation of each of the controllers in the ideal conditions for transmission power (100mW).

ACC - Control Experiment For the control of this experiment, the scenario is implemented using the ACC controller for the same reason as the other communication conditions: it does not use V2V communications.

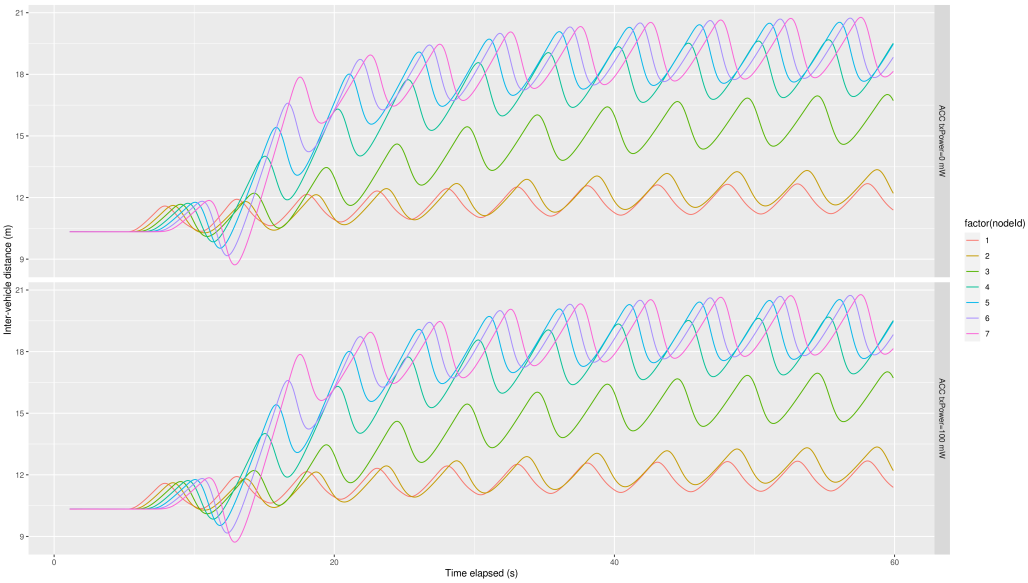

Figure 6.2.9: Varying transmission power: inter-vehicle distance w.r.t. time. a) txPower=0 mW b) txPower=100 mW

As expected, Figure 6.2.9 shows that the transmission power of the scenario does not affect the performance of the ACC controller.

CACC Controller The fact that PATH's CACC controller uses a spacing policy that optimises road throughput means that the vehicles will all be very close to one another. This is very beneficial for the case where the transmission power of the on-board communications unit is producing much less power for transmissions than expected. This controller also implements a leader-predecessor following information flow topology. This flow topology has already been observed to behave in a string stable fashion for most of the platoon, but because platooning information from the leader is required, in some cases, the vehicles that are farthest from the leader can lose communication with the leader and simply split away from the platoon. It is very likely that the controller will fail before it shows any kind of string instability.

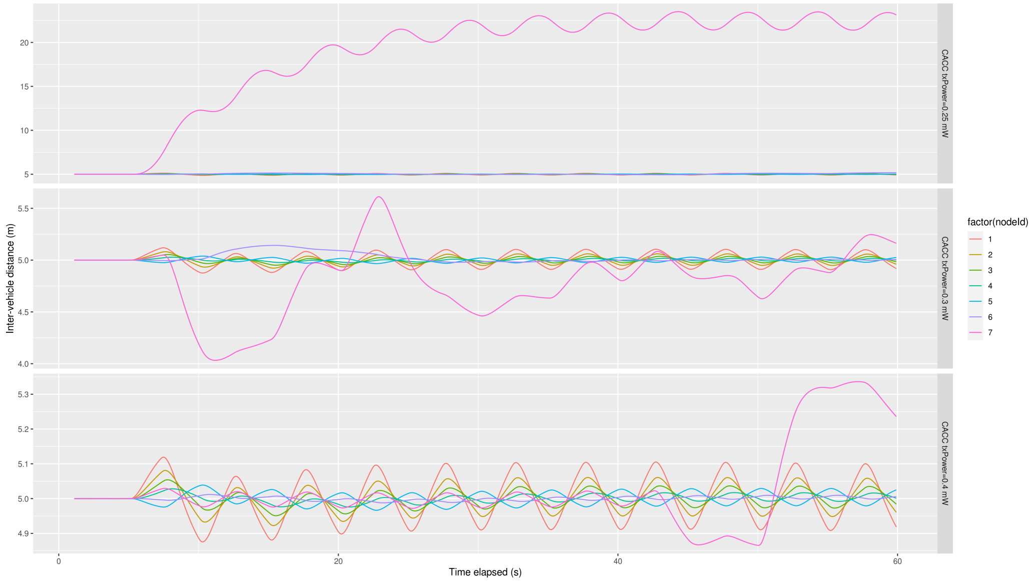

Figure 6.2.10: Varying transmission power for the CACC controller: inter-vehicle distance w.r.t. time. a) txPower=0.25 mW b) txPower=0.3 mW c) txPower=0.4 mW

It can be seen in Figure 6.2.10 that the predictions about the performance of the CACC controller for this scenario were correct. At 0.4 mW, the final follower in the platoon behaved normally until roughly 45 seconds into the simulation, at which point, it must have begun to experience packet loss in the platooning messages from the platoon leader. It started behaving erratically, and never recovered from the failure. The rest of the platoon stayed string stable for the duration of the simulation. At 0.3 mW, the behaviour that was observed in the previous scenario was seen again, but the disturbance that caused the final follower to split from the platoon occurred much earlier in the simulation. It should also be noted that the second-to-final follower also experienced a disturbance that caused it to stray from the string stable behaviour of the rest of the platoon, however, it was able to recover from the disturbance and complete the rest of the simulation as would be expected in ideal communication conditions. At 0.25 mW, the final follower immediately got separated from the rest of the platoon, and it never appeared to receive any platooning messages at all. The rest of the operation of the platoon cannot be seen on this plot, but it behaved as expected in a string stable manner throughout the simulation, except for a small disturbance that was observed in the second-to-final follower.

Ploeg Controller The Ploeg CACC controller uses a medium sized margin of safety and a information flow topology that lends itself well to adapting to adverse communication conditions. This is because the predecessor-following IFT only relies on reliable communication over the distance from one vehicle to its immediate predecessor, and as mentioned earlier, the medium-sized margin of safety means that the messages should only need to travel a maximum distance of roughly 20 metres.

Figure A.1.5 does not provide much information about the operation of the controller. This is because the amount of transmission power that was available to the vehicles in the platoon was very small for the PATH CACC controller, and the prediction is that this controller will be better equipped to handle a lack of transmission power. An investigation over a smaller range of values for the transmission power could be far more telling.

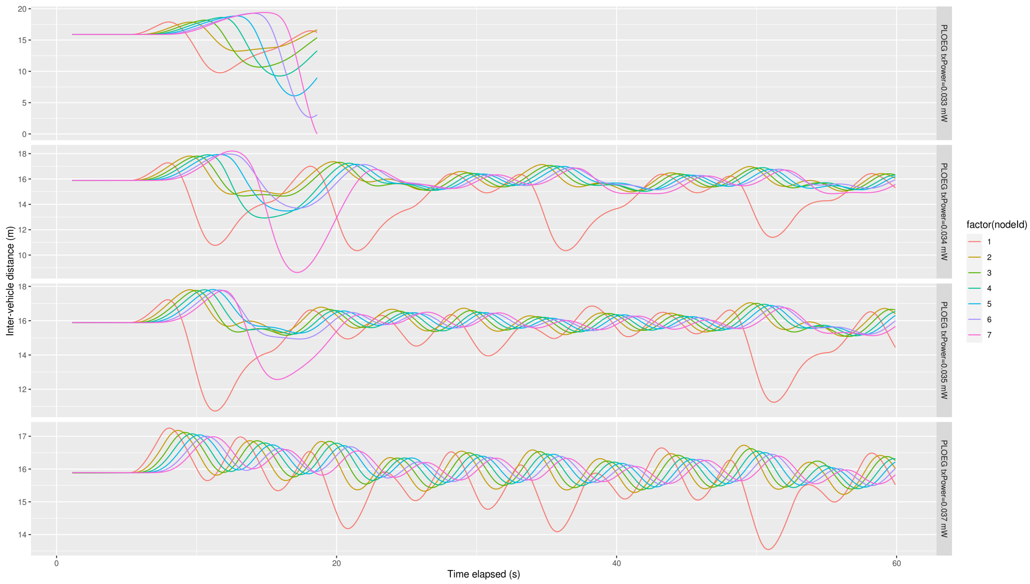

Figure 6.2.11: Varying transmission power for the Ploeg controller: inter-vehicle distance w.r.t. time. a) txPower=0.033 mW b) txPower=0.034 mW c) txPower=0.035 mW d) txPower=0.037 mW

Figure 6.2.11 shows that the predictions were correct, the controller was able to behave in a string stable fashion even for tiny values of transmission power. At the beginning of the simulation which shows the platoon each with a transmission power of 0.037 mW (Figure 6.2.11 d)), the platoon seems to behave with string stability. However, after roughly 18 seconds, the first follower begins to behave erratically and continues to behave erratically for the rest of the simulation. All of the other vehicles in the platoon behave with string stability for the duration of the simulation. At 0.035 mW, the behaviour that was observed for the previous simulation is observed again, but more pronounced. That is to say, that the first follower behaved erratically, but it behaved slightly more erratically that the previous simulation, and this had a knock-on effect on the rest of the platoon at certain points. At roughly 10 seconds into the simulation, the first follower decelerates quickly and the amplitude of the disturbance that was experienced by the other followers in the platoon was greater than that of the first follower, and the amplitude of the disturbance did not attenuate as it propagated down the platoon. The final follower also experienced a disturbance that caused it to behave unexpectedly for the first portion of the simulation, but it was quickly able to recover. Apart from the first disturbance, the rest of the platoon behaved with string stability. At 0.034 mW, similar behaviour is observed again. The first follower behaves erratically, and the final follower behaves erratically at the beginning, but recovers relatively quickly. However, because the first follower behaved so erratically, the rest of the platoon experienced some difficulty when trying to implement string stability. At several points(e.g., roughly 30 seconds and 35 seconds into the simulation) the amplitude of the disturbance is not attenuated along the platoon and therefore, the behaviour of the platoon is not string stable. At 0.033 mW, the simulation ended earlier than expected because the final follower crashed into the back of its predecessor. The only behaviour that can be seen in this simulation is very string unstable behaviour. The amplitude of the disturbances are greatly amplified along the platoon, which is what leads to the crash.

6.2.4 Actuator Delay

Actuator delay is a delay which is caused by the physical characteristics of how vehicles operate. An actuator which causes a vehicle to accelerate or decelerate (in normal conditions) takes 0.5s to fully activate. It is possible for the actuator delay to be greater or lesser, depending on the vehicle's manufacturing, and possibly even the age of the vehicle. Varying this delay effectively delays the responsiveness of the vehicles to the desired input calculated by the controller. Investigating the effect that actuator delay has on the operation of the sinusoidal scenario may give insight into the effect that actuator delay has on responsiveness.

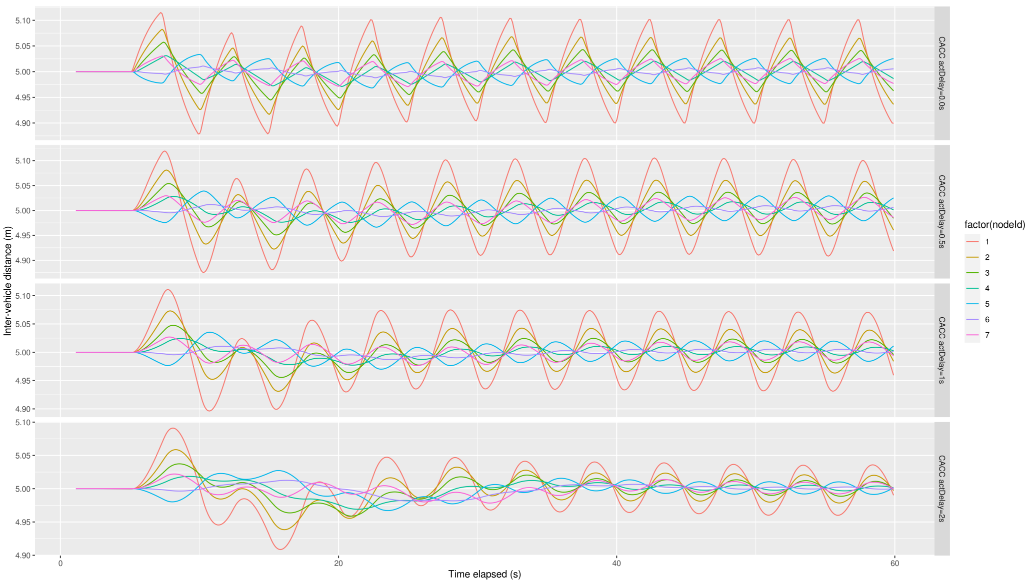

Figure 6.2.12: Varying actuator delay for CACC controller: inter-vehicle distance w.r.t. time. a) delay=0s b) delay=0.5s c) delay=1s d) delay=2s

Figure 6.2.12 displays interesting results. It can be observed that the amplitude of the inter-vehicle distance correlates almost directly with the actuator delay. Simulations which were run with longer than average actuator delay show smooth acceleration and deceleration. This is very different in contrast with the ideal simulation which included no actuator delay. having no actuator delay showed a greater amplitude in the difference of inter-vehicle distances due to the increased responsiveness. In fact, having greater actuator delay seems to give a more comfortable experience for the passenger in a vehicle. This result seems counter-intuitive, as such, investigating actuator delay for a scenario where the lead vehicle suddenly comes to a full stop may show the effects of actuator delay more clearly.

