Optimization-based Fault Mitigation in Automated Driving

Master thesis

Niels Christiaan Lodder

Delft University of Technology

TU Delft

TNO

Mechanical, Maritime and Materials Engineering Faculty
DELFT UNIVERSITY OF TECHNOLOGY

MASTER THESIS
MSC. MECHANICAL ENGINEERING, VEHICLE ENGINEERING TRACK

Optimization-based Fault Mitigation in Automated Driving

Niels Christiaan Lodder

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday March 25, 2021.

This thesis is confidential and cannot be made public until September 25, 2021.

The thesis committee consisted out of the following members:

dr. ir. J. Alonso Mora	Chairman	CoR, Delft University of Technology
dr. L. Ferranti	Supervisor	CoR, Delft University of Technology
dr. ir. B. Shyrokau	Committee member	CoR, Delft University of Technology
dr. ir. E. Silvas	Guest member	Integrated Vehicle Safety, TNO
ir. C.J. van der Ploeg	Guest member	Integrated Vehicle Safety, TNO

This thesis is made possible by and performed at the Integrated Vehicle Safety department of TNO, Helmond. The author is grateful for their cooperation and support throughout the whole process.

Acknowledgements

This document has been in the making for almost 10 years, back then not knowing how it would turn out, what topic it would be about and when I would present this to you. The only thing I knew back then is that I wanted to be proud of the work that I have done and I can tell you confidently that I am very proud. However, this of course would not have been possible without a number of people that have supported, encouraged or inspired me in the past 10 years.

First and foremost I have to thank my parents for the support they have given me during my studies, but actually through my whole life. It hasn't always been a smooth ride, we have had a lot of conversations, discussions and ups and downs together, but in the end they always had faith in me. They have given me the opportunity and space to develop myself as a person, to become a better human being, which I will be grateful for for the rest of my life.

Secondly, of course, with parents comes family, which I have to thank as well for all the laughs, interesting and inspiring conversations we had during family events and the random nights out together. This helped me to relax and recharge, be positive and look forward to what is ahead, but also not be afraid of the things to come or problems you might encounter.

Furthermore, I would like to thank my friends for distracting me when I needed to study and letting me practice my elevator pitches about the beauty of life on them. All jokes aside, you have always pulled through when I needed you, motivating me to make the right choices and telling me the truth when I needed an honest answer, or just the truth. You have also been a great motivation to me, getting me out of bed early and pulling me away from my laptop in the weekends to also get some fresh air and enjoy life. I really appreciate all the late night conversations, arguments and life experiences we have shared together and I am really looking forward to creating more memories together.

Last but not least, my supervisors from TNO, Chris van der Ploeg and Emilia Silvas, and my supervisor from the TU Delft, Laura Ferranti. Thank you so much for helping me through the past year, with the whole pandemic it was a bit of a struggle at the beginning but you guys pulled me through and helped me motivate and move forward each time. It is weird to have seen you more often on a screen than in real-life, but I am grateful for the conversations, discussions and virtual coffees together. You have taught me so much in the past year and pushed me to really understand every step that I was taking, something I am really grateful for and that I will take with me in my further career.

Niels Christiaan Lodder
Delft, March 2021

Abstract

With increased developments and interest in platooning and higher levels of automation (SAE level 3+), the need for safety systems that are capable of monitoring system health and maintain safe operation in faulty scenarios is increasing. Methods for the detection, isolation and identification of faults in automated and cooperative driving is increasing. Once the existence of a fault is known, one needs to classify its severity and decide between fail-operational and fail-safe mitigation to guarantee the safety of a faulty vehicle.

The considered scenario in this research consists out of a vehicle suffering from a severe fault, such as a power steering or rear tyre failure, whilst driving in an ACC string of vehicles on the right most lane of a highway. To accommodate failures in an automated vehicle, as a first contribution of this thesis a functional-safety architecture is proposed, which can enable safe operation in faulty scenarios. This architecture uses a nominal channel, a health monitor and a safety channel to incorporate all steps between nominal vehicle operation and fault mitigation. To demonstrate the increase in safety potential of the first contribution, its tactical decision making and fail-safe mitigation modules are implemented as a second contribution. The fail-safe mitigation uses an optimization-based algorithm to bring the faulty vehicle to a safe-state, being parked on the road shoulder. This manoeuvre is performed using nonlinear model predictive control (NMPC). To further highlight safety improvements of the functional-safety architecture, the prediction model of the NMPC is reconfigured. It uses the information from the fault detection and isolation module to optimize the tracking performance of the controller.

Assuming a string of ACC vehicles, results show different tactical decision making strategies the faulty-vehicle can perform to move to the road shoulder. The impact it has on the remainder of the string of vehicles shows a trade-off between stopping time and distance of the faulty vehicle and re-connection time for the remaining vehicles. Further results on the tracking performance of the NMPC show its robustness against severe faults and the increase in tracking performance when it uses the information from the proposed architecture. This highlights the safety improvement potential and need of both the functional-safety architecture and the fail-safe mitigation algorithm.

"If one does not fail at times,
then one has not challenged
himself"

– FERDINAND PORSCHE

Contents

Nomenclature

α_f	Front slip angle	rad
α_r	Rear slip angle	rad
Δt	Sampling time step	s
δ	Wheel angle	rad
$\dot{\delta}$	Rate of change in the wheel angle	rad/s
$\dot{a}_{x,c}$	Rate of change of the intended longitudinal acceleration	m/s^3
σ	Slack variable	
θ	Heading angle or yaw angle	rad
a_x	Longitudinal acceleration	m/s^2
a_y	Lateral acceleration	m/s^2
$a_{x,c}$	Intended longitudinal acceleration	m/s^2
$C_{\alpha f}$	Front cornering stiffness	N/rad
$C_{\alpha r}$	Rear cornering stiffness	N/rad
d_x	Longitudinal position	m
d_y	Lateral position	m
e_{tg}	Time gap error	s
f	State update function	
G_{dt}	Discrete time gain of the longitudinal dynamics	-
h_{dg}	Desired time gap	s
I_z	Inertia of the vehicle with respect to the center of gravity	$kg \cdot m^2$
J	Multi-objective cost function	
k	Discrete time step	
k_d	Derivative gain	-
k_p	Proportional gain	-
l_f	Distance from the center of gravity to the front axle	m
l_r	Distance from the center of gravity to the rear axle	m
m	Vehicle mass	kg
P	Prediction Horizon	
r	yaw rate	rad/s
S	Control Horizon	

s	Laplace variable	
s_{dt}	Discrete time pole of the longitudinal dynamics	-
$u(k)$	Control vector	
u_{PD}	Control output of the Proportional Derivative controller	
v_x	Longitudinal velocity	m/s
v_y	Lateral velocity	m/s
$w_{(...)}$	Cost function weights	
$x(k)$	State vector	
$z_{(...)}$	Reference values	

Abbreviations

ACC Adaptive Cruise Control. v, 13, 17, 28, 29

ADS Automated Driving System. 2

AFF Anomaly, Fault or Failure. 1, 2

ASIL Automotive Safety Integrity Level. 2, 3

CAD Cooperative and Automated Driving. 1, 3, 13

DDT Dynamic Driving Task. 2

FDD Fault Detection and Diagnosis. 2

FDI Fault Detection and Isolation. 2

FTC Fault Tolerant Control. 2

GLRT Generalised Likelihood Ratio Test. 2

GPS Global Positioning System. 2

IEEE Institute of Electrical and Electronics Engineers. 5

ISO International Organisation for Standardization. 2, 17

MPC Model Predictive Control. 2, 3, 15, 17, 28

NC Nominal Channel. 3

NMPC Nonlinear Model Predictive Control. v, 23

ODD Operational Design Domain. 2

SAE Society of Automotive Engineers. v, 1, 2

SC Safety Channel. 3

UIO Unknown Input Output. 2

List of Figures

List of Tables

1 Introduction

In 2017, the total European in-land freight transport via roads was 1.920.613 million tonne-kilometers [10], the majority of which is transported by trucks. As a consequence, trucks that travel long distances often drive large parts of their journey in convoy, thus behind other trucks. Previous research shows that road throughput could be increased when trucks would drive at a smaller time-gap to each other [18] and fuel consumption (thus CO_2 emissions) can decrease up to 20% subsequently [17]. To achieve this, cooperative and automated driving (CAD) systems are being developed in joint projects like SARTRE [26] and ENSEMBLE [16], but also by individual companies like TNO [22, 25, 32]. Platooning is a CAD system that aims to maintain a constant time-gap between two vehicles, using connectivity technology and automated driving support systems [9]. Implementing this shows the effects of platooning on reducing workload and stress of the drivers [7], by reaching higher levels of automation. SAE levels define these levels of automation and vary from zero to five as explained in Figure 1.1.

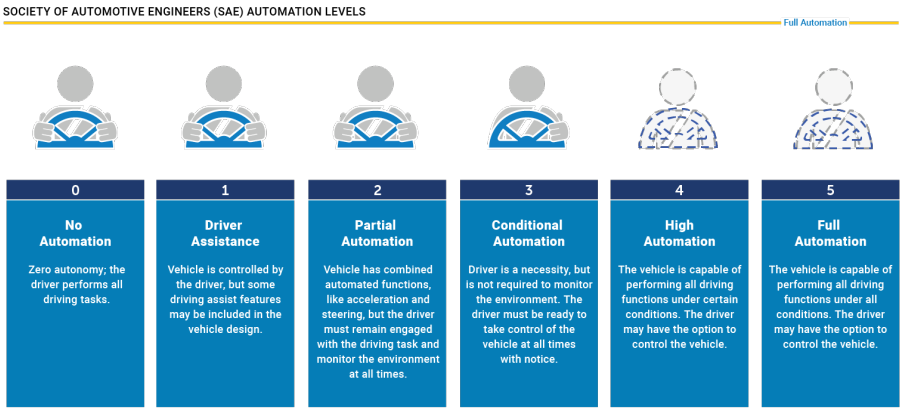

Figure 1.1: Explanation of the SAE levels [3]

1.1. Related Work

Ensuring safety for SAE level 3+ systems requires architectures that contain *health monitoring and management*, *safety-channels* and *fallback functionalities* [14]. In order to monitor the health of a system, anomalies, faults and failures (AFFs) should be detected and assessed during system operation. Furthermore, the AFFs should be mitigated depending on its severity, for which safety architectures are required.

1.1.1. System health monitoring

Within a health monitor, the system checks if there are any AFFs present, using diagnostic techniques. The survey [13] provides an overview of techniques that can be used to diagnose a fault, which implies three steps:

1. Detection
2. Isolation
3. Identification

These three steps are referred to as Fault Detection and Diagnosis (FDD) or Fault Detection and Isolation (FDI) and can use different approaches, being Data-driven, Model-, Signal- or Knowledge-based. Model- and Signal-based approaches are most suited for autonomous vehicles [15] due to the known vehicle models and varying signals that are monitored within a vehicle.

Fault detection can make use of a filter to determine if there is a fault, for example in [6], where an Unknown Input Output (UIO) filter and a $\mathscr{H}_-/\mathscr{H}_\infty$ filter are combined, or by making use of a residual generator, as done in [31], in which a polynomial matrix based residual generator creates the residual. This residual is nonzero when a fault is detected and zero when the system is in a healthy state.

Isolation finds the location of the fault by statistical techniques like a Generalised Likelihood Ratio Test (GLRT), as proposed in [6], but also by norm-based algorithms, e.g. adaptive or constant thresholding of the norm of the residual. A filter could also be used here, as demonstrated in [31], by implementing a moving-horizon least-squares filter with a n-steps moving horizon filter to isolate additive or multiplicative AFFs.

Following the extensive survey [13], fault identification uses advanced observer techniques such as Proportional (Multiple-) Integral P(M)I observers, adaptive observers, sliding mode observers or descriptor observers to establish the type, shape and size of the fault.

The severity of the concerned fault can be determined by comparing the outcome of the FDI or FDD to tailored thresholds, by using decision trees, or other advanced algorithms. Severity levels in automotive systems are categorized following ISO26262, by assigning an Automotive Safety Integrity Level (ASIL) to the AFF, which vary between A and D, A being the least severe and D being the most severe.

1.1.2. Mitigation

Once a severe fault is diagnosed, it is of foremost importance to bring the system to a safe state (i.e., make use of a safe and efficient fallback strategy). ISO26262 states that vehicles operating at SAE level 3 or higher, all Dynamic Driving Tasks (DDTs) should be performed by the Automated Driving System (ADS) [2]. This includes DDT fallback and achieving a minimal risk condition in case of a performance-relevant system failure or upon exiting the Operational Design Domain (ODD), for SAE levels 4 and 5 [1]. This implies that the vehicle should, without interference of a human driver, bring itself to a minimal risk condition when a failure within the vehicle occurs that makes nominal operation impossible (e.g., a brake system failure).

To mitigate the diagnosed AFFs, [13] briefly discusses fault tolerant control (FTC) strategies , where the system performance is maintained in the presence of faults, yet no real connection is made between the diagnosis and what mitigation measures should be taken. Similarly in [35], fault tolerant cooperative control is introduced, focusing on the mitigation strategies. All current work focuses on single mild faults, that require a limp home mode or degraded functionality and so far, there is no end-to-end system for both diagnosis and mitigation of faults of different types. What literature still is missing is a direct connection between the detection and mitigation of AFFs.

Research on performing mitigation strategies when a severe AFF occurs in an automated vehicle, e.g. a fallback strategy, is scarce. To this end, [29] focuses on trajectory planning in fallback scenarios by formulating the problem as an optimal control problem, without considering any faults. The authors of [34] describe an adaptive model predictive control (MPC) algorithm to simultaneously avoid potential collisions with surrounding vehicles and handle the presence of a front perceptive sensor failure. Yu and Luo propose in [37] a fallback strategy to park on the road shoulder, while having a loss of all redundant paths or GPS location. They plan a trajectory using a 5^{th} or 6^{th} degree polynomial, depending on the presence of obstacles. However, they decouple the longitudinal and lateral control of the vehicle, which might hinder the safe vehicle movement towards the road shoulder, especially in high risk scenarios. For their control the authors use a sliding mode controller and preview model for the motion control. Furthermore, in all previous works no failures are considered that influence the handling of the vehicle.

1.1.3. Safety Architectures

The safety mechanisms and architectures designed for CAD should be able to transition and bring a vehicle to a safe state, i.e. an operating mode with a reasonable level of risk. To address and accommodate the steps mentioned in Section 1.1.2, the authors of [19] proposed an architecture pattern with a safety channel suitable for automated driving applications and ASIL D, which is the highest risk class. In this work, the safety channel is divided into a health channel and limp home channel. However, it does not specify the functionalities and methods that should be used in these channels. Törngren et al. propose an architecture which includes a supervisor channel (SC) to the nominal channel (NC) in order to asses the states of the system in [30]. This proposed SC is capable of detecting errors within the NC and mitigate some errors, however the authors raise the question if the SC can also mitigate hazardous events caused by performance limitations.

