CRANFIELD UNIVERSITY

CHENHUI LIN

PATH TRACKING CONTROL OF A MULTI-ACTUATED AUTONOMOUS VEHICLE AT THE LIMITS OF HANDLING

SCHOOL OF AEROSPACE, TRANSPORT AND MANUFACTURING
Transport Systems

PhD
Academic Year: 2020–2021

Supervisor: Dr E. Velenis
Associate Supervisors: Dr S. Longo, Dr E. Siampis
June 2021

CRANFIELD UNIVERSITY

SCHOOL OF AEROSPACE, TRANSPORT AND MANUFACTURING
Transport Systems

PhD

Academic Year: 2020–2021

CHENHUI LIN

Path Tracking Control of a Multi-Actuated Autonomous Vehicle at the Limits of Handling

Supervisor: Dr E. Velenis
Associate Supervisors: Dr S. Longo, Dr E. Siampis
June 2021

This thesis is submitted in partial fulfilment of the requirements for the degree of PhD.

Abstract

Over the past few decades, autonomous vehicles have been widely considered as the next generation of road transportation. As a result, relevant technology has been rapidly developed, and one specific topic is enabling autonomous vehicles to operate under demanding conditions. This requires the autonomous driving controller to have a good understanding of the vehicle dynamics at the limits of handling, and is expected to improve the performance as well as safety of autonomous vehicles especially in extreme situations. Furthermore, there has been application of techniques such as torque vectoring and four-wheel steering on modern vehicles as part of the driver assistance system, while such multi-actuation can be deployed on an autonomous vehicle in order to further enhance its performance in response to challenging manoeuvres and scenarios.

This thesis aims to develop a real-time path tracking control strategy for an autonomous electric vehicle at the limits of handling, taking advantage of torque vectoring and four-wheel steering techniques for the enhanced control of vehicle dynamics. A nonlinear model predictive control formulation based on a three degree-of-freedom vehicle model is proposed for control design, which takes into account the nonlinearities in vehicle dynamics at the limits of handling as well as the crucial actuator constraints. In addition, steady-state references of steering inputs as well as vehicle states are generated based on a bicycle model and included in the control formulation to improve the performance. Two path tracking models with different coordinate systems are introduced to the control formulation, and compared to understand the more suitable one for the proposed path tracking purpose. Then the path tracking performance with different levels of actuation

is investigated. According to the high-fidelity simulation results, the vehicle achieves the minimum lateral deviation with the over-actuation topology including both torque vectoring and four-wheel steering, which illustrates that the over-actuation formulation can enhance the path tracking performance by enduing the vehicle with the best flexibility as well as stability during operation at the limits of handling.

Before being implemented on the vehicle, the performance of the proposed control strategy is further assessed with regards to real-time operation. After evaluating the control performance with different prediction horizons and sampling time, the most suitable setup is identified which compromises between the control performance and the capability of real-time execution. Finally, the control algorithm is implemented on a real vehicle for practical testing. The controller is tested in four different scenarios, and the results demonstrate that the proposed controller is capable of path tracking control and vehicle stabilisation for multi-actuated autonomous vehicles at the limits of handling.

In general, this thesis has proposed a path tracking controller for autonomous vehicles which takes into account nonlinear vehicle dynamics at the limits of handling. Following some necessary simplification, the developed controller has been successfully deployed on a real vehicle in real time, and the control performance has been validated in several challenging scenarios. The controller proves itself to be able to improve the vehicle's flexibility as well as to stabilise the vehicle at the limits of handling, and furthermore, it is able to accommodate relatively large side slip angles during the demanding manoeuvres as well.

Keywords

Autonomous vehicle; multi-actuation; torque vectoring; predictive control.