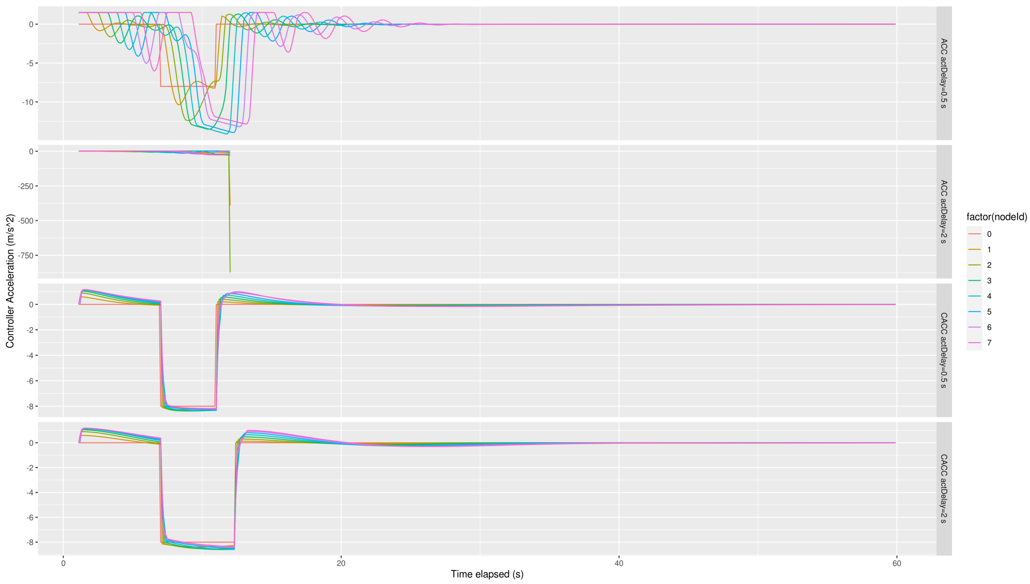

Figure 6.2.13: Varying actuator delay: inter-vehicle distance w.r.t. time a) ACC delay=0.5s b) ACC delay=2s c) CACC delay=0.5s d) CACC delay=2s

Figure 6.2.15 demonstrates that the ACC controller is more easily affected adversely by long actuation delays than the PATH CACC controller.

ACC Controller This controller has been used as an example of a controller that does not use V2V communications in order to organise the formation of a platoon for the other conditions. However, when measuring the effects of actuator delay on the performance of vehicles in a platoon, this controller is no longer the "control", as its platooning performance is not independent from the actuator delay that the platoon experiences as it was for the communication conditions. As displayed in the preamble for this section, the behaviour of a platoon in non-ideal actuation conditions can be difficult to predict. Because this kind of controller implements a constant time headway, it will be very easily affected by poor actuation delay.

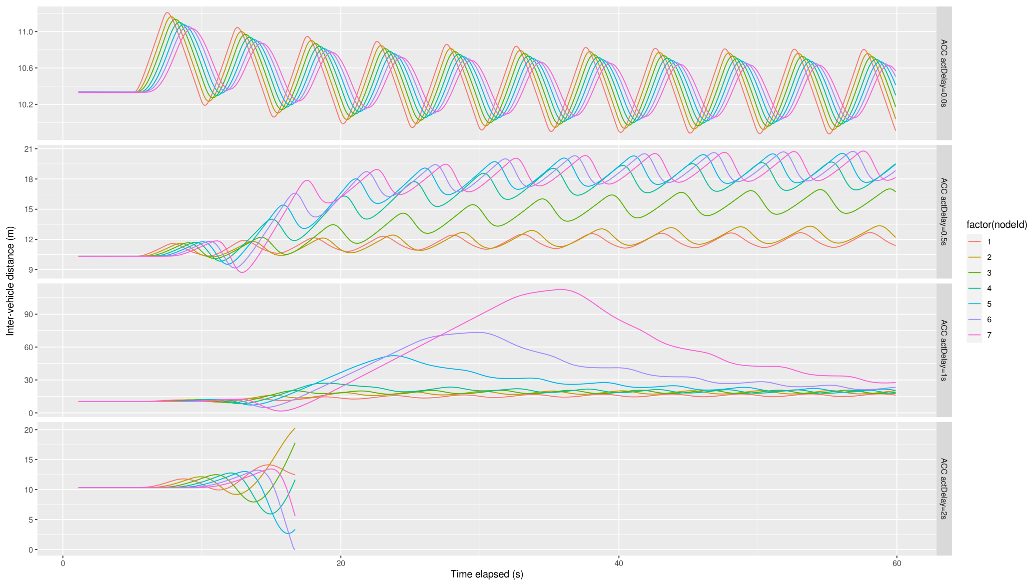

Figure 6.2.14: Varying actuator delay for the ACC controller: inter-vehicle distance w.r.t. time. a) delay=0s b) delay=0.5s c) delay=1s d) delay=2s

Figure 6.2.14 shows the behaviour of the platoon using the ACC controller for 4 different actuation conditions, and the behaviour of the platoon is drastically different for each of the conditions. This demonstrates the only prediction about the ACC controller, that it would not have consistent performance across all conditions. For an actuation delay of zero seconds, the ACC controller was able to display string stable behaviour. This is a result that has not been observed for the ACC controller so far in this project. For the plot the shows the platoon which experiences 0.5 seconds of actuation delay (this is the delay that is considered standard conditions of operation), the non-string stable behaviour that had been seen in the ACC controller is seen again. At an actuation delay of 1 second, the platoon behaved very erratically for the first oscillation of the platoon and displayed strong string unstable behaviour. The platoon managed to recover from the disturbance experienced by the final follower in the platoon, but it is no certainty that behaviour like that would not be repeated. At an actuation delay of 2 seconds, the platoon experienced a crash less than 20 seconds into the simulation. The crash resulted from an extremely unstable first oscillation, that caused the final two followers in the platoon to brake drastically but too late, and then collide.

It is clear that this controller is profoundly affected by the amount of actuation delay experienced by the vehicles in the platoon it is controlling. This could be an extremely useful result. As demonstrated by the previous sections, which dealt with non-ideal communication conditions, each of the adverse conditions could be mitigated by falling back to the communication-less ACC controller. As such, controllers which do not use V2V communications to form their platoons should be well understood, and they should be given every chance to be optimised. The goal of optimising communication-less controllers is that for real-world implementations, CAVs have extremely robust back ups. In this case, optimisation would mean finding a way to minimise the actuator delay of a CAV for its entire life-cycle. Minimising the actuation delay for all vehicles that would implement this controller could lead to drastic performance improvements, because as shown in Figure 6.2.14 having close to no actuation delay allows for string stable behaviour to be implemented.

CACC Controller Figure 6.2.12 displays the ability of PATH California's CACC controller to stay string stable across a wide range of values for actuator delay during a sinusoidal disturbance. The issues that were seen with the ACC controller are not repeated for the connected controller. As such, further investigation should be performed for the effect that a full-braking disturbance would have on a platoon controlled by the PATH CACC controller.

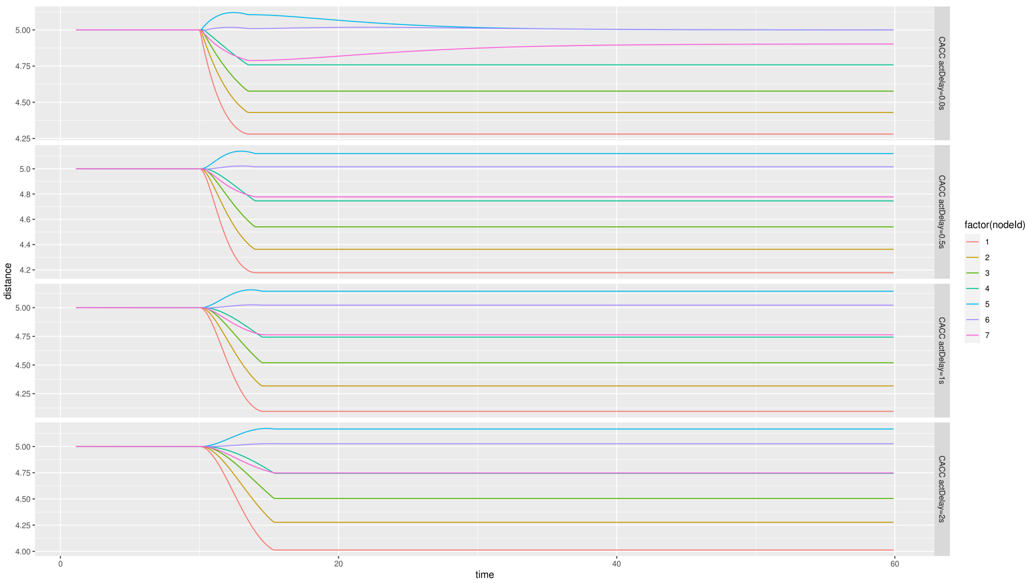

Figure 6.2.15: Varying actuator delay for the CACC controller for a full-braking disturbance: inter-vehicle distance w.r.t. time. a) delay=0s b) delay=0.5s c) delay=1s d) delay=2s

All of the plots in Figure 6.2.15 show that the inter-vehicle distance is amplifying along the platoon, so the platoon appears to present string unstable behaviour. However, for a full-braking disturbance, the vehicles come to a complete stop at the end, so the inter-vehicle distance is not the factor that is used to measure string stability. The rate of change of the inter-vehicle distance is the value that should be attenuated to display string stable behaviour. The plot shows that the first follower has to brake the hardest for the longest amount of time, and as the disturbance propagates down the platoon, each vehicle has to brake less and less.

Ploeg Controller The Ploeg CACC controller has a larger margin of safety than the PATH CACC controller, so the results that were observed when varying the actuator delay for the PATH controller should be observed again for the Ploeg controller. That is to say, the controller should be able to handle extreme values of actuator delay without causing a loss of string stability.

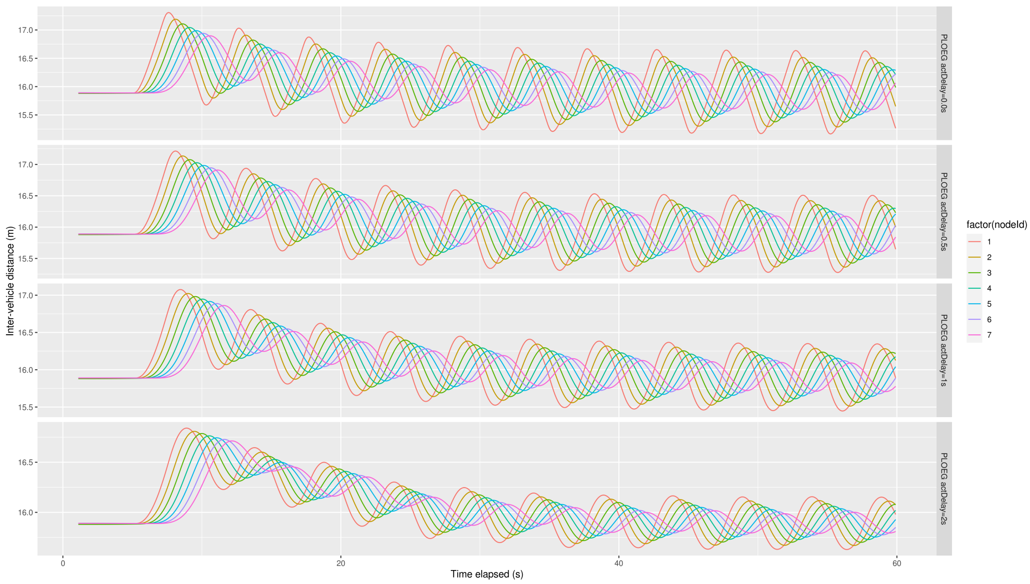

Figure 6.2.16: Varying actuator delay for the Ploeg controller: inter-vehicle distance w.r.t. time. a) delay=0s b) delay=0.5s c) delay=1s d) delay=2s

Figure 6.2.16 confirms the predicted behaviour. The controller remains string stable for all actuation conditions. It only strays from the ideal behaviour as the actuator delays get very long, and even when it strays from the ideal behaviour, the controller is able to recover from the disturbance and remain string stable.