1.2. Research objectives

To address some of the challenges which appeared in the literature, this research proposes two contributions:

1. *Functional Safety Architecture*
   The functional safety architecture accommodates fault diagnosis as well as mitigation for automated driving applications and aims to facilitate all required steps from nominal operation to bringing the vehicle to a safe-state in case of severe failures.

2. *Fail-safe mitigation algorithm*
   The fail-safe mitigation algorithm introduces a MPC-based controller with coupled longitudinal and lateral dynamics, to control a vehicle affected by two separate (severe) failures. This mitigation algorithm performs a fallback strategy to park a vehicle on the road shoulder, thus bringing it to a safe-state.

1.3. Document outline

The remainder of this research is structured as follows:

- Chapter 2 presents the contributions made in this research in paper format, organised as follows:
  - Section I introduces the problems that are solved and highlights relevant literature.
  - Section II describes the main components of the proposed architecture which enables nominal and fallback functionalities for an automated vehicle.
  - Section III provides the fail-safe mitigation algorithm.
  - Section IV presents the simulation results for various scenarios and fault severity levels.
  - Section V finally gives the conclusions and recommendations in brief.
- Appendix A describes the fail-safe mitigation algorithm in depth, such as the choices for the constraints and limits as well as the tuning process of the controller.
- Appendix B shows additional results and limit handling of the controller.
- Appendix C concludes on the thesis and provides a discussion on the work performed and future work to be done.

2

Conference paper

The content of this chapter has been submitted to the 32nd IEEE Intelligent Vehicles Symposium.

Optimization-based Fault Mitigation in Automated Driving

Niels Lodder[1,2], Chris van der Ploeg[2,3], Laura Ferranti[1], Emilia Silvas[2,3]

***Abstract*— With increased developments and interest in platooning and higher levels of automation (SAE level 3+), the need for safety systems that are capable to monitor system health and maintain safe operations in faulty scenarios is increasing. Methods for detection, isolation and identification of anomalies, faults and failures in automated and cooperative driving is increasing. Once the existence of a fault is known, there is a need to classify its severity and decide on appropriate and safe mitigating actions. To provide a solution to this challenge, in this paper a functional-safety architecture is proposed and an optimization-based mitigation algorithm is introduced. This algorithm uses nonlinear model predictive control (NMPC) to bring a vehicle, suffering from a severe fault, such as rear tire or power steering failure, to a safe-state. The internal model of the NMPC uses the information from the fault detection, isolation and identification to optimize the tracking performance of the controller, showcasing the need of the proposed architecture. Assuming a string of ACC vehicles, results show different tactical decision making strategies that the faulty-vehicle can use in handling the faults, what is the safety improvement potential and the impact these strategies have on the duration of the manoeuvres.**

***Index Terms*—Functional Safety, Operational Safety, Model Predictive Control, Fault Mitigation, Fail-safe**

I. INTRODUCTION

Cooperative and automated driving (e.g. platooning) has been widely researched in the past decades, showing its effects on reducing workload and stress of the drivers [1], but also on society. Driving in a platoon can increase road throughput by driving at closer distances [2] and can reduce fuel consumption (and therefore CO_2 emissions) up to 20% [3]. For both cooperative and automated driving (CAD), ensuring safety for higher levels of automation requires architectures that contain health monitoring and management, safety-channels and fallback functionalities [4], [5]. The safety mechanisms designed for CAD should be able to transition and bring a vehicle to a *safe state*, i.e. an operating mode without an unreasonable level of risk. In addition, vehicles operating in SAE level 4 or 5 should be able to automatically reach a minimal risk condition in case of a performance-relevant system failure [6]. This implies that the vehicle should, without interference of a human driver, bring itself to a minimal risk condition when a failure within the vehicle occurs that makes nominal operation impossible (e.g., a brake system failure).

To address the concerns above, the authors of [7] proposed an architecture pattern with a safety channel suitable for automated driving applications and Automotive Safety Integrity Level (ASIL) D, which is the highest risk class. In this work, the safety channel is divided into a health channel and limp home channel. However, it does not specify the functionalities and methods that should be used in these channels. The survey [8] provides an overview of methods that can be used to *diagnose* a fault, which implies three steps: (i) detection, i.e. determining whether there is a fault, (ii) isolation, i.e. the location of the fault; (iii) identification, i.e the type, shape and size of the fault. Furthermore, [8] briefly discusses fault tolerant control (FTC) strategies, where the system performance is maintained in the presence of faults, yet no real connection is made between the diagnosis and what mitigation measures should be taken. Similarly, in [9] fault tolerant cooperative control is introduced, focusing on the mitigation strategies. All current work focuses on single mild faults, that require a limp home mode or degraded functionality and so far, there is no end-to-end system for both diagnosis and mitigation of faults of different types.

Once a severe fault is diagnosed, it is of foremost importance to bring the system to a safe state (i.e., make use of a safe and efficient fallback strategy). To this end, [10] focuses on trajectory planning in fallback scenarios by formulating the problem as an optimal control problem, without considering any faults. The authors of [11] describe an adaptive model predictive control (MPC) algorithm to simultaneously avoid potential collisions with surrounding vehicles and handle the presence of a front perceptive sensor failure. Yu and Luo propose in [12] a fallback strategy to park on the road shoulder, while having a loss of all redundant paths or GPS location. However, they decouple the longitudinal and lateral control of the vehicle, which might hinder the safe vehicle movement towards the road shoulder, especially in high risk scenarios. Furthermore, in all previous works no failures are considered that influence the handling of the vehicle.

To address some challenges which appeared in the literature, a first contribution of this paper is a proposed functional safety architecture that accommodates fault diagnosis as well as mitigation for automated driving applications. The architecture aims to facilitate all required steps from nominal operation for bringing the vehicle to a safe-state in case of severe failures. Fig. 1 shows the example scenario considered, where in a string of automated vehicles, running in nominal conditions, one detects a fault and needs to automatically park itself on the road shoulder.

The second contribution of this paper is focusing on a vehicle affected by two separate failures, for which a MPC-based fail-safe mitigation algorithm is introduced with

[1] Department of Cognitive Robotics, Delft University of Technology, 2628 CD Delft, The Netherlands
[2] TNO - Integrated Vehicle Safety, 5708 JZ Helmond, The Netherlands
[3] Department of Mechanical Engineering, Eindhoven University of Technology, 5612 AZ Eindhoven, The Netherlands

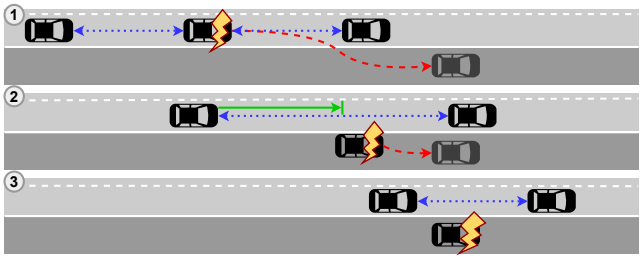

Fig. 1: Scenario description where a severe fault occurs in a string of automated vehicles

coupled longitudinal and lateral dynamics. Within this scenario, two mitigation strategies are investigated for the faulty vehicle to showcase the influence on the remainder of the string of vehicles: (i) The vehicle will brake inside the current lane, starting from the point that it receives the instruction to park on the road shoulder, and (ii) The vehicle will brake outside of the current lane, starting from the point that it has left the active lane.

This paper is organised as follows. Section II introduces the main components of the proposed architecture which enables nominal and fallback functionalities for an automated vehicle. Section III introduces the fail-safe mitigation algorithm and Section IV presents the simulation results for various scenarios and fault severity levels. Finally, conclusions and recommendations are described in Section V.

II. FUNCTIONAL SAFETY ARCHITECTURE

To ensure safe and comfortable operations, an automated vehicle architecture consists of three parts, namely, a nominal channel, a health monitor and a safety channel [5]. We propose here the architecture shown in Fig. 2, which is based on [5], and extended to accommodate different faults and/or different severity levels. Herein, the nominal channel performs all the nominal vehicle operation. The health monitor continuously monitors data coming from the vehicle to check whether this is operating in a healthy state, and the safety channel accommodates fail-safe mitigation to bring the vehicle to a safe-state when needed.

A. Nominal Vehicle Operation

Nominal vehicle operation refers to the operation of the vehicle under normal circumstances, that is, in the absence

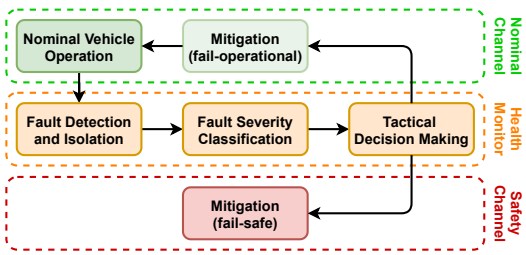

Fig. 2: Architecture approach for nominal and safety fallback functionality of an automated vehicle, including failures

of anomalies, faults or failures (AFFs) and driving within its operational design domain. In these conditions, the system can make use of all its functionalities and ensure safe vehicle control.

B. Fault Detection and Isolation

In order to asses AFFs, at first its presence and location should be known. This is done by respectively the detection and the isolation, where the detection solely focuses on the presence of an AFF. Subsequently, the isolation then determines the location of the AFF. Finally, the identification determines its type, shape and size, using advanced observer techniques such as Proportional (Multiple-) Integral observers, adaptive observers, sliding mode observers or descriptor observers [8].

C. Fault Severity Classification

The risk of the diagnosed fault can be classified using ASIL levels and safety channel hardware, to determine if it is safe for the vehicle to continue driving. If it is not safe for the vehicle, the module determines whether the vehicle can continue with degraded functionality or whether it should go to a safe-state.

D. Tactical Decision Making

In literature, this module is implemented both on a single- and multi vehicle level, if the vehicle has vehicle-to-vehicle communication and can drive in cooperative modes (e.g., platooning [13]). Tactical decision making is usually a needed nominal functionality that also contains a health monitoring and management component. In CAD, by using this module, the integrity of a string of vehicles can be maintained while, for example, one of the vehicles abruptly leaves the string of vehicles. In the context of the scenario described in Fig. 1, a benefit of this module is that the behaviour of the Lead Vehicle (LV) can be influenced such that the Trailing Vehicle (TV) can reconnect to the LV while optimizing certain parameters (e.g., fuel consumption).

E. Mitigation

Reducing the effect of an AFF is referred to as *mitigation*. Anomalies can lead to faults and consequently to failures, which are undesirables and potentially unsafe. In the context of a failure, i.e. a termination of an intended behaviour of an element or an item due to a fault manifestation [4], handling this failure means controlling the system in its presence. Depending on the outcome of the Fault Severity Classification (FSC) module, the strategy for the mitigation is chosen to be fail-operational or fail-safe.

a) Fail-operational: When the FSC module determines that the vehicle can safely continue operation, possibly with reduced functionality (also referred to as degraded or limp functionality), fail-operational mitigation is performed. Such mitigation is most commonly performed by FTC if the AFF concerns an actuator or process [9]. As exemplified in [14], FTC converts the system to be less or not at all dependent on the faulty component, using the information acquired in the health monitor.

b) Fail-safe: In case the FSC module determines that the vehicle is in a non-healthy state and cannot guarantee safe operation, fail-safe mitigation is performed by initiating a fallback manoeuvre to bring the vehicle to a safe-state. Similar to fail-operational mitigation, the information acquired in the health monitor is used.

III. FAIL-SAFE MITIGATION ALGORITHM

To describe the fail-safe mitigation algorithm proposed in this paper, we start from the scenario described in Fig. 1. Herein, three vehicles are assumed to drive automatically on the road (with functionality such as adaptive cruise control and lane keep assist active, i.e. the *nominal* functionality). As shown in Fig. 3, once a severe fault is occurring, the faulty vehicle needs to transition to a safe state with the help of its safety channel. The ACC-based longitudinal controller is ensuring a constant time-gap inter vehicle distance, with the following error dynamics

$$e_{tg} = h_{dg} - \frac{d_{x,i-1} - d_{x,i}}{v_{x,i}}, \tag{1}$$

where e_{tg} represents the time gap error between the two vehicles, h_{dg} indicates the desired time gap between the two vehicles, $d_{x,i-1} - d_{x,i}$ is the distance between the preceding vehicle and the ego vehicle, and $v_{x,i}$ is the ego vehicle velocity.

This error is controlled by a Proportional Derivative (PD) controller, often used in literature, as introduced in [15] with the following control law formulated, in the Laplace domain:

$$u_{PD} = e_{tg}(k_p + k_d s) \tag{2}$$

With u_{PD} the control output, k_p the proportional gain and k_d the derivative gain of the control law.

To ensure safe handling of the faulty vehicle, both longitudinal and lateral control are immediately taken over by the safety channel after AFF diagnosis. Without loss of generality, we assume here the faults are already detected and classified and focus on the Tactical Decision Making (TDM) and Fail-Safe Mitigation (FSM) modules from Fig. 2. The implemented TDM is explained in Section III-A and the controller used in FSM is explained in Section III-B.

A. Implemented Tactical Decision Making

Fig. 4 shows the implemented TDM module, in which the FSC module gives a message to the TDM module when the failure is classified and the vehicle should be parked on the road shoulder. The environmental module gives input that determines if the vehicle should brake in-, or out-of-lane,

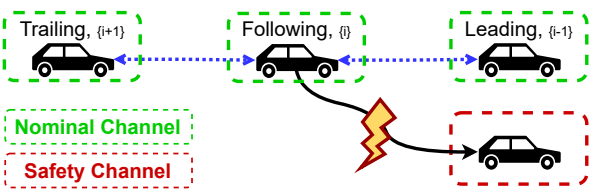

Fig. 3: Multiple ACC-driven vehicles of which one encounters a severe fault and needs to reach a safe state.

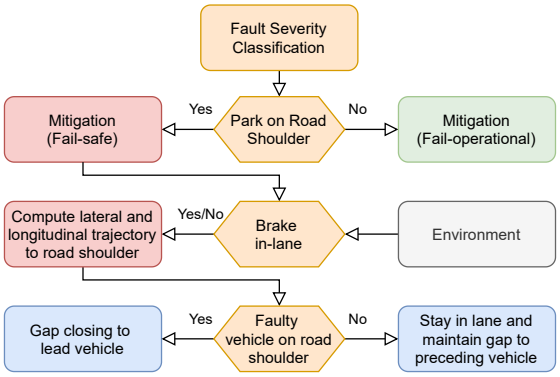

Fig. 4: Flow chart of implemented vehicle Tactical Decision Making (hexagons) and its effects on the trailing vehicle (blue blocks).

e.g. if the road shoulder is long enough to brake out-of-lane, otherwise brake in-lane is required. Eventually, the TDM module sends a message to the TV when it should close the gap back to the LV.