Contents

List of Figures

List of Tables

List of Acronyms

ABS	Anti-lock Braking System
AGV	Autonomous Ground Vehicle
BBW	Brake-By-Wire
CAN	Controller Area Network
CTSNF	Cartisian Formulation
COG	Centre of Gravity
CVLRF	Curvilinear Formulation
DOF	Degree-Of-Freedom
DYC	Direct Yaw Moment Control
ECU	Electronic Control Unit
EOM	Equation of Motion
ESP	Electronic Stability Programme
EV	Electric Vehicle
me FWS	Front Wheel Steering
GPS	Global Positioning System
GNSS	Global Navigation Satellite System
IMU	Inertial Measurement Unit
INV	Inverter
LQG	Linear Quadratic Gaussian
LQR	Linear Quadratic Regulator
LTV	Linear Time-varying

MF	Magic Formula
MIL	Model-In-the-Loop
MIMO	Multi-Input Multi-Output
MPC	Model Predictive Control
MUEAVI	Multi User Environment for Autonomous Vehicle Innovation
NLP	Nonlinear Program
NMPC	Nonlinear Model Predictive Control
PID	Proportional Integral
PID	Proportional Integral Derivative
PIL	Processor-In-the-Loop
QP	Quadratic Programming
RMS	Root Mean Square
RTK	Real-time Kinematic Positioning
RWD	Rear Wheel Drive
RWS	Rear Wheel Steering
SBW	Steer-By-Wire
SSR	Steady-State Reference
TV	Torque Vectoring
4WS	Four Wheel Steering

List of Symbols

$\alpha_F,\ \alpha_R$	Slip angle of front and rear tyres
β	Vehicle side slip angle at its centre of mass
$\delta_F,\ \delta_R$	Steering inputs at the front and rear wheels
ε_Ψ	Lateral deviation of the vehicle from reference path
ε_V	Tracking error of velocity
ε_y	Heading angle deviation of the vehicle from orientation of reference path
κ	Curvature
μ	Friction coefficient
Ψ	Yaw angle of the vehicle
$a_x,\ a_y$	Longitudinal and lateral acceleration of the vehicle at its centre of gravity
B	Stiffness factor of the Pacejka's Magic Formula
C	Shape factor of the Pacejka's Magic Formula
$C_F,\ C_R$	Cornering stiffness of the front and rear tyres
D	Peak factor of the Pacejka's Magic Formula
$f_x,\ f_y,\ f_z$	Longitudinal, lateral and vertical tyre force

g	Constant of gravitational acceleration
h	Height of the vehicle's centre off gravity from the ground
I_z	Moment of inertia of the vehicle about its vertical axis
l_F, l_R	Distances from the vehicle's centre of gravity to its front and rear axle
m	Mass of the vehicle
M	Yaw moment
r	Yaw rate of the vehicle about its centre of gravity
R_w	Tyre radius of the vehicle
S	Travelled distance of the vehicle along the reference path
t_p	prediction horizon of the controller
t_s	Sampling time of the controller
t_solve	Computation time of the controller
T_F, T_{RL}, T_{RR}	Torque input on the front axle, rear left wheel and rear right wheel
u	Input vector of the control system
V_x, V_y	Longitudinal and lateral velocity of the vehicle at its centre of gravity
w_L, w_R	Distances from the vehicle's centre of gravity to the centre of its left and right wheels
x	State vector of the control system
X	X positiosn of the vehicle in the Cartisian coordinate system
Y	Y positiosn of the vehicle in the Cartisian coordinate system

Acknowledgements

First of all, I would like to express my deepest gratitude to my supervisor Dr Efstathios Velenis for giving me the opportunity to pursue my PhD at Cranfield University, and for his guidance and support throughout my study. I would also like to appreciate the advice and help I have received from my associate supervisors Dr Stefano Longo and Dr Efstathios Siampis. I wouldn't be able to complete my PhD without them.

The practical testing is a quite challenging part of my research work, and it is my fortune to receive the huge assist and support from our school. Massive thanks to Dr James Brighton, Dr Kim Blackburn, Ms Gracinda Ferreira, Mr Simon Stranks and Dr Terence Richards. Furthermore, I do appreciate Delta Motorsport for providing such a great vehicle platform so that I can validate my research work in practical testing. Especially I would like to say thank you to Mr Celso Acurcio and Dr Nathan Ewin from Delta Motorsport for the instructions and assist on platform operation.

I would also like to thank everyone from Advanced Vehicle Engineering Centre that I have collaborated with over the past three years. Thank you all for sharing your knowledge and experience with me, and for giving me advice when I face problems.

Last but not least, a special thank you to my family, especially to my wife Lihong for all the love and support.