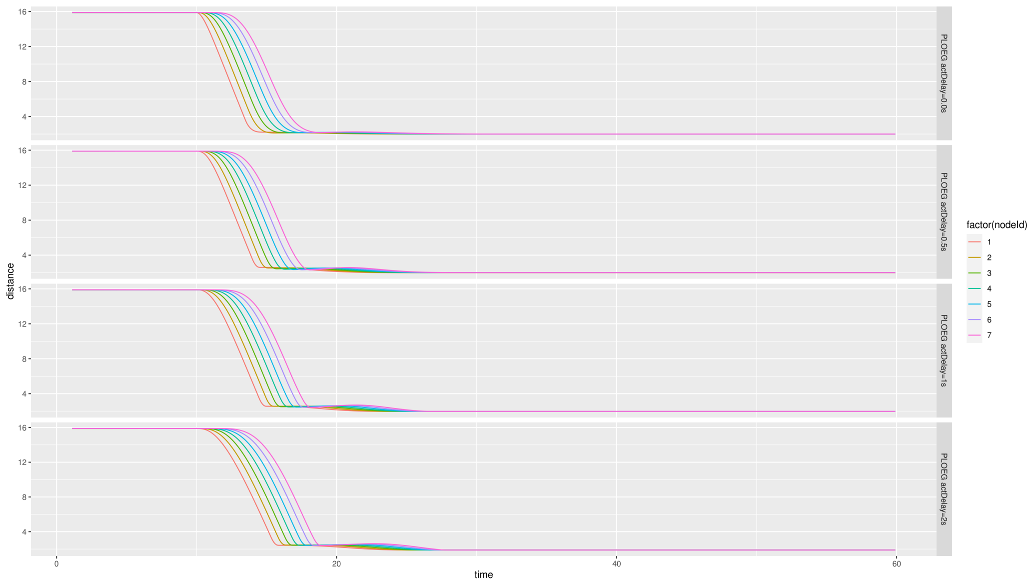

Figure 6.2.17: Varying actuator delay for the Ploeg controller for a full-braking disturbance: inter-vehicle distance w.r.t. time. a) delay=0s b) delay=0.5s c) delay=1s d) delay=2s

Yet again, further investigation was performed by testing the controller against a full-braking disturbance. The results are displayed in Figure 6.2.17. As expected, the controller is relatively unaffected by long actuator delays, no major incidents occur as the actuator delay grows.

6.2.5 Non-Merging Results Summary

For each of the combinations of platooning controllers and non-ideal conditions (where applicable), some value was found that represented the boundary between safe behaviour and unsafe behaviour. Table 6.1 summarises each of these values.

Table 6.1: Summary of results for all controllers in all conditions for non-merging scenarios

Controller	Packet loss rate	Background noise	Transmission power	Actuation Delay
ACC	N/A	N/A	N/A	0.5 s
PATH CACC	25%	-71 dBm	0.4 mW	N/A
Ploeg CACC	50%	-60.3 dBm	0.037 mW	N/A

6.3 Scenarios With Merging Manoeuvres

The next section concerns non-ideal communication conditions and non-ideal actuation conditions for scenarios that have merging manoeuvres. These scenarios differ from the previous set of scenarios because V2V communications are always required in order to implement a merging manoeuvre. The joining vehicle must query the platoon leader for information about the status of the platoon to check if the joiner is allowed to join. If the joiner is allowed to join, then the platoon leader will tell the joiner to accelerate in order to catch up with the platoon. The leader will poll the joiner to discern its distance from the back of the platoon. Once the joiner is close to the ideal spacing dictated by the controller, then the leader will tell the joiner to slow to near the speed of the platoon in order to ensure that the joiner can merge with the platoon smoothly. The biggest difference between these scenarios and the scenarios without merging manoeuvres is that network messages have to be sent over a much longer distance in order for the joiner and the platoon leader to communicate. Therefore, any issues with performance that were observed due to a lack of communications between the leader and any of the preceding vehicles should be observed for these scenarios in a more extreme fashion.

For each of the non-ideal communications situations, there should be two boundaries of limitations. The first boundary should occur whenever the joiner's first join request or the leader's acknowledgement to the joiner's request are lost over the network (as these are the longest range network messages). This kind of message loss should have little or no effect on the rest of the platoon, so if string stability was expected from the platoon, then the platoon should remain string stable. The second boundary should occur when the string stability or safety of the platoon is compromised as a result of the non-ideal communication conditions. However, the second boundary should be the same as the boundaries that were discussed in the scenarios without merging manoeuvres. This is because, if intra-platoon communications have broken down then there is a strong likelihood the the joiner's message will also be lost, and therefore the scenario should essentially be the same as the scenario without the merge.

6.3.1 Packet Loss Rate

As with the previous section which dealt with packet loss rate, the packet loss rate will be varied across multiple scenarios to investigate the performance of several controllers for non-ideal communications.

Control Experiment As mentioned in the preamble for this section, communications are required for these scenarios. As such, the controller which does not use V2V communications should experience faults once the packet loss rate is increased, because the platooning information will not be provided to the joiner. However, only a few messages need to be sent from the joiner to the leader in order to give a successful merging manoeuvre. As such, this controller should be able to withstand high rates of packet loss.

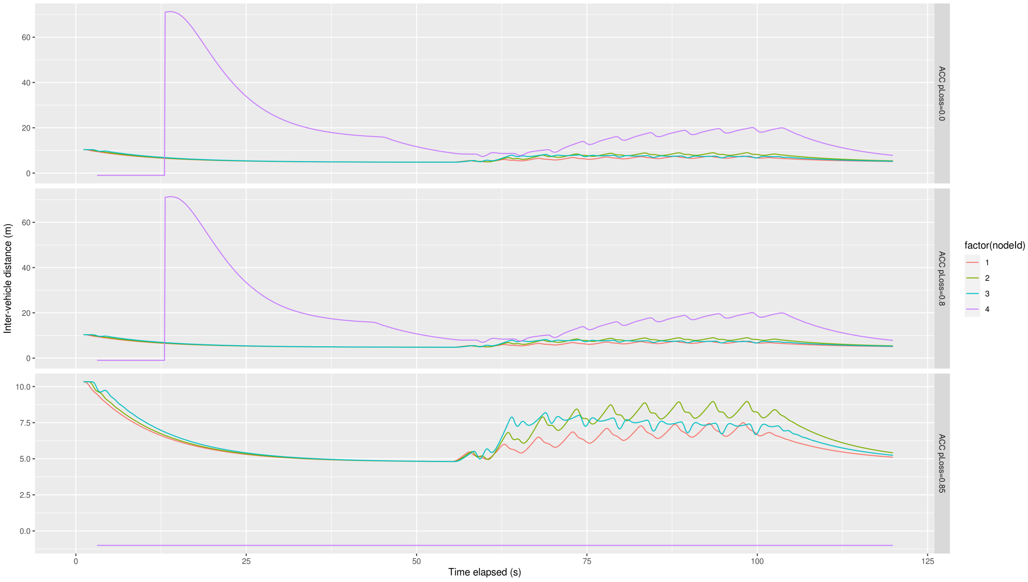

Figure 6.3.1: Varying packet loss rate for ACC controller: inter-vehicle distance w.r.t. time. a) pLoss=0 b) pLoss=0.8 c) pLoss=0.85

Figure 6.3.1 demonstrates that when the packet loss is less than 80%, the platoon behaves normally. But when the packet loss rate is increased to 85%, and therefore the joining messages are lost, the platoon behaves normally, but the joiner never joins the same lane as the rest of the platoon. When there is no vehicle in front of another vehicle, the inter-vehicle distance is recorded as -1 metres, this explains why the joiner's distance is -1 metres for the plot with packet loss at 85%.

CACC Controller In Section **??** the CACC controller was observed to act erratically before it lost its string stability. The controller began acting erratically at roughly 25-30%. The effects of packet loss on this scenario should be roughly the same, as the joiner will not be affected by a relatively low packet loss rate. But the packet loss rate that will affect the joiner is yet to be investigated for this controller.

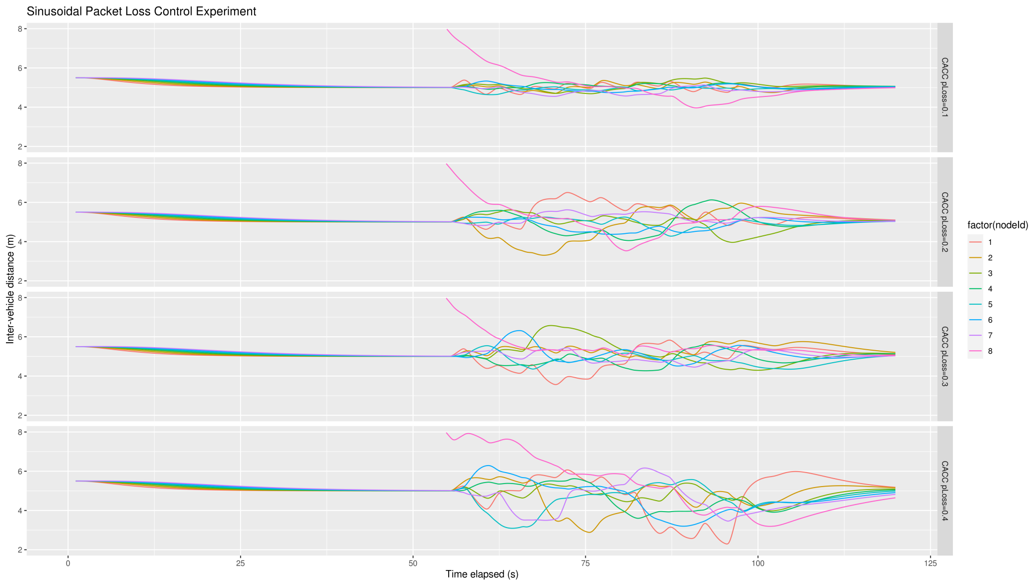

Figure 6.3.2: Varying packet loss rate for CACC controller: inter-vehicle distance w.r.t. time. a) pLoss=0.1 b) pLoss=0.2 c) pLoss=0.3 d) pLoss=0.4

Figure 6.3.2 confirms the predicted observations. The joiner was not affected by the small packet loss rates, so it joined the platoon in all of the conditions described in those plots. The platoon was not able to demonstrate string stable behaviour for small values of packet loss. A lack of string stability is not the main issue with this controller. The controller begins to act erratically far before it demonstrates predictable string unstable behaviour. This erratic behaviour is extremely similar to the behaviour observed in Figure 6.2.2.

It can reasonably be concluded that any adverse communications conditions that affect the PATH CACC controller would affect the regular functioning of the platoon before it would affect the function of a merge manoeuvre because it is quite vulnerable to packet loss.

Ploeg Controller As shown in Section **??**, the Ploeg CACC controller is able to behave with string stability up until roughly 50% packet loss. After that point the controller no longer shows strict string stability, but the inter-vehicle distances trend downwards. It would be expected for these scenarios to show the same behaviour.

Figure A.2.1 shows that the merge manoeuvre happens successfully when the packet loss rate is less than 90%. The scenario where packet loss rate is 90% shows that for this level of packet loss rate, the controller behaves very unstably and unsafely. It is not possible to determine if their is string stability present because the plots are too “zoomed out”. Further investigation will be performed between the values already observed.

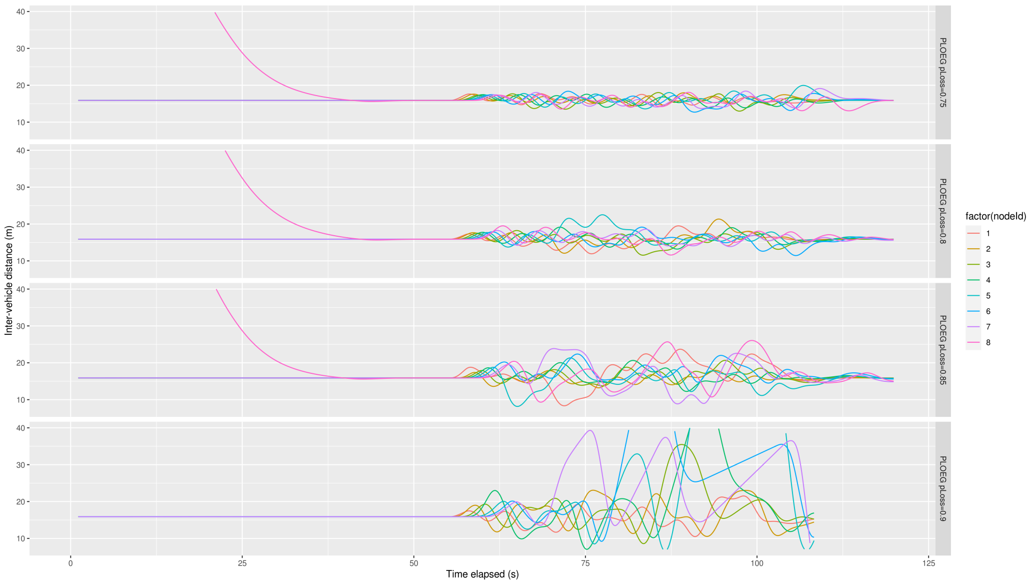

Figure 6.3.3: Varying packet loss rate for Ploeg controller: inter-vehicle distance w.r.t. time. a) pLoss=0.75 b) pLoss=0.8 c) pLoss=0.85 d) pLoss=0.9

Figure 6.3.3 demonstrates that the merging manoeuvre is successfully executed at 85% packet loss, but not 90% packet loss. This shows that the merging manoeuvre is not easily affected by packet loss rate. The figure also shows that at 75%, the controller acts in a string stable manner for the first oscillation, but as the simulation continues the stability of the behaviour degrades. The platoon acts in a string unstable fashion, but the amplitude of the disturbance is not large enough to cause very unsafe behaviour. At 80% packet loss the controller does not display string stable behaviour for any oscillations. This suggests that the boundary for packet loss rate is between

75% and 80%. At 85% packet loss, the controller behaves in a very unstable manner and it shows unsafe behaviour.