B. Functional Safety Mitigation Controller

MPC is often used to generate optimal control commands for the vehicle, [16], [17], by taking into account the vehicle dynamics and its limitations over a predefined time window, i.e. prediction horizon. A Nonlinear MPC (NMPC) performs the control in the safety channel and is required because of the combined longitudinal and lateral dynamics, following the continuous-time equations of the linear single-track dynamic bicycle model [18]:

$$\begin{aligned}\dot{v}_y(t) = &-\frac{C_{\alpha f} + C_{\alpha r}}{m v_x(t)} v_y(t) + \\ &\left(\frac{l_r C_{\alpha r} - l_f C_{\alpha f}}{m v_x(t)} - v_x(t)\right) r(t) + \frac{C_{\alpha f}}{m}\delta(t)\end{aligned} \tag{3}$$

$$\begin{aligned}\dot{r}(t) = &\frac{l_r C_{\alpha r} - l_f C_{\alpha f}}{I_z v_x(t)} v_y(t) - \\ &\frac{l_f^2 C_{\alpha f} + l_r^2 C_{\alpha r}}{I_z v_x(t)} r(t) + \frac{l_f C_{\alpha f}}{I_z}\delta(t)\end{aligned} \tag{4}$$

Where $C_{\alpha f}$ and $C_{\alpha r}$ are the respective front and rear cornering stiffness, m is the vehicle mass, l_f and l_r are the length from the front and rear axles to the center of gravity respectively, I_z is the vehicle's moment of inertia and finally, v_x, v_y, r represent the longitudinal velocity, lateral velocity and yaw rate, respectively.

The proposed MPC design requires a discrete-time update model, thus Equations (3) and (4) are discretized using the forward Euler method, to form the nonlinear state-update equations (Equation (6)), from the state vector $x(k)$:

$$x(k) = [a_x(k)\, v_x(k)\, v_y(k)\, d_y(k)\, r(k)\ \theta(k)]^{\mathrm{T}}, \tag{5}$$

where a_x, d_y and θ are the longitudinal acceleration, lateral

position and heading angle, respectively.

$$a_x(k+1) = s_{dt}a_x(k) + G_{dt}a_{x,c}(k) \tag{6a}$$
$$v_x(k+1) = v_x(k) + a_x(k)\Delta t \tag{6b}$$
$$v_y(k+1) = v_y(k) + \Delta v_y(k)\Delta t \tag{6c}$$
$$d_y(k+1) = d_y(k) + \big(v_y(k)\cos\big(\theta(k)\big) + v_x(k)\sin\big(\theta(k)\big)\big)\Delta t \tag{6d}$$
$$r(k+1) = r(k) + \Delta r(k)\Delta t \tag{6e}$$
$$\theta(k+1) = \theta(k) + r(k)\Delta t, \tag{6f}$$

Here s_{dt} and G_{dt} are respectively the discrete time pole and gain of the first-order transfer function representing the longitudinal dynamics, $a_{x,c}$ is the intended longitudinal acceleration, δ is the front wheel angle, Δv_y and Δr are the increments in v_y and r, Δt denotes the sampling time step and the indicator k denotes the discrete time step.

We can rewrite the model above in a more compact notation as follows:

$$x(k+1) = f(x(k), u(k)), \tag{7}$$

Where $u(k) := [a_{x,c},\, \delta]^{\mathrm{T}}$ is the control vector. The NMPC is formulated as

$$\min_{u} \sum_{k=1}^{N} J\left(x(k), u(k), z(k)\right) \tag{8a}$$
$$\text{s.t.} \quad x(k+1) = f\left(x(k), u(k)\right) \tag{8b}$$
$$x_{\min} \le x(k) \le x_{\max} \tag{8c}$$
$$u_{\min} \le u(k) \le u_{\max} \tag{8d}$$
$$\Delta u_{\min} \le \frac{u(k+1) - u(k)}{\Delta t} \le \Delta u_{\max} \tag{8e}$$
$$a_{y,\min} \le a_y(k) \le a_{y,\max} \tag{8f}$$
$$x(0) = x_{\text{init}} \tag{8g}$$
$$\forall k \in \{0, \ldots, N\}, \tag{8h}$$

where $z(k)$ contains the reference from the trajectory generation and J represents the multi-objective cost function:

$$J\Big(x(k), u(k), z(k)\Big) = w_{v_x}\big(z_{v_x}(k) - v_x(k)\big)^2 + w_{d_y}\big(z_{d_y}(k) - d_y(k)\big)^2 + w_\theta\big(z_\theta(k) - \theta(k)\big)^2 + w_{a_x}\big(a_{x,c}(k)\big)^2 + w_\delta\big(\delta(k)\big)^2 + \sigma \tag{9}$$

In which $w_{(\ldots)}$ are the respective weights and σ is a slack variable. Constraint (8b) indicates the dynamic coupling and constraints (8c), (8d) and (8e) indicate comfort and model limitations. Within which δ, $\dot{\delta}$ and $\dot{a}_{x,c}$ are based on the physical capabilities of the vehicle and limits of the dynamic bicycle model. The constraints on a_x, $a_{x,c}$, a_y and v_x are based on maximum allowed ACC braking, according to ISO 15622, comfort and highway speed limit respectively. The lateral acceleration a_y in (8f) is calculated by the following steady-state relation (imposed as a comfort constraint):

$$a_y = -\frac{C_{\alpha f} + C_{\alpha r}}{m v_x} v_y + \frac{l_r C_{\alpha r} - l_f C_{\alpha f}}{m v_x} r + \frac{C_{\alpha f}}{m}\delta \tag{10}$$

IV. SIMULATION RESULTS

For this simulation study, a string of vehicles is considered as depicted in Fig. 3, where all vehicles are modelled using the parameters given in Table Ia. These parameters correspond to a lab passenger vehicle available at TNO [1], used for research on cooperative and automated driving technologies. The constraint values used in the NMPC model are given in Table Ib.

The trajectory which the faulty vehicle will follow during the fail-safe mitigation is split into lateral and longitudinal movement, to best accommodate both our mitigation strategies. The lateral trajectory is generated by a 5$^{\text{th}}$ order polynomial, taken between current and goal waypoints with appropriate heading angles, following [12]. For the longitudinal trajectory only goal velocities are given, such that the controller determines the optimal control outputs within the given constraints, considering all relevant dynamics. Alternatively, as part of our future work, a local motion planner can also be incorporated in our architecture to adapt the trajectory online to avoid collisions with upcoming traffic (e.g., [19]).

The vehicle model that is used as a plant, to test the controller, is based around the continuous time counterparts in (6).

A. Controller settings

The tuning parameters for the Proportional Derivative (PD) controllers performing the longitudinal control for the ACC string of vehicles and the NMPC controller that performs the fallback manoeuvre are given in Tables II and III, respectively.

Table II shows the tuning parameters of each vehicle, where the LV is tuned differently compared to the FV and TV, as it is operating in cruise control and tracking a reference velocity instead of a time-gap to the preceding vehicle.

[1] https://www.tno.nl/en/focus-areas/traffic-transport/expertise-groups/research-on-integrated-vehicle-safety/

TABLE I: (a) Vehicle parameters; (b) Constraints used in the NMPC model

Parameter	value	unit
$C_{\alpha f}$	120	kN/rad
$C_{\alpha r}$	220	kN/rad
l_f	1.33	m
l_r	1.47	m
m	1845	kg
I_z	3580	$kg \cdot m^2$

(a)

Variable	Constraint (min / max)	unit
$\mid \delta \mid$	0.0873	rad
$\mid \dot{\delta} \mid$	0.0818	rad/s
a_x	-3.5 / 1.5	m/s^2
$a_{x,c}$	-3.5 / 1.5	m/s^2
$\dot{a}_{x,c}$	-14 / 6	m/s^3
v_x	1.26 / 33	m/s
$\mid a_y \mid$	2	m/s^2

(b)

TABLE II: Settings of PD controllers within each vehicle

Vehicle	k_p	k_d
Leading	5	0.3
Following / Trailing	-150	-2.5

TABLE III: Settings of NMPC used for the fallback manoeuvre

Variable	P	S	w_{v_x}	w_{d_y}	w_θ	w_{a_x}	w_δ
Value	30	30	10	100	1	0.5	1

From the dynamic bicycle model in (3) and (4) it can be derived that, as the velocity decreases towards zero, the eigenvalues of the linear differential equations grow towards $-\infty$. This phenomenon is numerically impossible to capture in the Forward Euler approximation used in this paper, as it would require the sampling time to be reduced to 0. Following this line of reasoning, to prevent numerical instability of the internal prediction model, a sampling time of $0.01\,s$ and a $v_{x,\min}$ of $1.26\,m/s$ is selected. The optimal selection of prediction, control horizons and the NMPC weights in the cost function are chosen as a trade-off between computational effort and tracking performance, aiming for lowest computational effort with minimal loss in tracking performance.

B. Failure scenarios and braking strategies considered

We present six simulation results: (i) two simulations compare braking in-lane and braking out-of-lane during the fallback manoeuvre, (ii) two simulations investigating robustness of the controller by implementing realistic failures in the vehicle model and (iii) two simulations investigating the behaviour of the controller if it is reconfigured, following the architecture proposed in Section II, adjusting relevant formulas and bounds.

The following failures are considered for (ii) and (iii):

Power steering failure: The steering output of the controller is decreased by 50% before it feeds through to the vehicle model.

Rear tyre failure: The rear cornering stiffness $C_{\alpha r}$ of the vehicle is decreased by 50% to mimic that one of the rear tyres has failed [20].

C. Results

To show the performance of the proposed method in bringing the vehicle to a safe state, two time moments are important, t_a, when the parking manoeuvre is initiated, and t_b, when the faulty vehicle has left the initial driving lane.

Braking in-lane versus braking out-of-lane

Table IV highlights the trade-off between the two mitigation strategies on the basis of stop time and distance versus re-connection time of the remaining vehicles on the road (TV to the LV). The stop time is calculated as the time between t_a and the time that the error on the goal velocity is less or equal to $0.01\,m/s$ and the error on the lateral position is less or equal to $0.001\,m$. The travelled distance between these two instances is the stopping distance. Re-connection time is calculated as the time between the instance that e_{tg} is larger than $0.4\,s$ and the instance that the e_{tg} stays below $0.01\,s$. Stopping time and distance is largely influenced by $a_{x,\min}$ as the lateral movement consumes less time compared to the longitudinal movement. Furthermore, the duration of the lateral movement has a major impact on the difference in closing time due to the distance and velocity difference it creates between both strategies.

TABLE IV: Comparison between braking strategies while going to road shoulder and the effect on upcoming traffic.

Braking mitigation strategy	Stop time [s]	Stop distance [m]	Trailer gap-closing time [s]	Timegap error at t_b [s]
In-lane	8.208	117.534	13.880	1.650
Out-of-lane	10.838	190.610	7.634	1.004

As expect, the timegap error at t_b shows that braking in-lane (BIL) results in a higher time-gap than braking out-of-lane (BOL) and therefore a longer closing time for BIL compared to BOL. This is underpinned by the velocity difference between the TV and LV at t_b and the acceleration length in Fig. 5. The lateral deviation in both strategies is equal, following Fig. 6, however the steering outputs show different behaviour in both strategies. This, helped by the decreased longitudinal velocity because of braking, translates into an increased yaw rate in the vehicle dynamics for BIL compared to BOL.

As BIL results in higher lateral loads on the vehicle dynamics, this strategy is used in further experiments and as a baseline comparison. Figs. 7 and 8 show the error difference between input/states the baseline (BIL without failure) and the input/states of the subsequent failure.

Robustness of the controller

Following the results in Fig. 7, the steering failure causes the controller to output higher steering inputs for the vehicle. Next to that, steering is less smooth and shows more abrupt changes in direction, caused by reaching the limit of the steering rate $\dot{\delta}$. This is also clearly visible in the yaw rate r, showing its influence on the lateral vehicle dynamics.

Furthermore, the results show that the flat tyre decreases the maximum controller setpoint and makes the initially

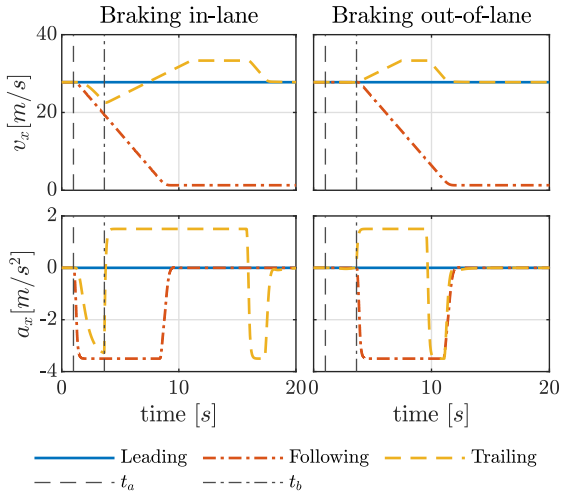

Fig. 5: Longitudinal velocities v_x and accelerations a_x during the lane changing strategies for all vehicles.

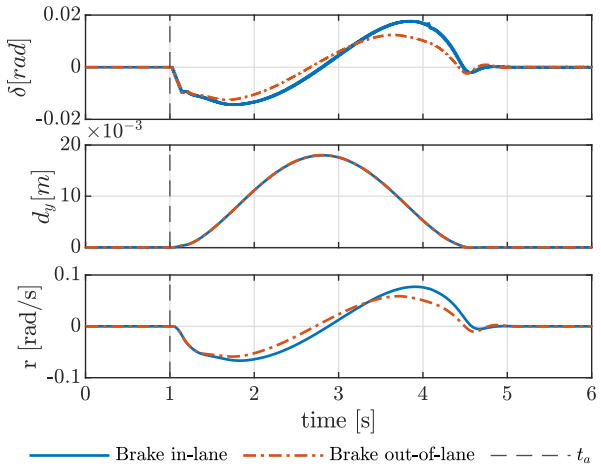

Fig. 6: Comparison between both mitigation strategies on steering output δ, lateral position d_y and yaw rate r

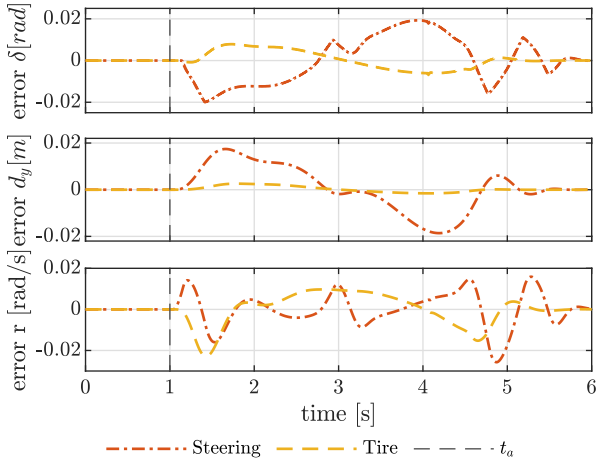

Fig. 7: Error plots of the controller with a steering failure and rear tire failure on steering output δ, lateral position d_y and yaw rate r compared to the baseline.

understeered vehicle show oversteered behaviour. The latter translating into the vehicle turning more compared to the baseline with the same steering input. This effect is also observed in the lateral position error, where the vehicle initially steers too much, thus deviates further from the path.

Reconfiguration of the controller

The reconfigured controller uses the information on the failures (Section IV-B) to update the internal NMPC model (Equation (6)). For the steering failure this means that $\delta(k)$ is transformed into $0.5\delta(k)$. Also, the bounds on δ and $\dot{\delta}$ are increased by a factor $\frac{1}{0.5}$. In case of the rear tire failure, $C_{\alpha r}$ is changed to half of its original value. Fig. 8 shows the results, in which the lateral control action is smoother for the steering failure but similar for the tire failure, compared to the non-reconfigured controller in Fig. 7. The magnitude of the steering output is comparable for both failures in relation to the non-reconfigured simulations.

The steering output and yaw rate of the tire failure can be compared with its non-reconfigured result, however the

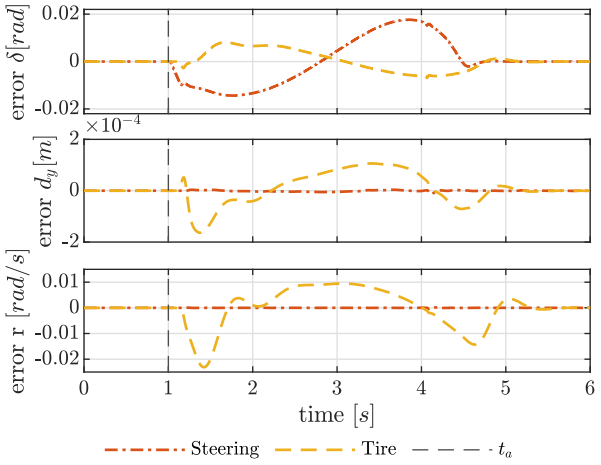

Fig. 8: Error plots of the reconfigured controller with a steering failure and rear tire failure on steering output δ, lateral position d_y and yaw rate r compared to the baseline.

maximum lateral deviation error is decreased by 92% for the reconfigured controller. For the steering failure, the error on lateral deviation is decreased to under $0.013\,mm$, a decrease up to 33%, and the yaw rate error to a maximum of $0.00037\,rad/s$, effectively eliminating the effect of the failure on tracking performance.

When evaluating all figures and Table IV, it is clear that the controller is capable of handling a power steering failure or a rear tyre failure. Especially when re-configuring the NMPC model, the performance is comparable to that of the system without a failure. Furthermore, as BOL results in a lower gap-closing time for the TV, thus disrupts the surrounding vehicles less than BIL, and results in lower dynamic loads thus higher comfort, it is recommended to use this mitigation strategy if the environment of the vehicle allows this.

V. CONCLUSIONS

The contributions of this research focuses on introducing a functional safety architecture that can handle multiple types of faults, the strategy and the fail-safe mitigation algorithm to park the vehicle on the road shoulder in case of severe failures. This enhanced architecture is needed to enable higher levels of automation and shows the need of health monitoring and safety channels to ensure safe operation when a failure occurs.

Our fail-safe mitigation strategy (tactical decision making and motion control) relies on a finite state machine and a tailored MPC formulation, controlling the lateral and longitudinal movement of the vehicle simultaneously. The results, shown for two severe failures (a power steering failure and a flat rear tire), highlight the trade-offs for different lane changing strategies for the faulty vehicle, i.e. braking in- and out-of-lane, and for the other vehicle in upcoming traffic. Furthermore, results also show that if the controller can have failure-awareness it can adapt and performance can be improved.

As future work we plan to validate our proposed architecture and fail-safe mitigation algorithm also through experi-

ments, to verify it using more scenarios and by incorporating the other needed components (such as fault diagnosis and severity classification).

REFERENCES

[1] D. D.Heikoop et al. "Effects of platooning on signal-detection performance, workload and stress: A driving simulator study", *Applied Ergonomics 60*, pp.116-127, 2017

[2] J. Lioris et al. "Platoons of connected vehicles can double throughput in urban roads", *Transportation Research Part C: Emerging Technologies*, vol. 77, pp. 1051-1061, 2016

[3] K. Y.Liang et al. "Heavy-Duty Vehicle Platoon Formation for Fuel Efficiency", *IEEE Trans. on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1051-1061, 2016.

[4] "ISO/DIS 26262-1: Road vehicles - Functional Safety", Geneva, Switzerland: International Organization for Standardization, 2018

[5] Khabbaz Saberi, A. et al. "An approach for functional safety improvement of an existing automotive system", *IEEE Int. Systems Conference*, pp. 277, 2015

[6] "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles", 2016

[7] Y. Luo et al. "An architecture pattern for safety critical automated driving applications: Design and analysis." *IEEE International Systems Conference7*, pp. 1–7, 2017

[8] Z. Gao, C. Cecati and S. X.Ding, "A survey of fault diagnosis and fault-tolerant techniques-part II: Fault diagnosis with knowledge-based and hybrid/active approaches", *IEEE Trans. on Industrial Electronics*,vol. 62, no. 6, pp. 3768-3774, 2015

[9] H. Yang et al. "Fault-tolerant coorperative control of multiagent systems: A survey of trends and methodologies", *IEEE Trans. on Industrial Informatics 16.1*, vol. 16, no. 1, pp. 4-17, 2020

[10] L. Svensson et al. "Safe Stop Trajectory Planning for Highly Automated Vehicles: An Optimal Control Problem Formulation", *IEEE Intelligent Vehicles Symposium*, pp. 517-522, 2018

[11] W. Xue et al. "A Fallback Approach for an Automated Vehicle Encountering Sensor Failure in Monitoring Environment" *IEEE Intelligent Vehicles Symposium*, pp. 1807-1812, 2018

[12] J. Yu and F. Luo, "Fallback Strategy for Level 4+ Automated Driving System", *IEEE Intelligent Transportation Systems Conference*, pp. 156-162, 2019

[13] L. Konstantinopoulou et al. "Specifications for Multi-Brand Truck Platooning", *ICWIM8, 8th Int. Conference on Weigh-In-Motion*, 8-p, 2019

[14] M. Khalili et al. "Distributed adaptive fault-tolerant control of uncertain multi-agent systems", *Automatica 87*, vol. 87, pp. 142-151, 2018

[15] G.J.L. Naus et al., "String-Stable CACC Design and Experimental Validation: A Frequency-Domain Approach", *IEEE Trans. on Vehicular Technology.*, vol.59, no.9, pp.4268-4279, 2010

[16] J.M. Maciejowski. "Predictive control: with constraints", *Pearson education*, 2002.

[17] E. van Nunen et al, *IEEE Int. Conference on Intelligent Transportation Systems*, Robust model predictive cooperative adaptive cruise control subject to V2V impairments, pp. 1-8, 2017.

[18] A. Schmeitz et al, *IEEE Int. Conference on Intelligent Transportation Systems*, Towards a generic lateral control concept for cooperative automated driving: theoretical and experimental evaluation, pp. 134-139, 2017.

[19] L. Ferranti et al. "SafeVRU: A research platform for the interaction of self-driving vehicles with vulnerable road users.", *IEEE Int. Vehicles Symposium.*, pp.1660-1666, 2019.

[20] A.J.C. Schmeitz et al. "Extending the Magic Formula and SWIFT tyre models for inflation pressure changes", *Vdi Berichte V.1912*, pp. 201, 2005

A

Fail-safe mitigation algorithm

In this chapter the fail-safe mitigation algorithm presented in Chapter 2 is further explained, stating the relevant choices of all components and giving their related settings.

A.1. Vehicle following

As declared in the introduction, this research originates from questions that are raised on platooning and CAD research. However, in this research, no actual platoon is implemented. Nonetheless, the ACC string of vehicles that is implemented is based on a platoon and shown in Figure A.1. It uses a leader-follower approach with a static leader and the platoon communicates via the predecessor following topology as both described in [28]. All vehicles are the same, thus have the same vehicle parameters, which means that it is a homogeneous platoon.

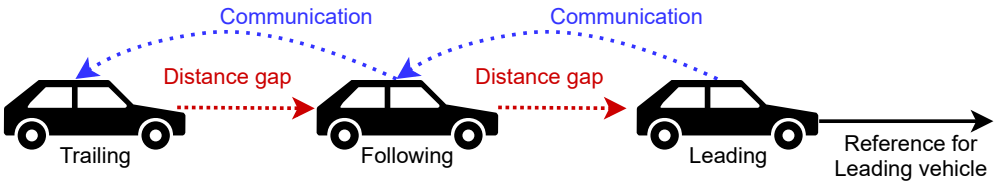

Figure A.1: Leader-follower structure of a platoon with predecessor following communication topology

In order to mimic this platoon as closely as possible, but using an ACC string of vehicles, a constant time headway of 1 s is chosen. For a homogeneous platoon of vehicles, this is deemed string stable according to [33]. The constant time headway is chosen as small as possible, in order to make the results as representative as possible for a platoon of vehicles, whilst using a string of ACC vehicles. constant time headway influences the difference between both mitigation strategies imminently due to the difference in time-gap error it creates, which then again increases the difference in gap-closing time of the trailing vehicle.

A.2. Vehicle model

Choosing the dynamic bicycle model bases itself around the capabilities and limitations of this model, supplying sufficient information but minimizing the complexity of the equations, thus computational effort for the controller. The model is used both inside the controller for update equations, as well as in the vehicle model that represents the dynamics of an actual vehicle. Figure A.2 shows the dynamic bicycle model and its variables, with Equations (A.1) and (A.2) showing its state update equations.

Some important pointers from Figure A.2 are the fact that the y-direction is positive to the left side. Also, the figure shows v and u as the lateral and longitudinal velocity respectively, however in this research, to overcome any confusion and keep notations consistent, v is taken as velocity with subscripts $_x$ and $_y$ for the

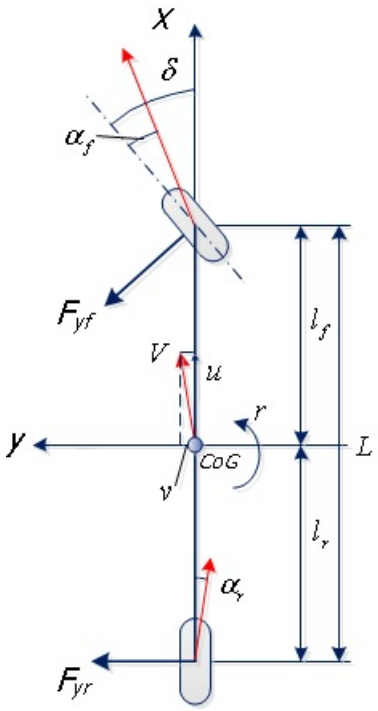

Figure A.2: Dynamic Bicycle Model [27]

longitudinal and lateral direction respectively. This is also implemented in Equations (A.1) and (A.2).

$$\dot{v}_y = -\frac{C_{\alpha f} + C_{\alpha r}}{m v_x} v_y + \left(\frac{l_r C_{\alpha r} - l_f C_{\alpha f}}{m v_x} - v_x \right) r + \frac{C_{\alpha f}}{m} \delta \tag{A.1}$$

$$\dot{r} = \frac{l_r C_{\alpha r} - l_f C_{\alpha f}}{I_z v_x} v_y - \frac{l_f^2 C_{\alpha f} + l_r^2 C_{\alpha r}}{I_z v_x} r + \frac{l_f C_{\alpha f}}{I_z} \delta \tag{A.2}$$

The major limitations and validity regions that the dynamic bicycle model has are as follows [27]:

- A linear relationship between lateral forces and slip angles, $F_{yf} = C_{\alpha f} \alpha_f$ and $F_{yr} = C_{\alpha r} \alpha_r$, is presumed
- The lateral tire force is proportional to the slip angle, $\sum F_y \approx F_{yf} + F_{yr}$, but only yields for small steering angles, $\delta \leq 5\,\text{deg}$
- Constant longitudinal velocity is assumed
- The vehicle should stay within linear region between the steering wheel angle and lateral acceleration, i.e. $a_y \leq 0.4g$

The assumption for constant longitudinal velocity is required to decouple the lateral and longitudinal dynamics of a vehicle, thus being able to simplify the dynamics significantly. Changing the longitudinal velocity nonetheless leads to inaccuracies in these dynamics when accelerating or decelerating out of curves. However, as [21] proves the reference velocity and path are well tracked when using the dynamic bicycle model with an added longitudinal model.

The added longitudinal model in this research bases itself around [23], with $\tau = 0.1\ s$, a time constant representing the engine dynamics, and $\phi = 0$, thus assuming there are no sensor or actuator delays present. Equation (A.3) shows the resulting longitudinal dynamics, in which $a_{x,c}$ is the intended and a_x the realised longitudinal acceleration.

$$a_x = \frac{1}{\tau s + 1} a_{x,c} \tag{A.3}$$

Furthermore, s denotes the system being in the Laplace Domain.

A.2.1. Discrete-time Stability

For the internal model of the MPC, the lateral dynamics of the vehicle model are discretized using the forward Euler method as explained in Chapter 2. With the derivative equations for v_y and r needed due to the dynamic bicycle model, the forward Euler method is chosen as a simple but effective method when the sampling time is sufficiently low These discretized equations should be stable during the whole manoeuvre, to ensure stable vehicle behaviour and stability of the controller over the entire prediction horizon. The stability of the forward Euler method can be determined by taking the eigenvalues of A_{DT}, following Equation (A.4). In which A_{CT} is a matrix containing continuous time state dynamics from the dynamic bicycle model.

$$A_{DT} = I + A_{CT}\Delta t \tag{A.4}$$

Figure A.3 shows the eigenvalues of the system, calculated for $v_{x,\min} \leq v_x \leq v_{x,\max}$ with steps of 0.01 m/s and a sampling time $\Delta t = 0.01$ s. This sampling time is chosen based on the stability region it generates, thus considering the lowest possible stable longitudinal velocity, the realistic estimation of the continuous time signal and on the increased prediction horizon when a smaller value is chosen. The figure shows that over the whole range of velocities, $v_{x,\min} = 1.26$ m/s and $v_{x,\max} = 33.3\bar{3}$ m/s, the eigenvalues are within the unit circle, thus ensure stability.