6.3.2 Background Noise

Background noise affects communications, and its effect on communications scales with distance between the two nodes that are communicating. As such, many of the effects that were observed in Section 6.2.2 may be repeated here, but before those effects are repeated, the communication between the platoon leader and the joiner will be affected.

Control Experiment The ACC controller does not use any communications to form platoons. This experiment could be used to identify the range of values for background noise that may affect the communication between the joiner and the platoon leader. The range that of values for background noise that are discovered in this experiment may not be the same values for all platoon formations. This is because the distance between the joiner and the platoon leader at the beginning of a simulation (specifically, when the joiner attempts to send the join request to the leader) can vary depending on the spacing policy put in place by the controller.

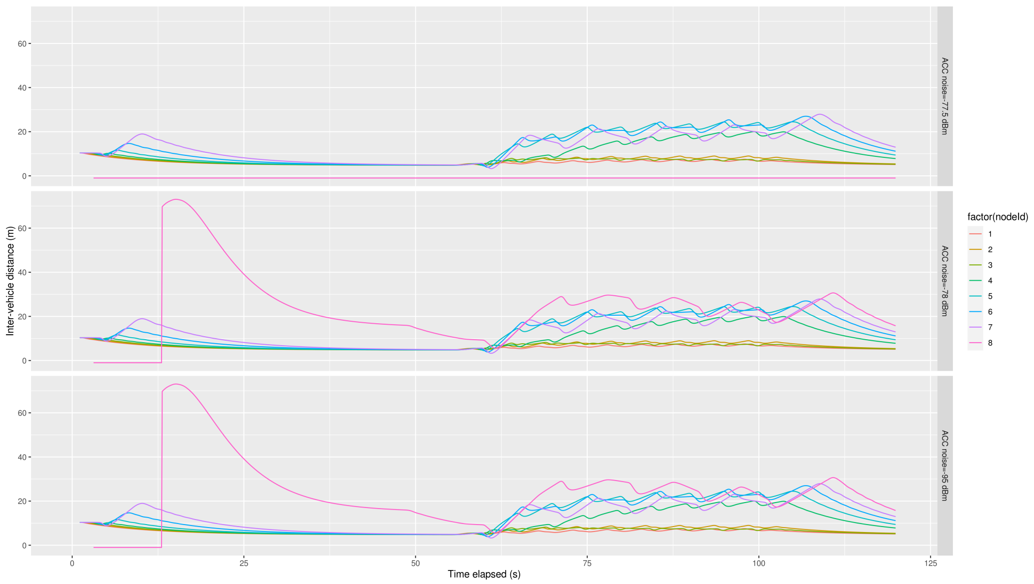

Figure 6.3.4: Varying background noise for ACC controller: inter-vehicle distance w.r.t. time. a) noise=-77.5 dBm b) noise=-78 dBm c) noise=-95 dBm

Figure 6.3.4 demonstrates the operation of the ACC controlled platoon for expected conditions of background noise power (-95 dBm). It also demonstrates the boundary of operation for communications between the joiner and the leader for the ACC controller. The boundary that can be seen is somewhere between -78 dBm and -77.5 dBm.

CACC Controller This controller has been shown to perform well in adverse communication conditions as a result of the fact that the vehicles travel close together. This controller should not display any string unstable behaviour before it begins to act unsafely. This controller should experience a similar boundary of operation for communications between the joiner and the leader to the ACC controller.

Figure A.2.2 shows that the CACC controller is able to form the platoon for values of background noise that are very close to the predicted boundary. It also shows that for background noise of -65 dBm the joiner is unable to join and other inter-vehicle communications can are effected. This comes as no surprise, because even for the scenarios that did not include a merging manoeuvre, the CACC controller began to behave unsafely at roughly -71 dBm.

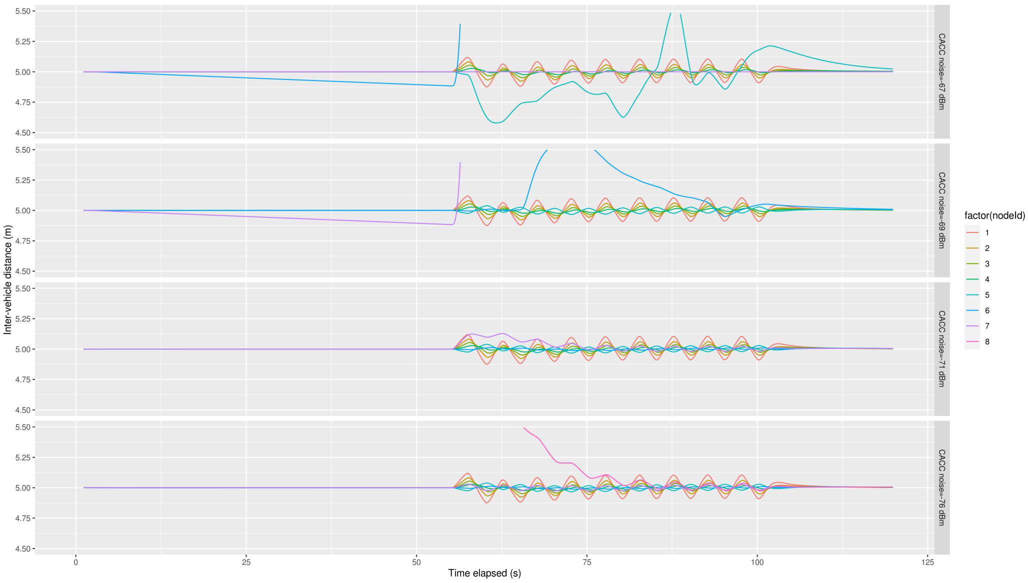

Figure 6.3.5: Varying background noise for CACC controller: inter-vehicle distance w.r.t. time. a) noise=-67 dBm b) noise=-69 dBm c) noise=-71 dBm d) noise=-76 dBm

All of the plots shown in Figure 6.3.5 have been "zoomed in" in order to distinguish some features that cannot be seen when the entire merge manoeuvre is displayed on the plots. The first important thing to note is that the joiner can be seen joining the platoon only for the scenario where the background noise is -76 dBm. This means that the communications between the leader and the joiner break down somewhere between -71 dBm and -76 dBm. This is close to the predicted range for the boundary to occur (-78 dBm, predicted by the ACC simulation). There is an interesting result demonstrated in the plot for -71 dBm: the fact that for a scenario without a merging manoeuvre, the platoon was behaving much more erratically at that point. The explanation for this behaviour is; in the scenarios without a merging manoeuvre, the platoon had less time to form the ideal platooning geometry before the oscillating disturbance began. This extra time that the platoon is afforded in the join manoeuvre scenarios could allow the minuscule errors that allowed the final follower to behave erratically in the no-join scenario to be resolved. This is why some error can be seen at -71 dBm, but not a lot. For the scenarios without merging manoeuvres, the platoon was behaving unsafely at -69.5 dBm. Therefore, it is unsurprising that at -69 dBm the final follower (not including the joiner because it never joined the platoon) breaks off from the rest of the platoon as a result of platooning message loss over the network. The second-to-last joiner is also observed to behave unsafely for that amount of background noise. at -67 dBm of background noise power, the joiner does not join the platoon, the final two followers split from the platoon when the disturbance begins, and the third-to-last follower behaves unsafely.

These results are the expected outcome of the performance of this controller. It demonstrates that any network loss that is enough to affect the platoon is enough to affect the initial joiner's message as well. The controller never reaches a point where it behaves in a string unstable manner. Vehicles simply split from the platoon because they lose their platooning messages over an unreliable network. This splitting from the back is caused by the leader-predecessor following IFT previously discussed for this controller.

Ploeg Controller The Ploeg controller has proven itself to be extremely resistant to adverse communication conditions in comparison to other controllers. However, similar that was observed for the CACC controller should be observed here, with regards to the joiner being able to join. The controller will have a boundary after which the joiner will not join the platoon, then the platoon's behaviour should mimic the scenario where there was no joining manoeuvre.

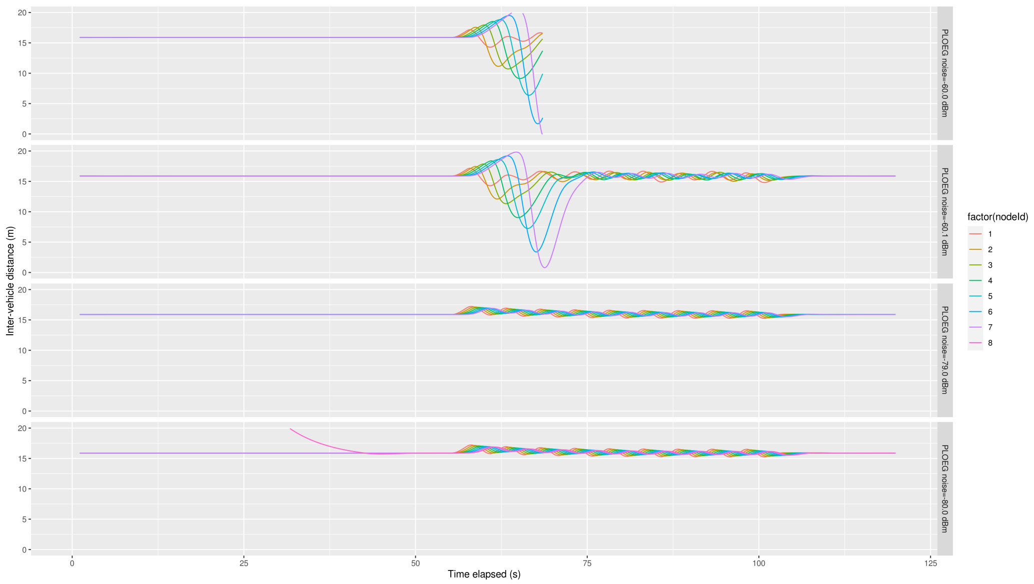

Figure 6.3.6: Varying background noise for Ploeg controller: inter-vehicle distance w.r.t. time. a) noise=-60 dBm b) noise=-60.1 dBm c) noise=-79 dBm d) noise=-80 dBm

Figure 6.3.6 shows that the platoon behaves exactly as expected. At -80 dBm the platoon behaves as though it was experiencing ideal conditions. The joiner was able to join the platoon, and string stability was demonstrated throughout the simulation. For -79 dBm the platoon behaved exactly the same as for -80 dBm (string stably), except the joiner was not able to join the platoon. For the scenarios without merging manoeuvres (Figure 6.2.7), it was observed that for -60.1 dBm, the platoon behaved in an string unstable fashion whereby the disturbance amplified along the string of vehicles until the final follower almost crashed into its predecessor. This exact same behaviour was displayed in Figure 6.3.6. The same is true for the comparison at -60 dBm, both scenarios crashed in the exact same fashion. This shows again, any background noise that affects intra-platoon communications will also affect a joiner's message. This is an unsurprising result because the way that background noise affects transmissions scales with the distance of the transmission, and in almost all cases, a joiner will be further from the platoon leader than the final follower.

6.3.3 Transmission Power

Transmission power is another control variable for simulations that affects communications and scales with the distance of the communications. The ideal value for this is a signal power of 100 mW. When tested in scenarios that do not include a merging manoeuvre, all of the controllers showed a strong resistance to the effect of low transmission power. All of the controllers were able to remain both safe and string stable for values of transmission power that were 20 times weaker than the ideal signal strength. However, because transmission power affects long-range transmissions more than short range transmissions, this set of experiments should show similar results to the experiments with merging manoeuvres that varied background noise. The joiner should be the first vehicle affected by low transmission power, and then the rest of the platoon should display similar behaviour to the scenarios without merging manoeuvres.