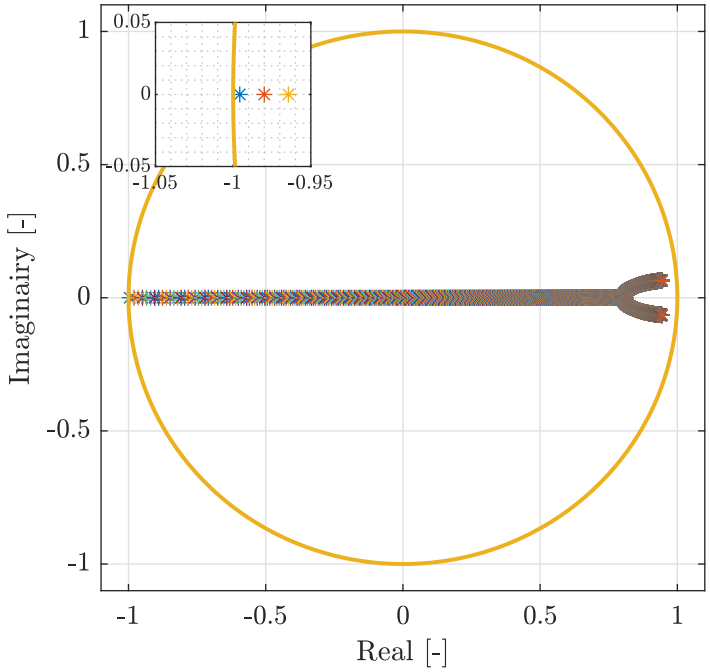

Figure A.3: Poles of the discrete time update equations of the lateral dynamics over the range $1.26 \leq v_x \leq 33.3\bar{3}$

The longitudinal model in Equation (A.3) is discretized using the Matlab function `c2d`, in order to use it in the MPC model. This function transforms the continuous time dynamic system from Equation (A.3) into a discrete time dynamic system using the sampling time Δt. The pole and gain from the discrete time dynamic system are $s_{dt} = 0.9048$ and $G_{dt} = 0.0952$ respectively, where $| s_{dt} | \leq 1$ thus the discrete time longitudinal dynamics model is also stable.

A.3. Trajectory generation

The required trajectory consists out of three states, longitudinal velocity v_x, lateral position d_y and heading angle θ. Longitudinal velocity and lateral position are required for the braking manoeuvre and lateral movement to the road shoulder respectively. Furthermore, the heading angle guarantees that the orientation of the vehicle is parallel to that of the road at the end of the manoeuvre. This is needed as otherwise the vehicle, in theory, could be perpendicular to the orientation of the road at the end of the manoeuvre.

As the vehicle drives on a straight part of a highway, the initial conditions are taken as $v_x(0) = 100$ km/h (≈ 27.78 m/s), $d_y(0) = 0$ m and $\theta(0) = 0$ rad and the goal conditions as $v_x(goal) = 1.4$ m/s, $d_y(goal) = -3.375$ m and $\theta(goal) = 0$ rad, to park the vehicle on the road shoulder. The desired goal longitudinal velocity is 0 m/s,

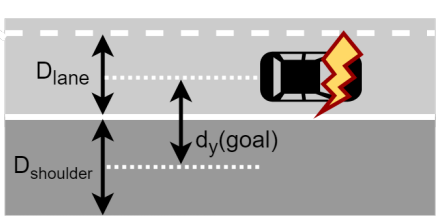

Figure A.4: Lane width of the active lane and road shoulder, including the goal lateral position

however due to the stability of the discretized state-update equations this is not feasible, as explained in Section A.2.1. For the lateral position, the goal is determined on the basis of the lane width of the right most lane in which it is driving and the lane width of the road shoulder, taking the distance between the centers of both, as shown in Figure A.4. The width of the right lane, D_{lane}, is taken to be 3.25 m following [11] and the width of the road shoulder, $D_{shoulder}$, is 3.5 m according to [24].

For the generation of the lateral trajectory, the Matlab function `quinticpolytraj` is used, this function generates a 5^{th} order polynomial between the given waypoints. These waypoints are based on positions with respective velocities, also acceleration bounds can be added. At each waypoint, the function assumes that the heading angle is zero, thus in order to create a smooth trajectory only the initial and goal positions are used as waypoints, with initial and goal velocities. As a consequence of the heading angle being zero at each waypoint, the longitudinal and lateral reference are separated to accommodate both mitigation strategies. The goal longitudinal position is determined by multiplying the initial longitudinal velocity with the desired time that the lateral movement should take, t_{man}. This time t_{man} is 3.5 s and determined iteratively such that the lateral acceleration is around 1.5 m/s^2, seen as a comfortable lane change manoeuvre [4], but the vehicle also leaves the current lane swiftly. The function outputs acceleration, velocity and position references, with time interval Δt and a total time of t_{man}. As no reference for the heading angle is supplied, this is calculated from the lateral position using the formula $\theta = \tan^{-1}\left(\frac{d_y(k+1)-d_y(k)}{v_x \Delta t}\right)$. Figure A.5 shows the reference trajectory for all three states. The reference stays constant after 6 s until the simulation is stopped.

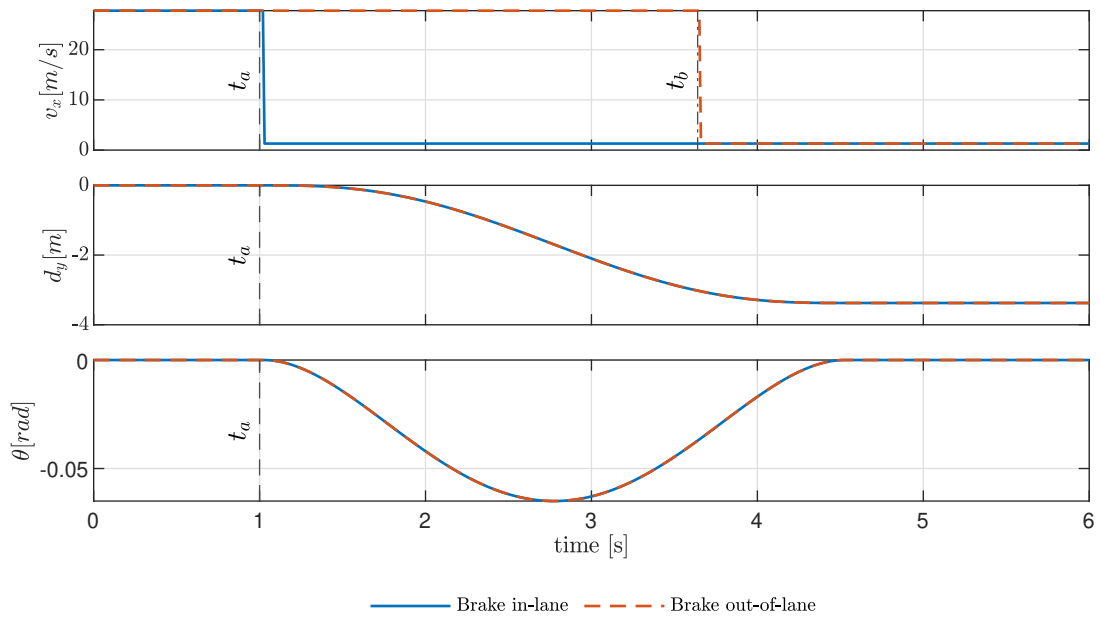

Figure A.5: Reference over time when parking on the road shoulder is initiated at $t_a = 1s$ and braking in current lane is performed

A.4. Model Predictive Control

As the working principle of MPC is extensively explained in literature (i.e. [20]), no further explanation will be given on this theory. The MPC in this research uses an interior-point algorithm to find the optimal solution at each time step, for the theory behind this algorithm [36] can be consulted. Furthermore, the choice of the algorithm is outside of the scope of this research, the interior-point algorithm is a default setting in the MPC toolbox from Matlab, thus not a design choice.

A.4.1. Constraints

Section A.2 already highlights some of the constraints that are required to use the dynamic bicycle model, one of which is that only small steering angles are allowed. For this reason the constraint on $|\delta|$ is 5 deg (0.0873 rad). Next to that, Section A.2.1 highlights the required lower limit on the longitudinal velocity of 1.26 m/s, the upper limit is taken as 120 km/h (≈ 33.33 m/s).

Lateral acceleration a_y is, together with longitudinal acceleration, the most profound limitation within this research. The lateral acceleration is limited by the dynamic bicycle model to $0.4g$, as this is the end of the linear region of the relation between steering wheel angle and lateral acceleration. However, due to comfort constraints and the importance of comfort in the acceptance of automated manoeuvres in automated driving, the limit for a_y is chosen at the limit that can be reached in normal highway driving according to [4], namely 2 m/s^2.

Next to constraints initiated by the dynamic bicycle model or comfort, constraints that enhance the representation of real vehicle behaviour are implemented as well. For that reason. the longitudinal acceleration is different compared to longitudinal deceleration, as a vehicle in general has a lower acceleration compared to its deceleration. The deceleration limit is set to the maximum allowed ACC braking, according to ISO 15622, which is $-3.5m/s^2$. The maximum acceleration is chosen to be $1.5m/s^2$, which is seen as normal driving behaviour according to [5]. Furthermore, the rate of change of the acceleration and deceleration are chosen such that they can reach their respective maxima in 0.25 s, which translates into $-14 \leq \dot{a}_{x,c} \leq 6$ m/s^3. As tyres or steering wheel cannot rotate at infinite speed, the steering rate $\dot{\delta}$ is limited to 75 degrees per second at the steering wheel. Dividing this by the steering ratio (1 : 16, i.e. 1 rad of change at the steering wheel is $\frac{1}{16}$ rad at the tires) and transforming this from degrees to radians results in the steering rate at the wheels of the vehicle, $|\dot{\delta}| \leq 0.0818$ rad/s.

A.4.2. Tuning

Whilst constructing and tuning the controller, simulations would often take over 3600 s to simulate $10-15$ s because the optimization algorithm within the controller could not find feasible solutions in parts of the simulation. This is undesirable on the basis of computation time and effort, but most of all because of the infeasibility of the problem. It was found that the infeasible solutions predominantly originated due to the violation of $v_{x,\min}$, or by the system becoming unstable. Section A.2.1 solved both these issues, however to prevent the simulation from running whilst finding infeasible solutions, the simulation is automatically stopped when it cannot find a feasible solution five times. The stopping of the simulation is referred to as *crashed* in the remainder of this research. Furthermore, tuning is performed on the braking in-lane mitigation strategy due to the higher dynamical load, as explained in Chapter 2.

Prediction and Control horizons The prediction horizon P and control horizon S determine the look-ahead time and the number of optimized variables of this horizon respectively. In the Matlab MPC toolbox, which is used in this research, the final value of S is used to predict the further $P-S$ time steps if there is a difference between the two, under the condition that $P \geq S$. The tuning is performed using the weights presented in Chapter 2.

Both prediction and control horizons are varied between 10 and 100, each time first increasing the prediction horizon and than also increasing the control horizon to the same level. Figure A.6 shows the results on the controller output and subsequent lateral position error and longitudinal velocity. For the lateral control, the difference in lateral position error is only visible when zoomed in on certain areas, which is made visible in Figure A.7. Here the 'dip' in steering input caused by the sideslip angle is shown on the left and the end of the lateral manoeuvre is shown on the right. The dip shows that in general a lower P and S result in a reduced dip and smoother trajectory following, however the error in lateral position is also slightly higher. In contrary, at the end of the lateral manoeuvre the error is smallest for $P = S = 10$, but largest for $P = 30, S = 10$ and equal for all other variations.

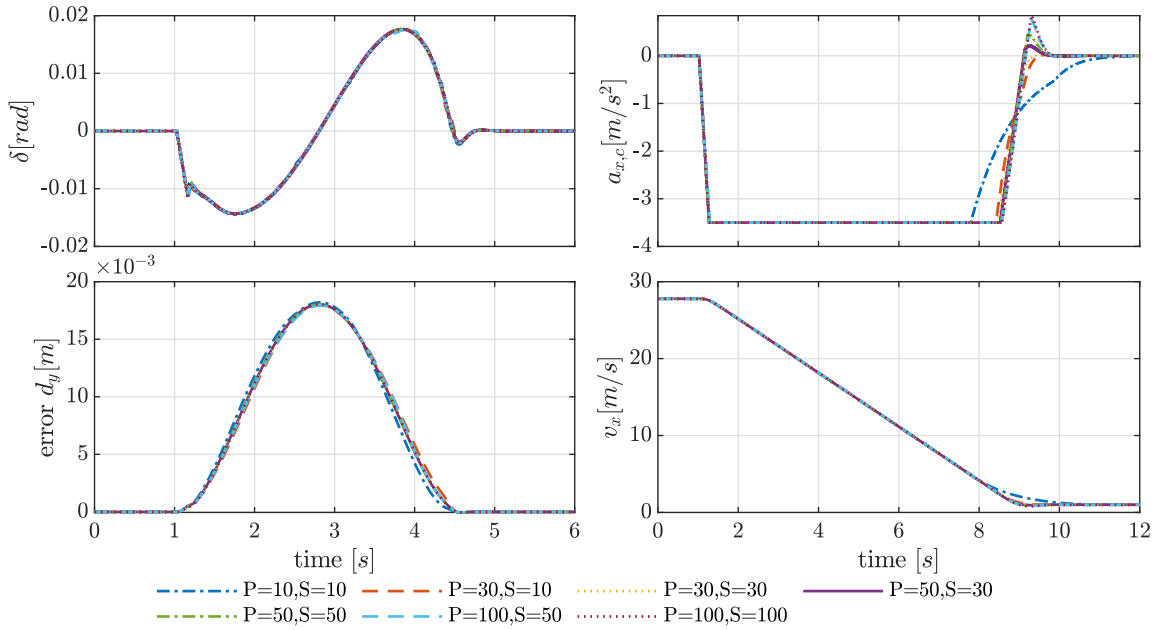

Figure A.6: Tuning of prediction and control horizons

Now looking at the longitudinal manoeuvre, there are more varieties visible towards the end of the manoeuvre. The beginning of the manoeuvre is the same due to the sudden change in reference, which makes each set of horizons immediately go towards the maximum output possible. Zooming in on the latter part of the longitudinal manoeuvre in Figure A.8, it is clearly visible that $P = S = 10$ decreases its breaking first and subsequently takes longest to reach the goal velocity. $P = S = 30$ reaches the goal velocity quickest. Furthermore, all horizons being ≥ 50 result in positive acceleration of the vehicle, thus having braked too hard. From a human driving perspective this is undesirable behaviour as the vehicle should slow down to its goal velocity and not accelerate after decelerating. However, as there are no constraints applied on the controller to prevent this underdamped behaviour, this is normal behaviour of the controller.

From these results a prediction horizon of 30 and a control horizon of 10 or 30 are the most desirable. To make a decision between these two and also to gain insight into the computational effort of all horizons, Table A.1 shows their run-times. From this table it is clear that a larger horizon results in a longer run-time, which is expected as for a larger horizon, the optimization algorithm should optimize more values thus the computational load is higher. A remarkable thing to notice is the fact that if P and S aren't equal, the size of P predominantly determines the run-time of the simulation, caused by the extra amount of calculations that have to be made to optimize the final value of S, thus increasing the computational load significantly. As a consequence, it is desired to have equal P and S. For the reasons mentioned above, the chosen settings for the prediction and control horizon are both 30.

Table A.1: Simulation run-time difference for tuning the prediction and control horizon

Prediction Horizon (P)	Control Horizon (S)	Simulation Run-time [s]
10	10	87
30	10	504
30	30	339
50	30	984
50	50	908
100	50	6496
100	100	7958

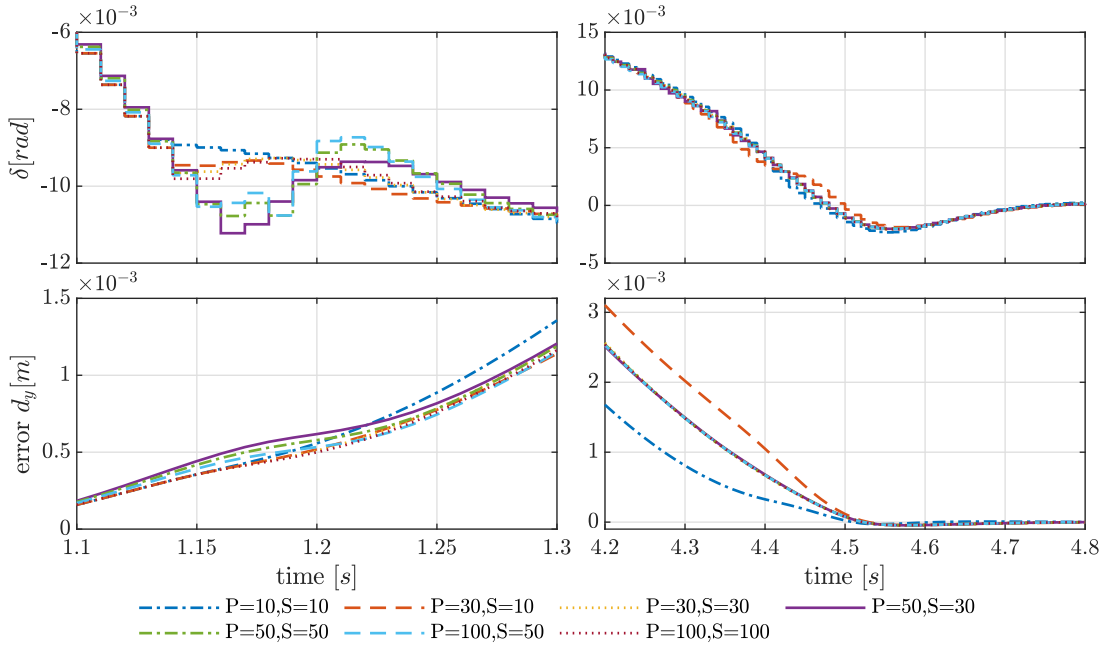

Figure A.7: Tuning of prediction and control horizons, zoomed in on most important parts for lateral control

Cost function The settings obtained in the previous parts are used to tune the cost function of the controller and for the sake of completeness given Table A.2.

Table A.2: Settings used for tuning the cost function

Variable	Δt	P	S	t
Value	0.01	30	30	0,...,25

The eventually chosen tuning parameters are given in Chapter 2 and in the following figures indicated as *tuned*. To indicate the differences that the tuning parameters make and to demonstrate the behaviour of the controller, single weights within the cost function are varied to larger and smaller values, as well as set to zero (i.e. the subsequent term is removed from the cost function thus has no effect on the controller output) and the remainder of the weights are equal to the tuned version. Figure A.9 shows the control outputs of all tuning variations and Figure A.10 shows their impact on the tracking of the reference.

From these figures it is clear that the controller reacts to changes in the cost function and tracking performance can be influenced by changing the weights. Especially when disregarding the longitudinal velocity v_x in the cost function (i.e. $w_{v_x} = 0$), the controller does not track the longitudinal velocity and only performs the lateral movement. Here it highlights that the lateral trajectory is generated at constant velocity as the tracking of the heading angle is best tracked when the velocity does not change.

The controller did crash in two instances, being when $w_{a_{x,c}} = 0$ and $w_{v_x} = 100$. In both instances the controller crashes after roughly 9 seconds, which is around the time the longitudinal velocity reaches its goal, thus comes close to its lower limit. This immediately highlights the reason of the crash, namely the lower limit set to the velocity, as the controller cannot find feasible solutions to stay above that velocity. Decreasing the lower bound on longitudinal velocity does result in a feasible solution, however this is not accepted as a possibility due to the stability of the model. To overcome the crashing of the controller, $a_{x,c}$ is added to the cost function, thus $w_{a_{x,c}}$ is taken as nonzero, and w_{v_x} is taken < 100. Furthermore, when increasing $w_{a_{x,c}}$ the controller relaxes the braking more early compared to the tuned setting, which results in reaching the goal velocity at a later time instance, thus increasing the stopping time. As the vehicle should stop as quickly as possible this is undesired.

The oscillatory behaviour of the steering output when $w_\delta = 0$ is undesired because of the disturbances it can cause within the system. Although the tracking performance is excellent with this setting, δ is added to the cost function to remove these oscillations.

From Figure A.10 the effect of $w_{d_y} = 0$ is clearly visible, the vehicle does make a smooth lateral movement,

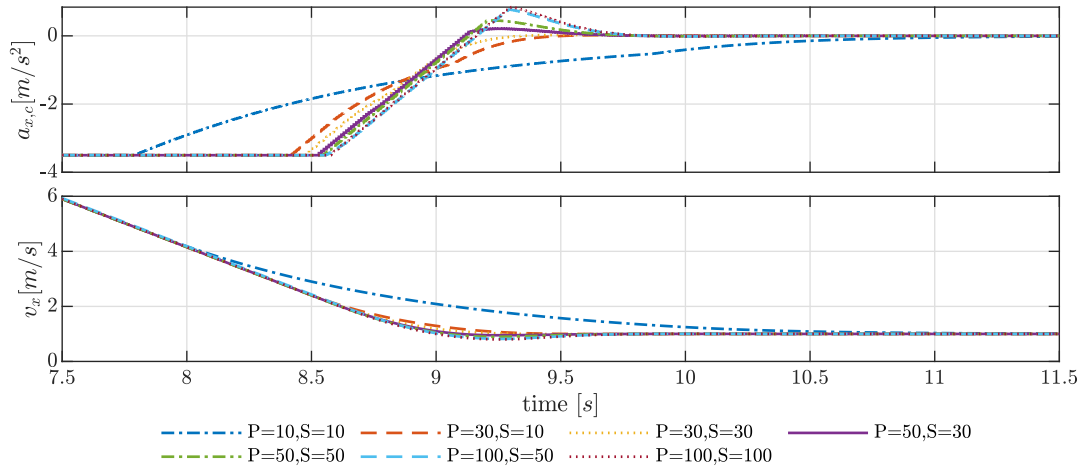

Figure A.8: Tuning of prediction and control horizons, zoomed in on most important part for longitudinal control

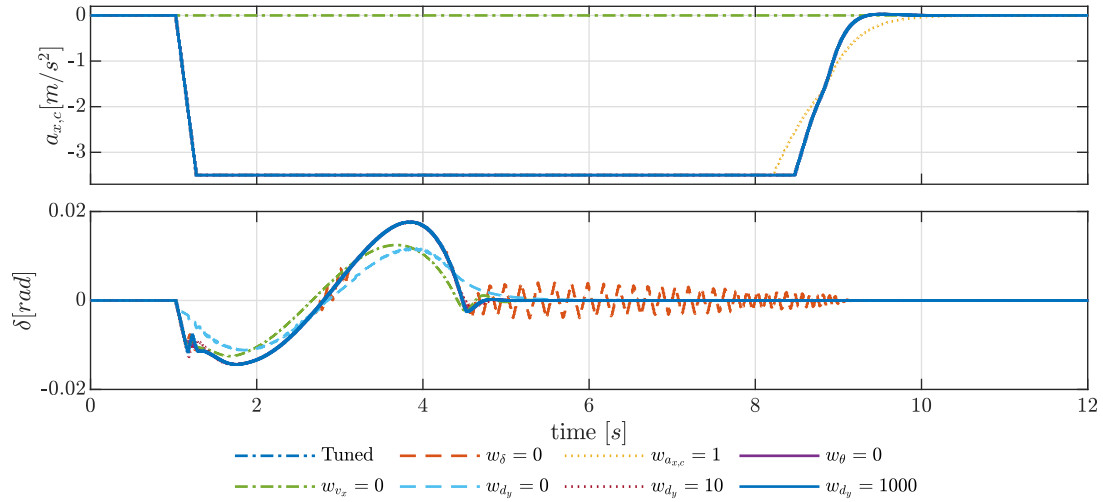

Figure A.9: Cost function tuning, showing the difference in controller output

however does not reach the desired goal position as the position is not tracked. The movement is thus performed by tracking the heading angle, which it does not do perfectly due to the difference in weight between w_{v_x} and w_θ.

In conclusion, the necessity of adding $a_{x,c}$ and δ to the cost function is highlighted and the changes in weights for v_x, d_y and θ show their effects on the overall tracking performance.

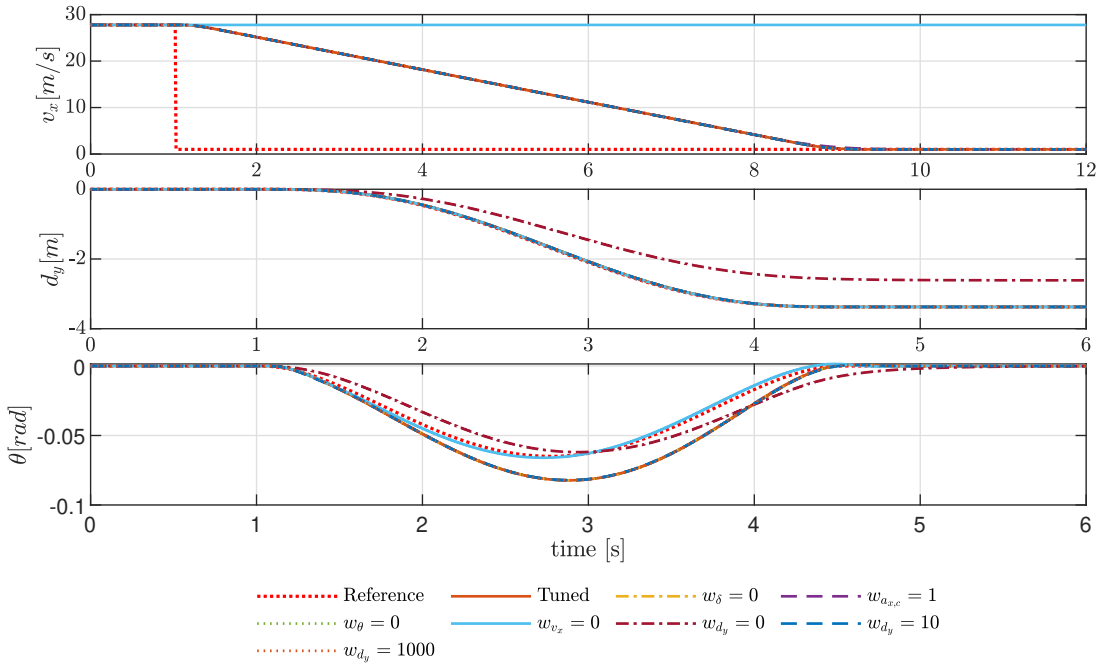

Figure A.10: Cost function tuning, showing the difference in state output compared to the reference

B

Additional Results & Limit Handling

This chapter gives additional results to provide further insight into the results of all simulations, as well as the handling and the dynamics of the vehicle towards the limit of the constraints.

B.1. Additional results

As the dynamic bicycle model uses the equations for yaw rate r and lateral velocity v_y to represent the vehicle dynamics, for the sake of completeness, Figure B.1 shows the lateral velocity v_y of all simulation results executed in Chapter 2. This figure also shows the changed vehicle dynamics due to the tyre failure, resulting in an increase in lateral velocity of almost 0.18 m/s.

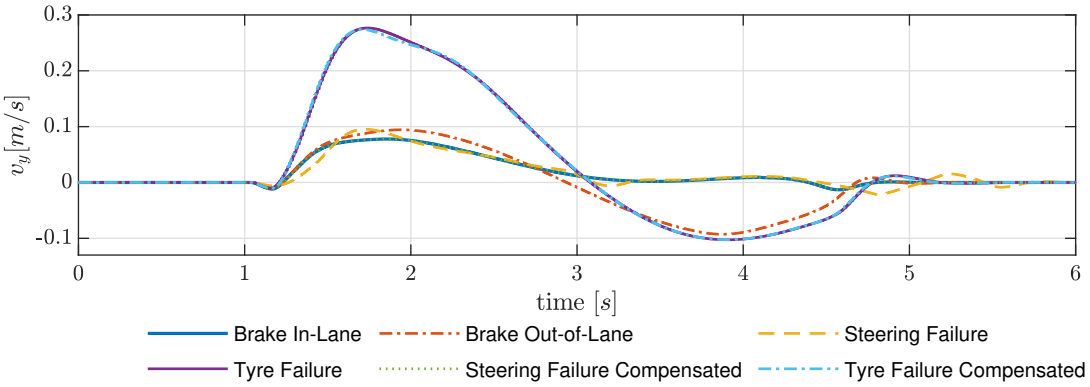

Figure B.1: Lateral velocity of all simulations

In literature and in this research up until now, only singular faults and failures are considered. For that reason, both the steering and rear tyre failure are implemented simultaneously. Subsequently, the internal model of the NMPC is changed using the information about both failures, thus reconfiguring the controller. Figure B.2 shows the results for the original and reconfigured controller, showing that the original controller is not robust enough to control the vehicle at low velocities. At high velocities the controller has to use the maximum allowed steering rate and changes in steering direction multiple times.