Control Experiment The ACC controller only uses V2V communications to allow new vehicles to join the platoon. The joining message must be sent to the platoon leader in order to be acknowledged. For these scenarios, the joiner is at the back of the platoon, as such the distance between the joiner and the platoon leader should be the greatest transmission distance that is required in the simulation. Therefore, the value that is found to cause the joining message to be lost should be a similar value for all of the other controllers.

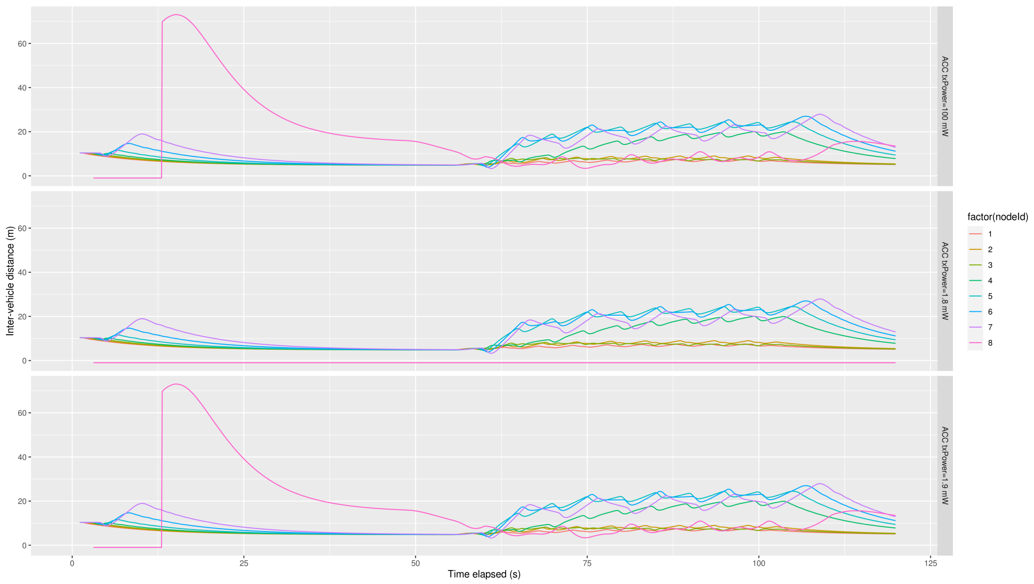

Figure 6.3.7: Varying transmission power for ACC controller: inter-vehicle distance w.r.t. time. a) txPower=100 mW b) txPower=1.8 mW c) txPower=1.9 mW

Figure 6.3.7 shows that the message between the joiner and the platoon leader is lost when the transmission power of the joiner is lowered to 1.8 mW. As stated above, this value should be close to the value that is observed for all of the connected controllers.

CACC Controller PATH's CACC controller has proven to be robust under non-ideal communication conditions. The predicted behaviour for this controller is that it will never behave in a string unstable manner. Firstly, the communication between the platoon leader and the joiner should be lost at roughly the same value for transmission power that was seen for the ACC controller. Finally, the controller should begin to act in an erratic manner, and lose its final follower(s) before it acts string unstably.

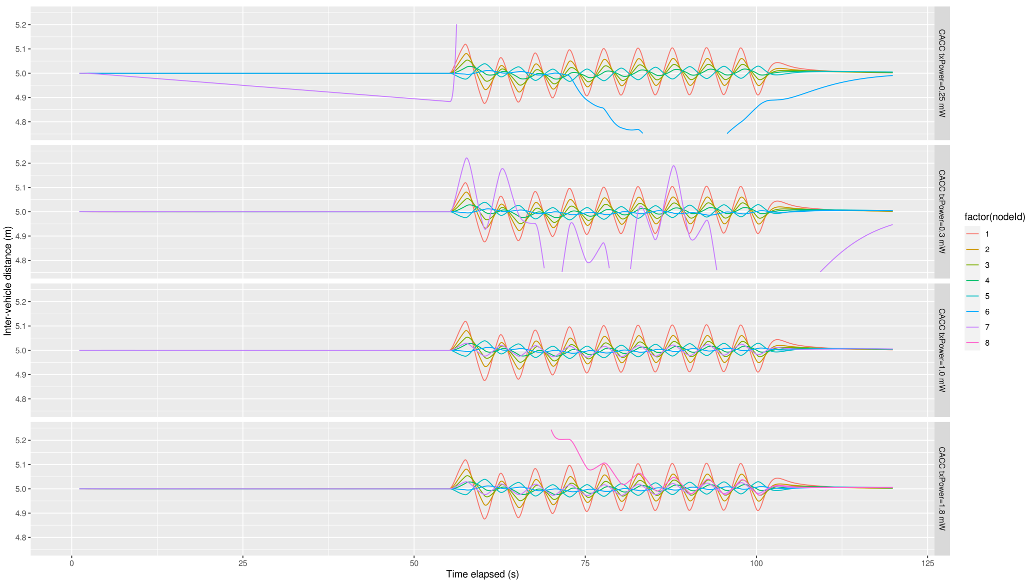

Figure 6.3.8: Varying transmission power for CACC controller: inter-vehicle distance w.r.t. time. a) txPower=0.25 mW b) txPower=0.3 mW c) txPower=1 mW d) txPower=1.8 mW

Figure 6.3.8 confirms the predicted behaviour. The platooning message between the platoon leader and the joiner is sent and received for a value of 1.8 mW of transmission power and the joiner is able to join the platoon. A value of 1 mW for transmission power shows that the platoon behaves as expected in ideal conditions minus the joiner being able to join the platoon. Similar to Figure 6.2.10 for small values of transmission power, the final follower breaks off from the back of the platoon and begins to act erratically and unsafely. It is worth noting that the prediction was confirmed, the platoon never acted in a string unstable manner.

Ploeg Controller The Ploeg CACC controller has proven to be the controller that is most robust when exposed to non-ideal communication conditions. This trend should be continued for these scenarios. It should be expected that the transmission power needed to confirm the message between the joiner and the leader is slightly larger than the 1.8 mW value that was shown for the ACC controller. This is because the Ploeg CACC controller uses a larger spacing policy and therefore the message will need to be sent over a longer distance than in the control experiment. Apart from that, the controller should prove to be as robust against small values for transmission power as it showed in Figure 6.2.11.

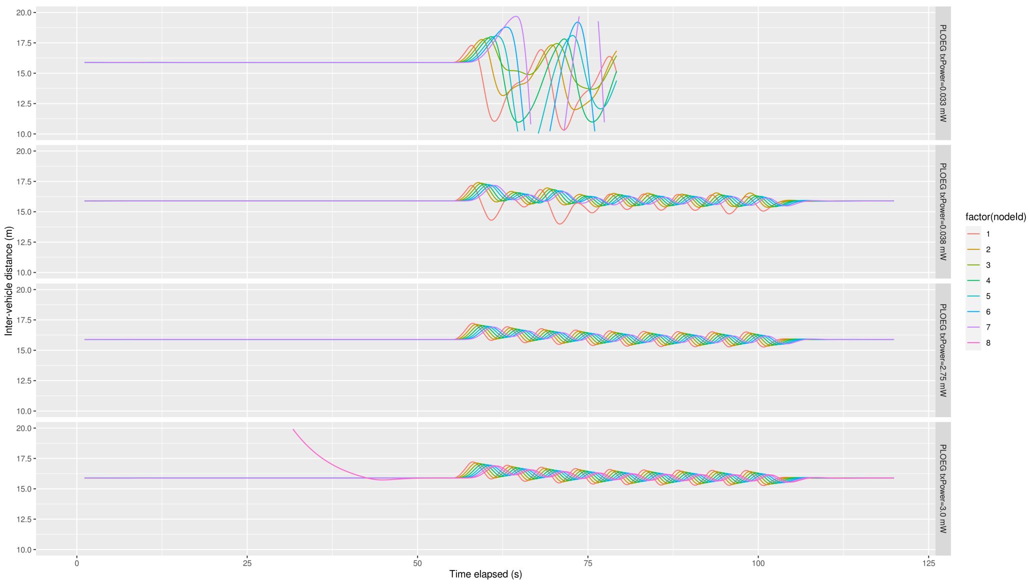

Figure 6.3.9: Varying transmission power for Ploeg controller: inter-vehicle distance w.r.t. time. a) txPower=0.033 mW b) txPower=0.038 mW c) txPower=2.75 mW d) txPower=3 mW

Figure 6.3.9 demonstrates that for values of transmission power greater than 2.75 mW, the joiner's message is able to reach the platoon leader. This fits with the predictions that the amount of transmission power needed for this controller would be greater than that for the ACC controller and the PATH CACC controller because this controller has a larger margin of safety built into its spacing policy. The rest of the behaviour that is observed is almost identical to the behaviour that was observed in Figure 6.2.11, again proving the prediction.

6.3.4 Actuator Delay

Section 6.2.4 demonstrated that the connected controllers were all able to withstand adverse actuation conditions far better than the ACC controller. Even for the most extreme case tested (an actuator delay of 2 seconds), all of the connected controllers were able to form platoons that behave in a string stable manner. Contrast this behaviour with what was observed for the ACC controller, where the controller was only able to form a string stable platoon for a negligible actuator delay, but was completely unable to create a cohesive platoon for any non-ideal conditions. For 0.5 seconds of actuator delay, the ACC controller was able to create a platoon that behaved predictably, but the platoon was extremely string stable. The conclusion drawn from this section was that a constant-time headway spacing policy is very difficult to enforce with non-ideal conditions.

ACC Controller This controller is expected to behave exactly the same for the joining manoeuvre as it did for the scenario without a joining manoeuvre. This is because the main difference between this scenario and the others is the presence of a joiner at the end of the platoon. For non-ideal actuation conditions, this joiner would not be expected to behave any differently when this joiner is added.

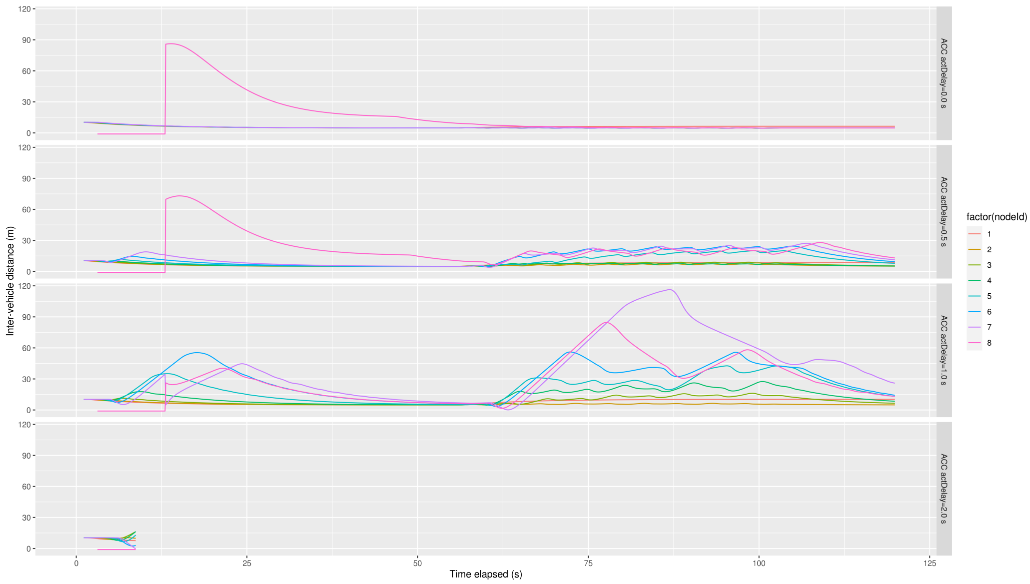

Figure 6.3.10: Varying actuation delay for ACC controller: inter-vehicle distance w.r.t. time. a) delay=0s b) delay=0.5s mW c) delay=1s d) delay=2s

Figure 6.3.10, demonstrates that the ACC controller behaves exactly the same as it did for the non-join manoeuvre scenarios. When compared to Figure 6.2.14, the predictions are confirmed. The ideal (0 seconds) actuation conditions allowed the controller to form a completely string stable platoon that allowed the joiner into the platoon, then continued to behave in a string stable fashion for the remainder of the simulation. For 0.5 seconds of actuation delay, the platoon was just as string unstable as the platoon as expected. Then for 1 second of actuation delay, the same erratic behaviour that was observed previously is seen again. And finally, for 2 seconds of actuation delay, the platoon experienced a crash as a result of extremely string unstable behaviour. Each of these results show again that minimising actuation delay is vital for the safe operation of a platoon that uses the ACC controller. Also, as discussed before, if each of the connected controllers must have a communication-less controller (that probably would use a constant-distance spacing policy) to fall-back on, then all autonomous vehicles must endeavour to minimise actuator delay to ensure safe autonomous driving.