The reconfigured controller controls the vehicle without any problems, only the change in dynamics is visible from the rear tyre failure. This highlights the importance of the architecture even more, using the information about the failures and reconfiguring the controller makes the vehicle controllable when it originally was not robust enough.

B.2. Limit handling

Two limits of the controller are highlighted in this section, being i) handling the imposed inequality constraint on lateral acceleration a_y and ii) exploring the limits when having a steering failure.

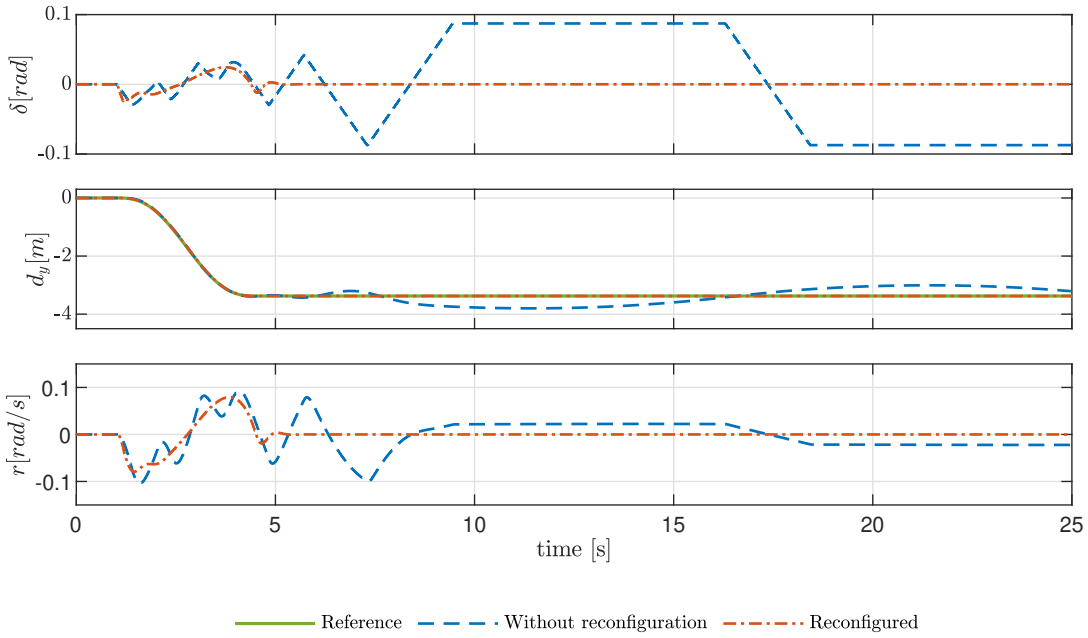

Figure B.2: Both rear tyre and steering failure at the same time, comparing the standard and reconfigured controller

B.2.1. Inequality constraint lateral acceleration

To understand the behaviour of the controller around the limit of the given inequality constraint $\mid a_y \mid \leq 2\ m/s^2$ (indicated in red in Figure B.3), the manoeuvre time t_{man} is decreased until the controller crashes.

Decreasing t_{man} means that the vehicle moves faster to the road shoulder, which increases the dynamic load on the vehicle and subsequently the lateral acceleration. With the standard settings of the controller (see Chapter 2), the shortest possible manoeuvre time t_{man} which the vehicle can handle (i.e. find solutions for the algorithm at all iterations and time steps) is 3.2 s. The results in Figure B.3 show that when the controller notices the vehicle will violate the constraint, around 1.2 s, it at first uses the maximum change in steering input (i.e. the limit of $\dot{\delta}$), to keep itself within the limit. Afterwards, the controller stabilizes the steering and keeps it roughly constant and with that maintaining a roughly constant lateral acceleration, while maximizing the change in lateral position thus minimizing the tracking error.

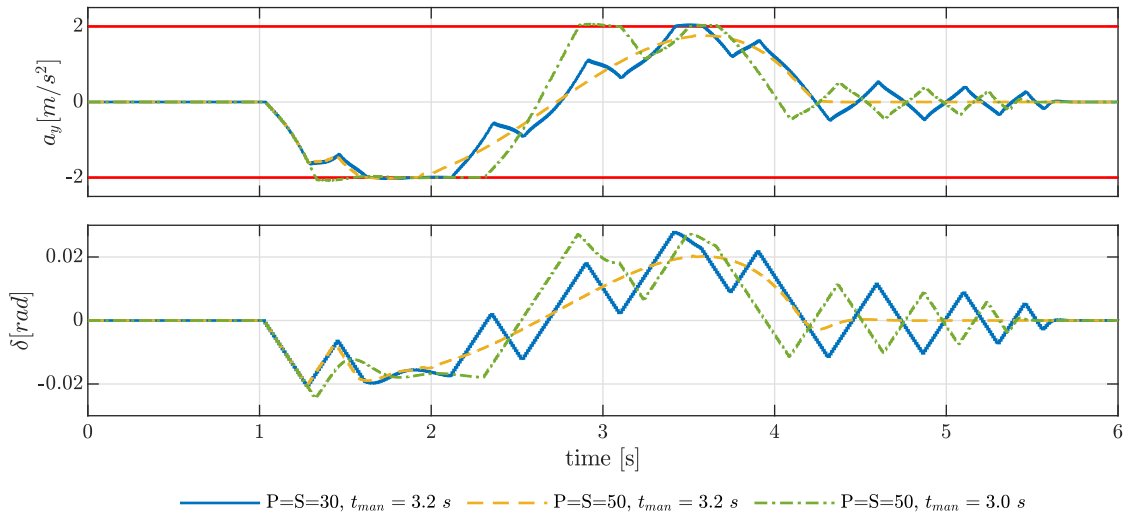

Figure B.3: Lateral acceleration and steering input at the limit of the lateral acceleration, showing the limit handling capabilities of the controller

To gain insight into the behaviour of the controller when it can predict reaching the limit of the constraint more early, the prediction and control horizon are increased to 50. As Figure B.3 shows, this does not change

the initial reaction of the controller significantly. However, the time it operates at the limit of a_y is reduced and after moving away from this limit, there is smooth control in the steering input, not going into the limit of $\dot{\delta}$. This thus is a big improvement compared to $P = S = 30$.

With the increased horizons, the minimal manoeuvre time decreases to 3.0 s, which is a reduction of 0.2 s compared to the original setting. This reduction shows that with a longer horizon, the controller is capable of handling more extreme scenarios, thus increases its capabilities. The results show similar behaviour to the original horizons, smoothing out δ when the limit is reached and often using the maximum steering rate. Another thing that becomes visible is that with the increased horizons, the goal lateral position is reached quicker, as the steering movements stop more early.

Whilst looking closely at the results, the inequality constraint is violated slightly. This is due to the inaccuracy the forward Euler discretization method introduces. The measurements of a_y that Figure B.3 shows are from the continuous time output of the vehicle, not from the calculation of a_y in the controller. When consulting the predicted a_y from the controller, it showed that the vehicle would stay exactly on the limit of the inequality constraint.

B.2.2. Steering Failure

The steering failure is implemented as shown in Figure B.4, where the output of the controller is reduced by 50% before it is fed through to the vehicle model.

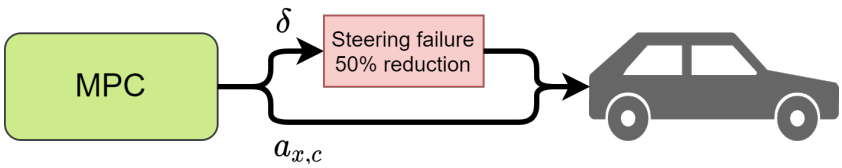

Figure B.4: Explanatory figure of the implementation of the steering failure

To know the maximum reduction in steering output with which the controller, with its standard settings, can still perform the manoeuvre to the road shoulder and settle around the goal lateral position, the reduction in steering output is iteratively increased. At a reduction of 59%, the controller is still capable of settling around the goal lateral position before the goal velocity is reached. However, when reducing it further to 60%, the vehicle will keep swerving around the goal lateral position for a long period of time, as Figure B.5 shows.

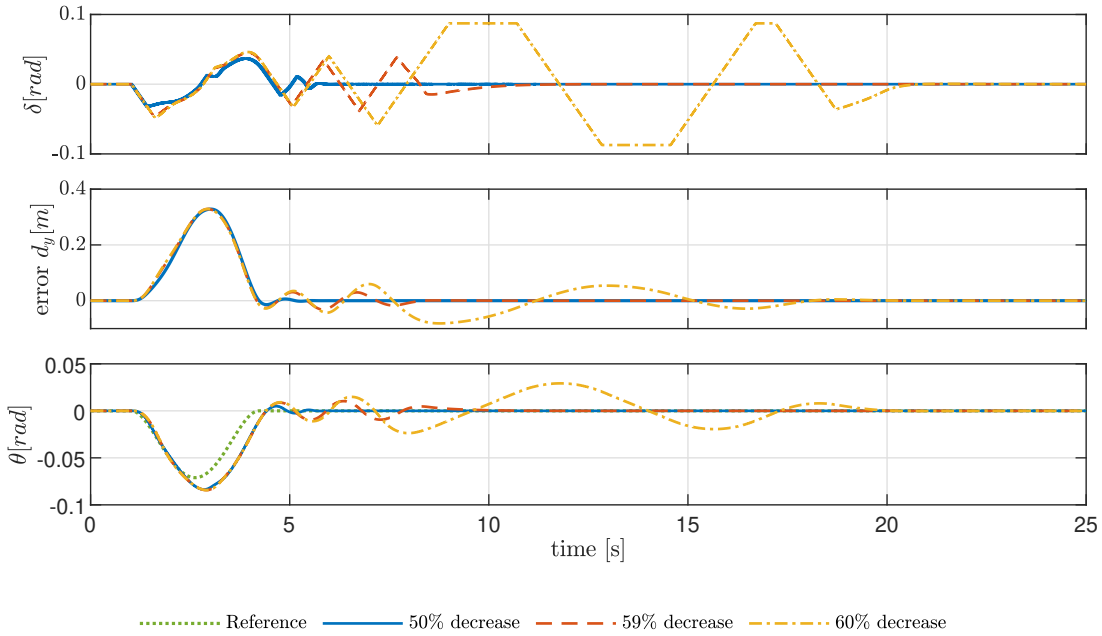

Figure B.5: Comparing 59 and 60 % decrease in steering output on steering output, lateral error and heading angle

Furthermore, Figure B.5 shows that at a reduction of 60%, δ reaches its limit multiple times and for several seconds, making the heading angle of the vehicle increase when trying to reach the goal position. This is

predominantly visible at lower velocities, as this makes the lateral movement take longer and as a result lets the vehicle overshoots the lateral position d_y as well. This indicates that the prediction horizon is not sufficiently long enough to start reducing the steering on time to not overshoot the goal lateral position. The error on lateral position shown in Figure B.5 is calculated by subtracting the reference trajectory from the output data of the vehicle model. At lower velocities, the limiting factor within the manoeuvre is predominantly $\dot{\delta}$. However, as shown by the reconfiguration of the controller in Chapter 2, when the limit is extended to match the capabilities at 0% reduction, the controller performs as good as in a healthy situation.

A reduction of 50% is chosen to be used in this research to indicate a significant reduction in steering output, while staying within the limit of δ and maintaining a comparable lateral manoeuvre time to that of a healthy vehicle.

C

Conclusion & Discussion

This chapter provides an overview of the conclusions that can be drawn from this research in Section C.1, followed by a discussion in Section C.2, in which also recommendations are made on future work.

C.1. Conclusion

With the development of platooning systems and the increase in research on autonomous driving, safety architectures and fallback algorithms are required to ensure safety when faults occur within the vehicle or within a string of vehicles. This research proposes a functional safety architecture approach, showing the process of handling faults in automated driving (i.e., fail-operational mitigation), as well as a fail-safe mitigation algorithm in the case of a severe fault. The proposed architecture supplies all steps from the detection of the fault to the different mitigation strategies, dependent on the fault severity. The fail-safe mitigation algorithm performs a fail-safe manoeuvre to the road shoulder and operates within a safety channel that runs parallel to the nominal channel.

C.1.1. Functional Safety Architecture

The proposed architecture underlines the need of having three separate layers to ensure safety in critical fallback scenarios, being a nominal channel, a health monitor and a safety channel. The health monitor is required to detect faults and classify their severity in order to ensure safe control and make a distinction between fail-safe and fail-operational mitigation. With higher levels of automation, both fail-safe and fail-operational mitigation strategies are required due to the longer period of time needed when a transition of control is requested. For fail-safe mitigation it is of essence to have a separate controller, situated in the safety channel, to ensure redundancy when the initial controller fails or hardware components fail. This controller is specifically designed for fallback scenarios and handling faults or failures.

C.1.2. Tactical Decision Making

Given a fault-severity, the tactical decision making module will determine the mitigation strategy and if a degraded functionality is desired or the vehicle needs to be brought to a safe state, i.e. stop in lane, park on the road shoulder, drive to repair shop. This mitigation strategy depends on multiple factors, e.g. the severity of the fault, the environment, the traffic context and the scenario. When smaller timegaps are considered in a string of vehicles, i.e. an ACC string or even more relevant a platoon, the higher the need and the influence of having a well-designed tactical decision making module. By analysis of the braking in-lane and braking out-of-lane strategies used in this research, it is shown that the braking out-of-lane strategy is preferred if the impact on the remainder of the string of vehicles should be minimized. Furthermore, when it is required to act fast whilst bringing the vehicle to a safe-state, braking in-lane is a better strategy.

C.1.3. Fail-safe mitigation algorithm

Trajectory generation can have different targets, depending on the severity classification of the fault and the need to act fast or not. Furthermore, depending on the scenario, the optimal strategy might be different, e.g. if the emergency lane is limited in length or blocked until a given point. To accommodate for the controller incorporating functional failures, one needs to use a model-based control method, such as MPC, that houses

a vehicle model. The results of this thesis show that by using a model-based method and compensating for the faults in the controller design, the errors compared to a healthy vehicle can be reduced by up to 92%.