CACC Controller As mentioned in the preamble to this section. The expected behaviour of the connected controllers is that they should not differ from the behaviour seen in Section 6.2.4. This is because the circumstance has not greatly changed. If a large change is observed as a result of the presence of a joiner, this would be an interesting result.

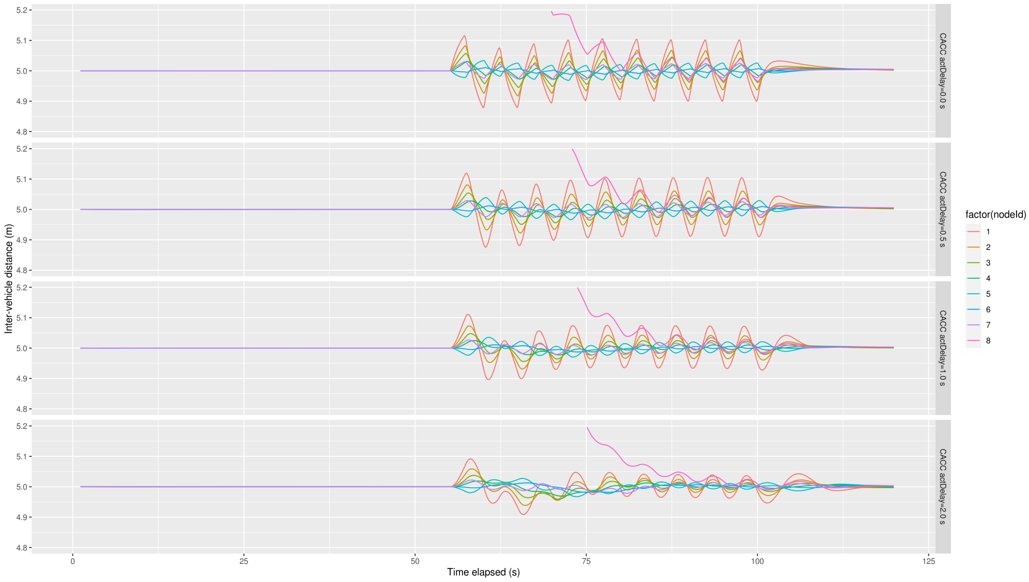

Figure 6.3.11: Varying actuation delay for CACC controller: inter-vehicle distance w.r.t. time. a) delay=0s b) delay=0.5s mW c) delay=1s d) delay=2s

Figure 6.3.11 shows that the behaviour of the CACC controller has not differed from Figure 6.2.12. The only notable difference is the joiner, and for greater actuation delays, the joiner took longer to become a completely string stable part of the platoon. For 0 seconds of actuation delay, the joiner became a string stable part of the platoon at roughly 80 seconds into the simulation. However, for 2 seconds of actuation delay, the joiner did not become a string stable part of the platoon until roughly 95 seconds into the simulation. This result is significant, because it shows that the actuation delay of vehicles in a platoon affects the performance of connected controllers. It does not affect the controller's ability to eventually become string stable, but it does affect its ability to form platoons in a timely manner.

Ploeg Controller The Ploeg controller was largely unaffected by long actuation delays in the non-merging scenarios. Its only deviation from the expected behaviour was that for 2 seconds of actuation delay, the platoon fell behind from the leader for the first few oscillations, but it eventually fully corrected and displayed string stable behaviour for the rest of the simulation. The prediction is that the controller should behave exactly the same as that. As a result of the behaviour observed in the PATH CACC controller, it is also assumed that the joiner will take longer to become a string stable part of the platoon for longer actuation delays.

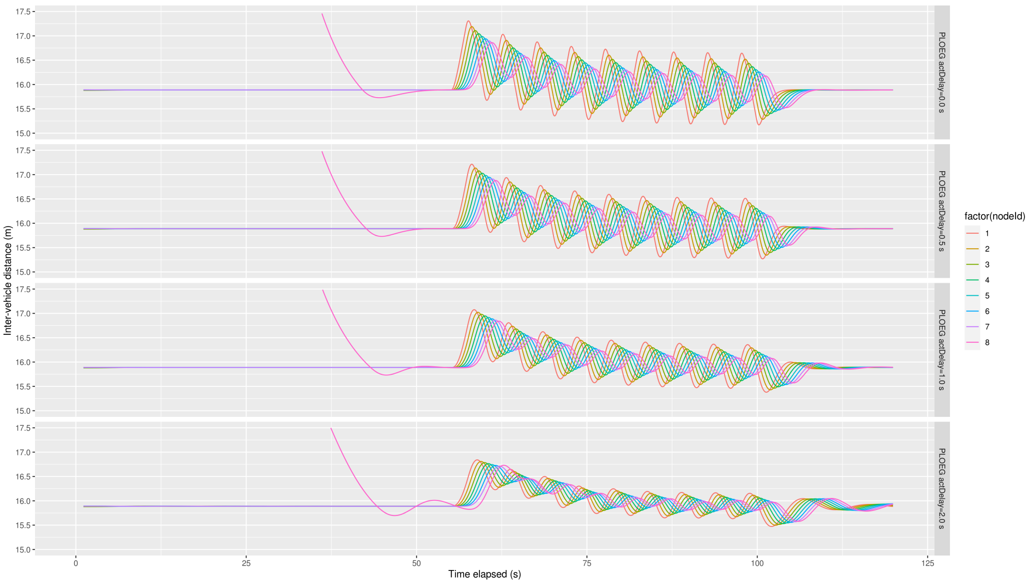

Figure 6.3.12: Varying actuation delay for Ploeg controller: inter-vehicle distance w.r.t. time. a) delay=0s b) delay=0.5s mW c) delay=1s d) delay=2s

Figure 6.3.12 confirms the prediction that the controller would be largely unaffected by the change in actuation delay. But, the prediction that the joiner would take longer to become a string stable part of the platoon for longer actuation delays is not fully confirmed because it joins at almost the exact same time for all of the actuation conditions. This result shows that actuation delay does not affect all connected controllers equally. It can be assumed that the larger margin of safety that the Ploeg controller uses compared to the PATH CACC controller contributes to the fact that the joiner is almost completely unaffected by actuator delay.

6.3.5 Merging Results Summary

For each of the combinations of platooning controllers and non-ideal conditions (where applicable), some value was found that represented the boundary between safe behaviour and unsafe behaviour. Each of these values were found to be the same for merging and non-merging manoeuvres. Therefore the results are identical to Table 6.1.

For each of the combinations of platooning controllers and non-ideal communication conditions (where applicable), some value was found that represented the boundary between the join manoeuvre successfully happening. Table 6.2 summarises each of these values. The value for packet loss rate and the PATH CACC controller is marked as "not applicable" because the controller began behaving extremely unsafely before the join manoeuvre was affected at all.

Table 6.2: Summary of results for all controllers in all communication conditions for merging scenarios

Controller	**Packet loss rate**	**Background noise**	**Transmission power**
ACC	80%	-78 dBm	1.9 mW
PATH CACC	N/A	-76 dBm	1.8 mW
Ploeg CACC	85%	-80 dBm	3 mW

Chapter 7

Conclusions and Future Work

The motivations, contributions, approach, and challenges of this project were laid out in chapter 1. Each of the concepts that are needed to perform informed analysis of experiments were detailed in chapter 2. Research related to the work conducted in the project and the simulation platforms used to collect results were examined in chapter 3. Chapter 4 discussed the methodology used to collect results using the aforementioned simulation platforms. The challenges of this project, and how they were resolved were detailed in chapter 5. Chapter 6 interpreted each of the results that were collected at a low level. Some comparisons were drawn between non-merging scenarios and merging scenarios, as some behaviours were repeated across the two scenarios. This chapter will discuss the larger trends that were observed over the course of the experiments and further work that could be done to expand on the work of this project.

Packet Loss Rate

Varying the packet loss rate in Section 6.2.1 and Section 6.3.1 demonstrated that packet loss rate can have a significant effect on the operation of a platoon for connected controllers. Packet loss rate affected connected controllers by making them behave unpredictably, it did not always affect string stability until it had already affected the overall safety of the platoon. This is an interesting result, as it demonstrates that string stability is not the only factor that needs to be considered for the safety of a platooning controller. The PATH CACC controller, which is optimised for road throughput and energy efficiency was easily affected by packet loss rate. This is because of the relatively small inter-vehicle spacing that the controller enforces, leading to a small margin for error. The Ploeg CACC controller uses a slightly larger spacing policy than the PATH CACC, therefore it was more resistant to the negative effects of packet loss. From this observation, the conclusion is that larger spacing policies are better equipped to handle packet loss.

One weakness of investigating packet loss rate when discussing it as a factor that can affect real world scenarios, is that packet loss rate is an effect of non-ideal conditions, not a condition itself. Packet loss may arise from unreliable channels, but there must be a physical reason for the channel being unreliable. In a platooning scenario, the condition that is causing the unreliable channel may have other effects e.g., background noise can cause packet loss, but it scales with distance, whereas packet loss rate alone does not scale with distance.

Background Noise

Background noise simulates an environment with a lot of radio interference that can affect all V2V transmissions. An example of a scenario where radio interference could be high enough to affect communications is in the event of a signal-jamming attack. Section 6.2.2 shows that the minimum background noise needed to create sufficient packet loss for any of the controllers to begin failing is -71dBm. Skomal [43] demonstrates that even in urban environments where radio transmissions would be densest and most likely to cause interference, the observed power of this background noise should not exceed -110dBm (given that our transmission frequency is 500MHz). This result shows that the likelihood of background noise affecting these V2V communications is very low in day-to-day situations. If the noise of an environment exceeds the roughly -71dBm threshold, then it is more likely that the communications are deliberately being interfered with rather than interfered with by some background noise.

A denial of service (DoS) jamming attack could be very effective if a controller has no way of detecting when it is being attacked. Waiting for packet loss statistics, or calculating packet delay (like in [27]) could take too long in this case. A coordinated attack could successfully be carried out by jamming the V2V signal, then causing the platoon leader to brake suddenly. If the vehicle relied solely on packet loss rate statistics or packet delay to calculate when it should degrade to a communication-less controller, then the vehicles in the platoon would crash into one another for this kind of attack. Therefore, background noise is a communication condition that should be closely monitored by the platooning controller if the platooning controller needs to be aware of attacks.

Otherwise, background noise affected communications in the predicted fashion. In that, the effect that background noise has on a transmission is proportional to the distance between the two nodes that are trying to communicate. Therefore, using a predecessor-following IFT, like the Ploeg CACC, is the best strategy for combating background noise, as the average transmission is between only a vehicle and its predecessor. This conclusion follows from the fact that the PATH CACC controller was more easily affected by background noise than the Ploeg CACC controller. The inter-vehicle spacing in the Ploeg controller was greater than in the PATH controller, so the only factor that could have contributed to this behaviour is the IFT. PATH uses a leader-predecessor following IFT. This is why, when the PATH controller began to behave erratically, only the vehicles at the end of the platoon (far away from the leader) were affected. This is a significant result, as it shows that more “complex” IFTs can lead to weaknesses in the platoon.

Transmission Power

Transmission power is a communication condition that has not been well researched in the field of autonomous vehicle platooning. This is likely because a vehicle with small transmission power would be a rare occurrence in a real world scenario (let alone a whole platoon of them), and is something that could not happen suddenly. This is due to it being a communication factor that is dependent on the hardware on board a vehicle, not the quality of a network. However, because it has not been well investigated, its behaviour cannot simply be assumed. Investigation into transmission power confirmed the assumptions that it is not a factor that needs to be readily considered for platooning safety. The simulations that were run involved altering the transmission power of every vehicle in a platoon. This would be an unlikely occurrence in the real world. The negative effects that were observed rarely resulted in extremely unsafe behaviour, except in unrealistically small values for transmission power.

Actuator Delay

Investigating the effects of actuation delay on all of the controllers yielded interesting results. The connected controllers were hardly affected by extremely poor actuation conditions. The behaviour of the connected controller for unrealistically small values for actuation delay (no delay) were not vastly different from their behaviour for long actuation delays. The main difference was that for small values of actuation delay, the connected controllers appeared to act in a jerky fashion that would cause an uncomfortable ride for a passenger. However, safety and string stability were not affected. On the other hand, the ACC controller was very easily affected by actuation delay. It showed that for no actuation delay, the controller behaved in a string stable manner. For the expected actuation delay (0.5 seconds) the controller behaved in a string unstable manner but never acted in a fashion that may be unsafe enough to cause a crash. For extremely large values of delay (2 seconds) the platoon could end up crashing into one another due to string unstable behaviour. This is an important result, as it shows that despite actuation delays not affecting CACC controllers a large amount, ACC controllers are very sensitive to actuation delays.

Final Conclusions

A key observation that must be made is that results which showed changes in platooning behaviour for transmission power and background noise are unrealistically non-ideal. For transmission power, the ideal power is 100 mW; the PATH controller (which performed worse than the Ploeg controller) required the transmission power of every single vehicle in a platoon to lowered to less than 1 mW to affect normal platooning behaviour. For background noise, the expected power of the noise is -95 dBm. The PATH controller was only greatly affected by background noise at -70 dBm, and the Ploeg controller at -60 dBm. Therefore it can reasonably be assumed that if a network's conditions degrade to the point where transmission power and/or background noise have the negative effects shown in this project, then the network and/or hardware is being deliberately tampered with by an external force. As described above, this may be the effects of an intentional attack on a platoon, as such, these communication conditions may only need to be monitored for platoons which suspect an attack.

Another observation that was mentioned in Section 6.3 is that merging scenarios present no further safety risks than regular platooning conditions. This is because, many of the adverse effects of the non-ideal communication conditions affected platoons in two stages. The first stage occurred when the joining vehicle's message to the platoon leader was not received due to adverse conditions. The next stage was when the platoon began to behave in an unsafe manner. These two stages took place at very different values for the communication condition in all cases. This is likely due to the fact that the longest range transmission that was needed in the merging scenarios was the joiner's initial communications with the platoon leader, and the communication conditions were affected by range of transmission.

A conclusion that can be drawn from the observation of the connected controllers is that if non-ideal network conditions are detected and the controller does not degrade to some communication-less alternative, the platoon becomes a liability on the road. Therefore, a communication-less fall-back *must* be implemented for all connected controllers. Combining the conclusion that all connected controllers must be programmed to have a communication-less controller and communication-less controllers are sensitive to actuation delay, there is another conclusion that can be drawn. All vehicles that implement autonomous platooning controllers must endeavour to minimise actuation delay in order to maximise the controller's ability to behave in a string stable manner for all conditions.

The questions that this research project posed were: do non-ideal communication and non-ideal actuation conditions have any significant effect on the behaviour (particularly the string stable behaviour) of different platoon controllers? And if they have a significant effect, when do these effects render a particular platooning controller unsafe?

In short, the answer to the first question is: yes, each of the non-ideal conditions were able to affect at least one of the platooning controllers enough to have a significant impact on its string stable behaviour. The second question has many answers, which were outlined in the Results chapter (chapter 6), but it was shown that each condition had a boundary, past which, some of the platooning controllers began to display unsafe behaviour.

Future Work

The possible future work for this project would include investigating different non-ideal communication conditions that could have a large effect on a network like packet delay (discussed in Section 3.1.1). Another approach to investigating non-ideal actuation could be taken by isolating braking actuation from acceleration actuation. It would also be interesting to combine non-ideal communication conditions and non-ideal actuation conditions to investigate the effects on a platoon.

Appendices

Appendix A

Results

A.1 Non-merging scenarios

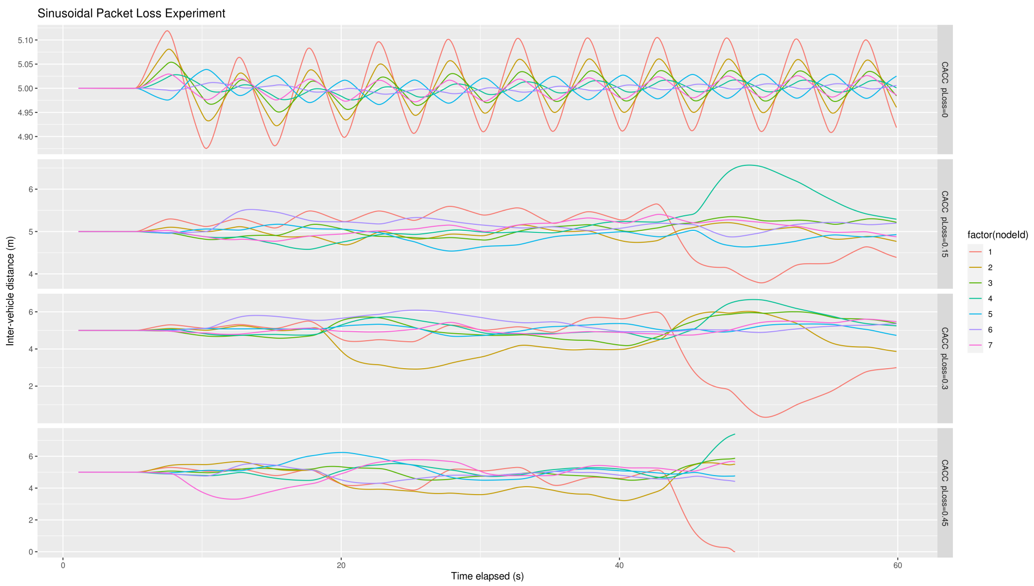

Figure A.1.1: Varying packet loss rate for CACC controller over a wide range: inter-vehicle distance w.r.t. time. a) pLoss=0 b) pLoss=0.15 c) pLoss=0.3 d) pLoss=0.45

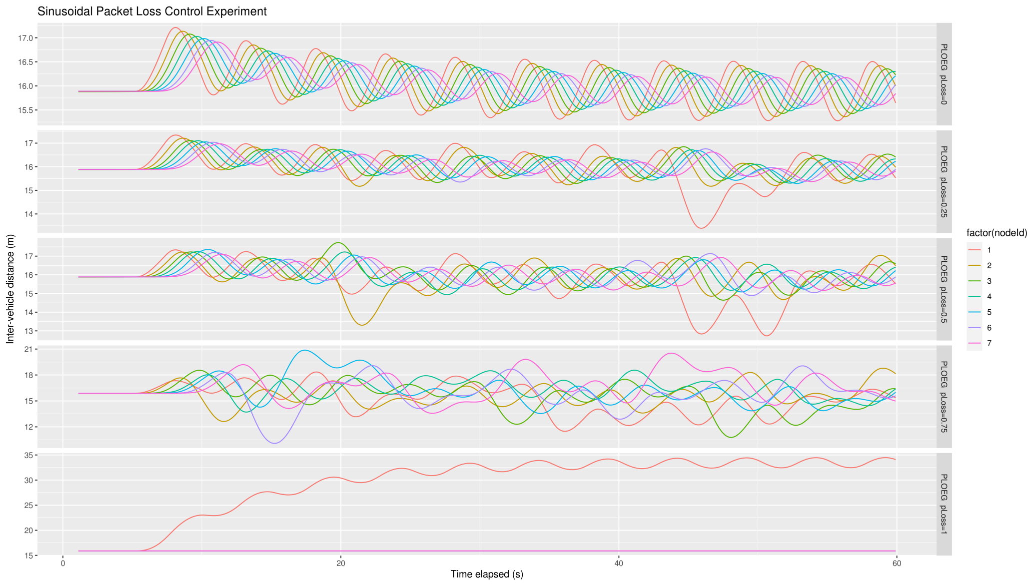

Figure A.1.2: Varying packet loss rate for Ploeg controller over a wide range: inter-vehicle distance w.r.t. time. a) pLoss=0 b) pLoss=0.25 c) pLoss=0.5 d) pLoss=0.75 e) pLoss=1

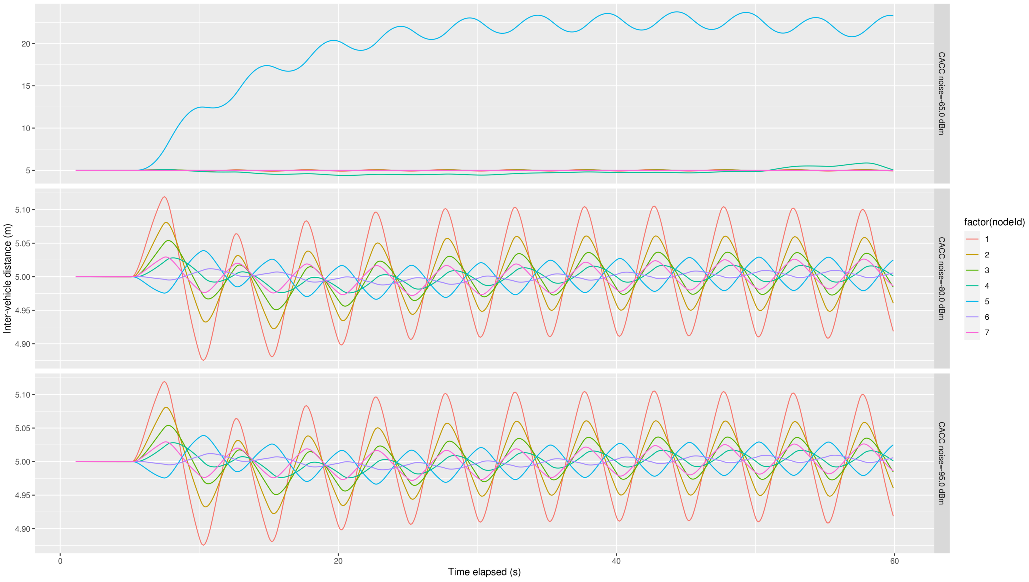

Figure A.1.3: Varying background noise for the CACC controller: inter-vehicle distance w.r.t. time. a) noise=-65 dBm b) noise=-80 dBm c) noise=-95 dBm

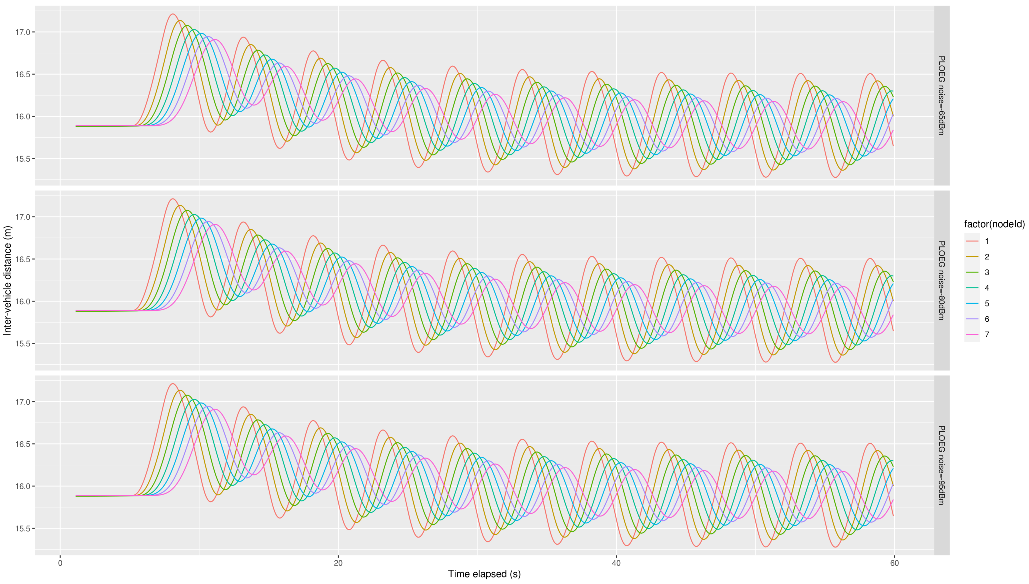

Figure A.1.4: Varying background noise for the Ploeg controller: inter-vehicle distance w.r.t. time. a) noise=-65 dBm b) noise=-80 dBm c) noise=-95 dBm

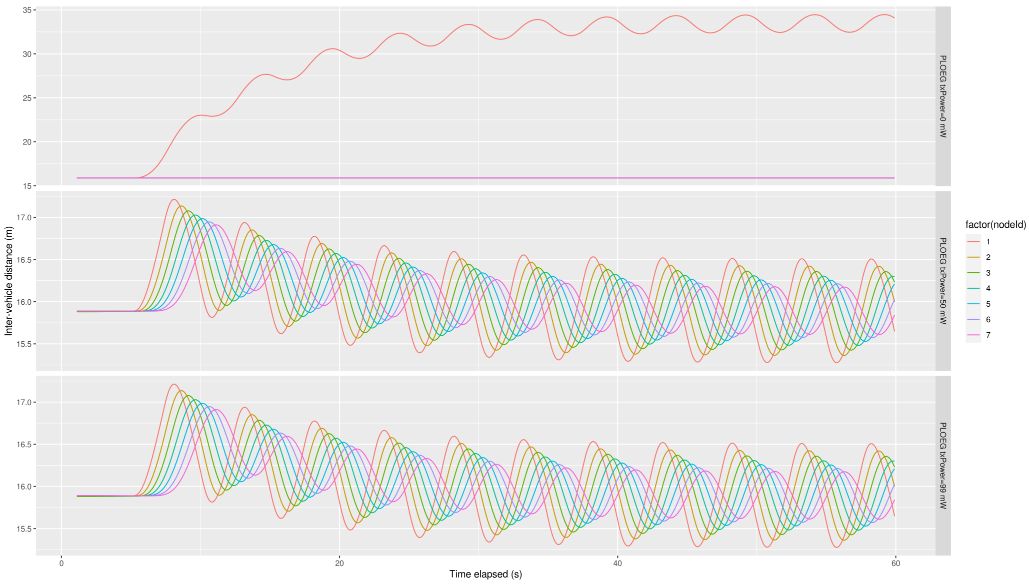

Figure A.1.5: Varying transmission power for the Ploeg controller: inter-vehicle distance w.r.t. time. a) txPower=0 mW b) txPower=50 mW c) txPower=99 mW