C.1.4. Results

When parking the vehicle on the road shoulder, the resulting performance in lateral deviation depends from fault to fault, which in turn requires the adapting of the controller to each specific fault. We showcase in this research how to deal with two specific faults, namely a power steering failure and a rear tyre failure. Taken together, these results indicate that the controller is robust against both faults and is capable of reaching an accuracy of 0.027 m in lateral position error, using a nonlinear MPC. Next to that, the mitigation strategy for braking out-of-lane results in a 65 % increase in time gap compared to braking in-lane, however in a 24 % decrease in stop time and 38 % decrease in stop distance. When driving in a platoon, or from a fuel efficiency point of view, it might be undesirable to create very large time gaps as they require more and longer acceleration/deceleration the remaining vehicles.

C.2. Discussion & Future Work

Within this section some limitations of the current methods are discussed and recommendations are given for future work.

C.2.1. Fail-safe Mitigation Algorithm

When converting the chosen prediction and control horizons to look-ahead times, the vehicle only looks ahead 0.3 s. As it is recommended for human drivers to keep a 2 s gap to the preceding vehicle [8] and the ACC string of vehicles used in this research has a time-gap of 1 s to its predecessor, a look-ahead time of 0.3 s would be unrealistic in real life scenarios. With such short look-ahead times the vehicle will not be able to adequately react to sudden changes in its environment or objects appearing in front of the vehicle. However, the trajectory generation can be made responsible for the collision avoidance, thus assuring a collision free trajectory is given to the controller. Then the controller should only track the given collision free trajectory and a look-ahead time of 0.3 s would suffice. In a situation where the controller does incorporate trajectory generation with collision avoidance, the look-ahead time of both horizons should be extended to at least the recommended 2 s. As this will result in long computation times, highlighted in Section A.4.2, at first there is a need to speed up the algorithm within the MPC to enable real-time implementation. A possible solution to this would be choosing a different solver or using a different toolbox than the one used in this research. To asses real-time implementability this topic should be explored in future work.

For the trajectory generation a 5^{th} order polynomial function is used, based on only two waypoints and is not checked for potential collisions with surrounding vehicles or other objects. Furthermore, the waypoints are assumed to have a 0 heading angle, as explained in Appendix A.3, which means that (partly) curved roads cannot be generated using this function, thus testing scenarios are limited to straight roads. An alternative method could be online trajectory generation, which not only is capable of handling multiple scenarios in varying environments, but following recent research also has the possibility to avoid collisions [12]. This is desired as chances are that the road shoulder is not empty at the time of initiation of the fallback manoeuvre, thus the vehicle should avoid these parts in order to not crash while performing the fallback manoeuvre. For these reasons, the addition of online trajectory generation with collision avoidance is highly recommended for future work. The topic online trajectory generation was out of the scope of this thesis, however a modular approach is followed to accommodate other trajectory generation methods.

Currently the algorithm is only tested on a straight road, which means there are no lateral forces acting on the vehicle due to driving in a curve. To asses the controller in more complex scenarios, such as curved roads, it is highly recommended to test such scenarios in future work. The effect these more complex scenarios could have is that the vehicle reaches the limit in a_y and is not capable of steering further without violating this constraint. As a consequence, the vehicle could not be able to follow the desired trajectory closely and swerve of the road.

C.2.2. Model limits

In general, the controller keeps the vehicle within all imposed constraints and bounds. However, results show that the bound on $\dot{\delta}$ is a limiting factor of the controller when operating around the limit of the inequality constraint on a_y. Although the controller does track the reference correctly and stabilizes in adequate time when reaching the limit of a_y, the lateral overshoot increases from 0.02 m to 0.07 m at the shortest possible

t_{man} for $P = S = 30$ and $P = S = 50$ respectively, which is equivalent to an increase of 250%. Furthermore, when reconfiguring the controller the bounds on δ and $\dot{\delta}$ are increased. As a critical note it should be said that the increase of $\dot{\delta}$ depends on the capabilities of the steering actuator that is within the vehicle, as a physical actuator also has its limits. This is out of the scope of this work, but is recommended to look into when performing real-life testing. A solution could be to change the bounds on $\dot{\delta}$ to its physical limit and impose an inequality constraint for the desired limits, depending on the condition of the vehicle.

By using the dynamic bicycle model, the vehicle is only able to reach a minimal velocity of 1.26 m/s, while it should come to a full stop, i.e. 0 m/s. As the intention of the manoeuvre is to bring the vehicle to a standstill, this is not desired. However, this is not deemed an issue as this is a relatively low velocity and the final part of the manoeuvre could be performed by a feed-forward controller, to bring the vehicle to a standstill.

C.2.3. String of vehicles

This research uses a string of ACC vehicles to mimic the effects that both mitigation strategies (braking in-lane or out-of-lane) have on a platoon of vehicles. As the control methodologies of ACC vehicles and a platoon of vehicles is different, the results will vary when a platoon methodologie is used. However, the string of ACC vehicles does give some clear insights in the differences between both mitigation strategies as demonstrated in Chapter 2. Nonetheless, for future work it is recommended to implement an actual platooning structure to get more insight into different strategies and approaches the platoon can have to this problem. An example could be that the platoon optimizes its vehicles to minimize the closing time for the trailing to the leading vehicle, which could mean that the leading vehicle slows down to let the trailing vehicle close in more quickly. Furthermore, having a larger string of vehicles or platoon could also give additional insights due to the braking propagation the fallback manoeuvre initiates. Also varying the vehicle that suffers from a failure, e.g. the third vehicle in a string of four vehicles, could give additional insights into desired control actions or optimization philosophies for the remainder of the platoon. Both the additions of more vehicles and varying the faulty vehicle within the platoon are therefore recommended for future work.

Bibliography

[1] Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. 2016.

[2] ISO/DIS 26262-1: Road vehicles - Functional safety. *Geneva, Switzerland: International Organization for Standardization*, 2018.

[3] Automated vehicles for safety, Jun 2020. URL `https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety`.

[4] D. Ammon. Vehicle system dynamics challenges on the way to autonomous driving. In *4th International Munich Chassis Symposium. Hrsg. von ATZlive Springer Vieweg. Proceedings of the 4th International Munich Chassis Symposium. Wiesbaden: Springer Vieweg*, pages 37–47, 2013.

[5] I. Bae, J. Moon, and J. Seo. Toward a comfortable driving experience for a self-driving shuttle bus. *Electronics*, 8(9):943, 2019.

[6] B. Boulkroune, S. Van Aalst, E. Van Nunen, and Y. Descas. Fault detection and isolation system for four-wheels drive electric vehicles. *Conference on Control and Fault-Tolerant Systems, SysTol*, pages 110–116, 2019. ISSN 21621209. doi: 10.1109/SYSTOL.2019.8864749.

[7] D.D. Heikoop, J.C.F. de Winter, B. van Arem, and N.A. Stanton. Effects of platooning on signal-detection performance, workload, and stress: A driving simulator study. *Applied Ergonomics*, 60:116–127, 2017. ISSN 18729126. doi: 10.1016/j.apergo.2016.10.016.

[8] CEDR Conférence Européenne des Directeurs des Routes. Distance between vehicles. *CEDR report 2009/10.1 TGRoadSafety / DistancebetweenVehiclesReport*, 2010. URL `https://www.cedr.eu/download/Publications/2010/e_Distance_between_vehicles.pdf`.

[9] European Automobile Manufacturers Association EAMA. What is truck platooning? 2017. URL `https://www.acea.be/uploads/publications/Platooning_roadmap.pdf`.

[10] European Commission - Eurostat. Freight Transport Statistics. *Publications Office of the European Union*, (June 2019):1–12, 2015. URL `http://epp.eurostat.ec.europa.eu/statistics_explained/index.php/Freight_transport_statistics#`.

[11] EuroTest. Road works 2007 guidelines in europe. URL `https://web.archive.org/web/20120311004223/http://www.eurotestmobility.net/eurotest.php?itemno=224&lang=EN`.

[12] L. Ferranti, B. Brito, E. Pool, Y. Zheng, R. M. Ensing, R. Happee, B. Shyrokau, J. F.P. Kooij, J. Alonso-Mora, and D. M. Gavrila. SafeVRU: A research platform for the interaction of self-driving vehicles with vulnerable road users. *IEEE Intelligent Vehicles Symposium, Proceedings*, 2019-June(Iv):1660–1666, 2019. doi: 10.1109/IVS.2019.8813899.

[13] Z. Gao, C. Cecati, and S.X. Ding. A survey of fault diagnosis and fault-tolerant techniques-part II: Fault diagnosis with knowledge-based and hybrid/active approaches. *IEEE Transactions on Industrial Electronics*, 62(6):3768–3774, 2015. ISSN 02780046. doi: 10.1109/TIE.2015.2419013.

[14] A. Khabbaz Saberi, Y. Luo, F. Pawel Cichosz, M. Van Den Brand, and S. Jansen. An approach for functional safety improvement of an existing automotive system. *9th Annual IEEE International Systems Conference, SysCon 2015 - Proceedings*, pages 277–282, 2015. doi: 10.1109/SYSCON.2015.7116764.

[15] E. Khalastchi and M. Kalech. On fault detection and diagnosis in robotic systems. *ACM Computing Surveys*, 51(1):1–24, 2018. ISSN 15577341. doi: 10.1145/3146389.

[16] L. Konstantinopoulou, A. Coda, and F. Schmidt. Specifications for multi-brand truck platooning. In *ICWIM8, 8th International Conference on Weigh-In-Motion*, pages 8–p, 2019.

[17] K.Y. Liang, J. Mårtensson, and K.H. Johansson. Heavy-Duty Vehicle Platoon Formation for Fuel Efficiency. *IEEE Transactions on Intelligent Transportation Systems*, 17(4):1051–1061, 2016. ISSN 15249050. doi: 10.1109/TITS.2015.2492243.

[18] J. Lioris, R. Pedarsani, F.Y. Tascikaraoglu, and P. Varaiya. Platoons of connected vehicles can double throughput in urban roads. *Transportation Research Part C: Emerging Technologies*, 77:292–305, 2017. ISSN 0968090X. doi: 10.1016/j.trc.2017.01.023.

[19] Y. Luo, A. K. Saberi, T. Bijlsma, J. J. Lukkien, and M. van den Brand. An architecture pattern for safety critical automated driving applications: Design and analysis. In *2017 Annual IEEE International Systems Conference (SysCon)*, pages 1–7, 2017. doi: 10.1109/SYSCON.2017.7934739.

[20] J.M. Maciejowski. *Predictive control: with constraints*. Pearson education, 2002.

[21] T. Ming, W. Deng, S. Zhang, and B. Zhu. MPC-Based Trajectory Tracking Control for Intelligent Vehicles. In *SAE Technical Paper*. SAE International, 04 2016. doi: 10.4271/2016-01-0452. URL `https://doi.org/10.4271/2016-01-0452`.

[22] O. op den Camp and J. van de Sluis. Concept framework for the safety assessment of platooning trucks enabled by v2v communication. *arXiv preprint arXiv:2007.08193*, 2020.

[23] J. Ploeg, B. T. M. Scheepers, E. van Nunen, N. van de Wouw, and H. Nijmeijer. Design and experimental evaluation of cooperative adaptive cruise control. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 260–265, 2011. doi: 10.1109/ITSC.2011.6082981.

[24] Rijkswaterstaat. Richtlijn ontwerp autosnelwegen 2017: Roa2017. 2017. URL `https://puc.overheid.nl/doc/PUC_154078_31/1/#8bbb2533-84fc-499b-bb41-15fb41f84fb7`.

[25] A. K. Saberi, J. Vissers, and F. P. A. Benders. On the impact of early design decisions on quality attributes of automated driving systems. In *2019 IEEE International Systems Conference (SysCon)*, pages 1–6, 2019. doi: 10.1109/SYSCON.2019.8836917.

[26] SARTRE. Safe road trains for the environment; developing strategies and technologies to allow vehicle platoons to operate on normal public highways with significant environmental, safety and comfort benefits. 2012. URL `https://cordis.europa.eu/project/id/233683`.

[27] B. Shyrokau. *Vehicle Dynamics A; Lateral Motion I - Steady-state Handling*. Delft University of Technology; 3ME, 2019.

[28] A. Soni and H. Hu. Formation control for a fleet of autonomous ground vehicles: A survey. *Robotics*, 7 (4):67, 2018.

[29] L. Svensson, L. Masson, N. Mohan, E. Ward, A.P. Brenden, L. Feng, and M. Törngren. Safe Stop Trajectory Planning for Highly Automated Vehicles: An Optimal Control Problem Formulation. *IEEE Intelligent Vehicles Symposium, Proceedings*, 2018-June(Iv):517–522, 2018. doi: 10.1109/IVS.2018.8500536.

[30] M. Törngren, X. Zhang, N. Mohan, M. Becker, L. Svensson, X. Tao, D. Chen, and J. Westman. Architecting safety supervisors for high levels of automated driving. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1721–1728, 2018. doi: 10.1109/ITSC.2018.8569945.

[31] C. van der Ploeg, M. Alirezaei, N. van de Wouw, and P.M. Esfahani. Multiple faults estimation in dynamical systems: Tractable design and performance bounds. *arXiv preprint arXiv:2011.13730*, 2020.

[32] E. van Nunen, F. Esposto, A. K. Saberi, and J. Paardekooper. Evaluation of safety indicators for truck platooning. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1013–1018, 2017. doi: 10.1109/IVS.2017.7995847.

[33] L. Xiao and F. Gao. Practical string stability of platoon of adaptive cruise control vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1184–1194, 2011. doi: 10.1109/TITS.2011.2143407.

[34] W. Xue, B. Yang, T. Kaizuka, and K. Nakano. A Fallback Approach for an Automated Vehicle Encountering Sensor Failure in Monitoring Environment. *IEEE Intelligent Vehicles Symposium, Proceedings*, 2018-June (Iv):1807–1812, 2018. doi: 10.1109/IVS.2018.8500392.

[35] H. Yang, Q.L. Han, X. Ge, L. Ding, Y. Xu, B. Jiang, and D. Zhou. Fault-tolerant cooperative control of multiagent systems: A survey of trends and methodologies. *IEEE Transactions on Industrial Informatics*, 16(1):4–17, 2020. ISSN 19410050. doi: 10.1109/TII.2019.2945004.

[36] Y. Ye. *Interior point algorithms: theory and analysis*, volume 44. John Wiley & Sons, 2011.

[37] J. Yu and F. Luo. Fallback Strategy for Level 4+ Automated Driving System. *2019 IEEE Intelligent Transportation Systems Conference, ITSC 2019*, pages 156–162, 2019. doi: 10.1109/ITSC.2019.8917404.