A.2 Merging scenarios

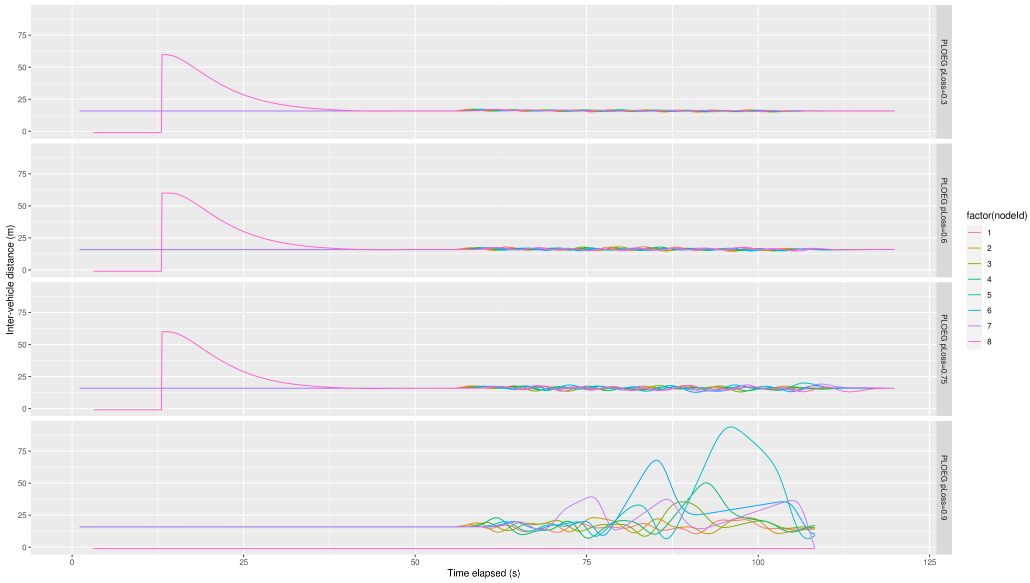

Figure A.2.1: Varying packet loss rate for Ploeg controller: inter-vehicle distance w.r.t. time. a) pLoss=0.3 b) pLoss=0.6 c) pLoss=0.75 d) pLoss=0.9

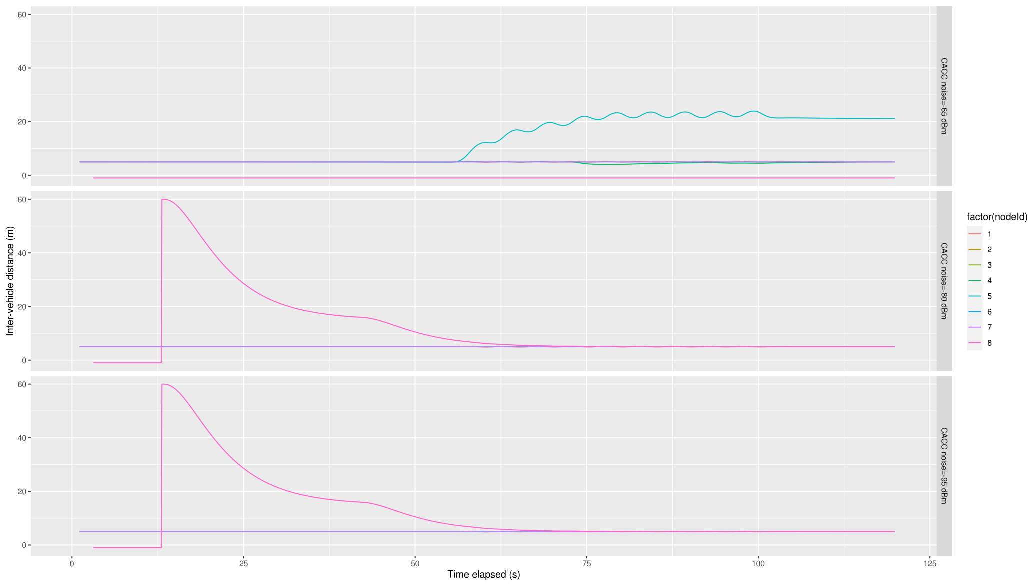

Figure A.2.2: Varying background noise for CACC controller: inter-vehicle distance w.r.t. time. a) noise=-65 dBm b) noise=-80 dBm c) noise=-95 dBm

Bibliography

[1] Veins, copyright Christoph Sommer, "Veins achitecture," 2013. [Online; accessed May 15, 2021].

[2] M. Turcotte, "Dependence on cars in urban neighbourhoods," *Canadian Social Trends*, vol. 85, pp. 20–30, 2008.

[3] I. Cameron, T. Lyons, and J. Kenworthy, "Trends in vehicle kilometres of travel in world cities, 1960–1990: underlying drivers and policy responses," *Transport policy*, vol. 11, no. 3, pp. 287–298, 2004.

[4] T. Vanderbilt, *Traffic: Why We Drive the Way We Do (and What It Says About Us)*. Vintage Books, 2009.

[5] R. Hoogendoorn, B. van Arerm, and S. Hoogendoom, "Automated driving, traffic flow efficiency, and human factors: Literature review," *Transportation Research Record*, vol. 2422, no. 1, pp. 113–120, 2014.

[6] A. Talebpour and H. S. Mahmassani, "Influence of connected and autonomous vehicles on traffic flow stability and throughput," *Transportation Research Part C: Emerging Technologies*, vol. 71, pp. 143 – 163, 2016.

[7] F. Codevilla, E. Santana, A. M. López, and A. Gaidon, "Exploring the limitations of behavior cloning for autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9329–9338, 2019.

[8] Z. Yao, T. Xu, Y. Jiang, and R. Hu, “Linear stability analysis of heterogeneous traffic flow considering degradations of connected automated vehicles and reaction time,” *Physica A: Statistical Mechanics and its Applications*, vol. 561, p. 125218, 2021.

[9] A. I. Delis, I. K. Nikolos, and M. Papageorgiou, “Simulation of the penetration rate effects of acc and cacc on macroscopic traffic dynamics,” in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 336–341, IEEE, 2016.

[10] A. Festag, “Cooperative intelligent transport systems standards in europe,” *IEEE communications magazine*, vol. 52, no. 12, pp. 166–172, 2014.

[11] J. B. Kenney, “Dedicated short-range communications (dsrc) standards in the united states,” *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.

[12] C. Bergenhem, S. Shladover, E. Coelingh, C. Englund, and S. Tsugawa, “Overview of platooning systems,” in *Proceedings of the 19th ITS World Congress, Oct 22-26, Vienna, Austria (2012)*, 2012.

[13] G. J. L. Naus, R. P. A. Vugts, J. Ploeg, M. R. J. van de Molengraft, and M. Steinbuch, *Senior Member IEEE*, “String-stable CACC design and experimental validation: A frequency-domain approach,” *IEEE Transaction on Vehicular Technology*, vol. 59, no. 9, 2010.

[14] S. Feng, Y. Zhang, S. E. Li, Z. Cao, H. X. Liu, and L. Li, “String stability for vehicular platoon control: Definitions and analysis methods,” *Annual Reviews in Control*, vol. 47, pp. 81–97, 2019.

[15] L. Peppard, “String stability of relative-motion pid vehicle control systems,” *IEEE Transactions on Automatic Control*, vol. 19, no. 5, pp. 579–581, 1974.

[16] K.-C. Chu, "Optimal dencentralized regulation for a string of coupled systems," *IEEE Transactions on Automatic Control*, vol. 19, no. 3, pp. 243–246, 1974.

[17] D. Swaroop and J. Hedrick, "String stability of interconnected systems," *IEEE Transactions on Automatic Control*, vol. 41, March8 1996.

[18] C. Wang and H. Nijmeijer, "String stable heterogeneous vehicle platoon using cooperative adaptive cruise control," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pp. 1977–1982, IEEE, 2015.

[19] C. Flores, *Control architecture for adaptive and cooperative car-following*. PhD thesis, Université Paris sciences et lettres, 2018.

[20] M. Goli and A. Eskandarian, "A systematic multi-vehicle platooning and platoon merging: Strategy, control, and trajectory generation," in *ASME 2014 Dynamic Systems and Control Conference*, American Society of Mechanical Engineers Digital Collection, 2014.

[21] F. Morbidi, P. Colaneri, and T. Stanger, "Decentralized optimal control of a car platoon with guaranteed string stability," in *2013 European Control Conference (ECC)*, pp. 3494–3499, 2013.

[22] J. Heinovski and F. Dressler, "Platoon formation: Optimized car to platoon assignment strategies and protocols," in *2018 IEEE Vehicular Networking Conference (VNC)*, pp. 1–8, 2018.

[23] H. Min, Y. Yang, Y. Fang, P. Sun, and X. Zhao, "Constrained optimization and distributed model predictive control-based merging strategies for adjacent connected autonomous vehicle platoons," *Communication and Fog/Edge Computing Towards Intelligent Connected Vehicles (ICVS)*, vol. 7, 2019. Special Section.

[24] M. Goli and A. Eskandarian, "Merging strategies, trajectory planning and controls for platoon of connected, and autonomous vehicles," *Inter-*

national Journal of Intelligent Transportation Systems Research, vol. 18, pp. 153–173, 2020.

[25] A. Mosebach, S. Röchner, and J. Lunze, “Merging control of cooperative vehicles,” *IFAC-PapersOnLine*, vol. 49, pp. 168–174, 2016.

[26] Y. A. Harfouch, S. Yuan, and S. Baldi, “An adaptive switched control approach to heterogeneous platooning with intervehicle communication losses,” *IEEE Transactions on Control of Network Systems*, vol. 5, no. 3, pp. 1434–1444, 2018.

[27] J. Ploeg, E. Semsar-Kazerooni, G. Lijster, N. van de Wouw, and H. Nijmeijer, “Graceful degradation of cooperative adaptive cruise control,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 488–497, 2014.

[28] R. Teo, D. M. Stipanovic, and C. J. Tomlin, “Decentralized spacing control of a string of multiple vehicles over lossy datalinks,” in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*, vol. 1, pp. 682–687, IEEE, 2003.

[29] A. Alipour-Fanid, M. Dabaghchian, H. Zhang, and K. Zeng, “String stability analysis of cooperative adaptive cruise control under jamming attacks,” in *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*, pp. 157–162, IEEE, 2017.

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

[31] K. Devika, N. Sridhar, V. R. S. Yellapantula, and S. C. Subramanian, “Control of heavy road vehicle platoons incorporating actuation dynamics,” in *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)*, pp. 1434–1439, IEEE, 2019.

[32] M. di Bernardo, A. Salvi, S. Santini, and A. S. Valente, "Third-order consensus in vehicles platoon with heterogeneous time-varying delays," *IFAC-PapersOnLine*, vol. 48, no. 12, pp. 358–363, 2015.

[33] A. S. Valente, S. Santini, and P. A. Palladino, *Cooperative Driving in Inter-Vehicular Communication Network*. PhD thesis, Ph. D. thesis, University of Naples, Federico II-Italy, 2015.

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

[35] R. Rajamani, H.-S. Tan, B. K. Law, and W.-B. Zhang, "Demonstration of integrated longitudinal and lateral control for the operation of automated vehicles in platoons," *IEEE Transactions on Control Systems Technology*, vol. 8, no. 4, pp. 695–708, 2000.

[36] J. Ploeg, "Analysis and design of controllers for cooperative and automated driving," 2014.

[37] J. Ploeg, D. P. Shukla, N. van de Wouw, and H. Nijmeijer, "Controller synthesis for string stability of vehicle platoons," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, April 2014.

[38] J. Alleleijn, H. Nijmeijer, S. Öncü, and J. Ploeg, *Lateral string stability of vehicle platoons*. D & C, Eindhoven University of Technology, 2014. Internship report.

[39] J. Ploeg and R. de Haan, "Cooperative automated driving: From platooning to maneuvering.," in *SMARTGREENS*, p. 9, 2019.

[40] Plexe, copyright Michele Segata, "Plexe documentation," 2021. [Online; accessed May 15, 2021].

[41] C. Cseh, "Architecture of the dedicated short-range communications (dsrc) protocol," in *VTC'98. 48th IEEE Vehicular Technology Conference. Pathway to Global Wireless Revolution (Cat. No. 98CH36151)*, vol. 3, pp. 2095–2099, IEEE, 1998.

[42] M. Segata, F. Dressler, and R. Lo Cigno, “Jerk beaconing: A dynamic approach to platooning,” in *2015 IEEE Vehicular Networking Conference (VNC)*, pp. 135–142, 2015.

[43] E. N. Skomal, “The dimensions of radio noise,” in *1969 IEEE Electromagnetic Compatibility Symposium Record*, pp. 18–28, 1969.