Chapter 1

Introduction

1.1 Overview

Over the past few decades, due to the increasing demands to improve the safety, efficiency and comfort of road vehicles, autonomous ground vehicles (AGVs) have been widely considered as the next generation of road transportation. AGV is often used to describe vehicles that are able to perform part or all of the dynamic driving tasks, including driving environment perception, manoeuvre planning and vehicle motion control in both longitudinal and lateral direction [1]. Figure 1.1 shows a schematic view of the operation of an AGV.

With regards to the development of AGVs, the control techniques for AGVs have been rapidly developed with a great deal of research work carried out for various objectives, including path and motion planning [2, 3, 4], path tracking [5, 6], obstacle detection and avoidance [7, 8] and so forth. Among these topics, the path tracking control, as one of the fundamental functions of AGVs, is the main focus of this thesis.

The dynamical capability of a vehicle is basically determined by the maximum available tyre-road friction force. A large number of car accidents are caused by inexperienced

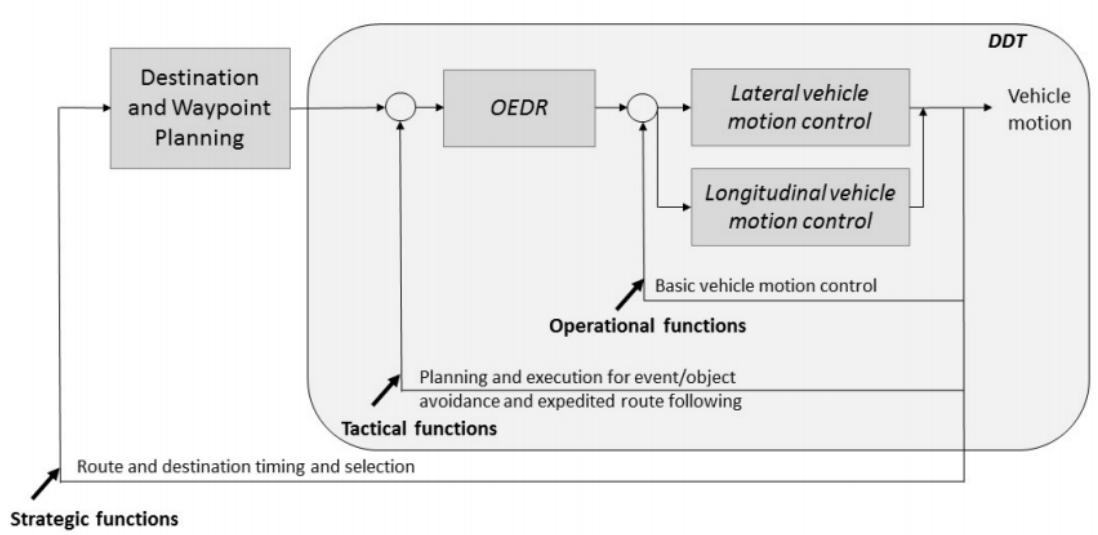


Figure 1.1: Schematic view of the operation of an AGV. [1].

drivers carrying out aggressive manoeuvres that exceeds the friction limits, and in response to that, techniques such as electronic stability programme (ESP) have been developed to improve a vehicle's stability. While on the other hand, a professional driver is able to operate the car at its friction limits, which extends the mobility of the vehicle in extreme conditions. Inspired by this, if an autonomous vehicle is able to make full use of its dynamical capability, the path tracking performance can be maximised. Hence, this thesis is particularly focused in the path tracking and stability control of autonomous vehicles at the limits of handling.

1.2 State of the Art

Regarding the path tracking problem, geometric path tracking controllers are able to track a path only with the geometry of vehicle kinematics and of the reference path, but they are less suitable for control at the limits of handling due to the lack of knowledge on vehicle dynamics. In comparison, control methods that involve vehicle dynamics such as H-infinity technique and sliding mode controllers are able to provide more robust control performance. In addition to these, optimal control theories such as linear quadratic regula-

tor (LQR) has also been introduced to the scope, and are able provided better performance in terms of tracking deviation in some research [9].

Among various control methods, model predictive control (MPC) is found as an effective technique for autonomous vehicle control. It is able to achieve better control performance than geometric and LQR controllers, and is more suitable for multi-variable systems. Futhermore, MPC has shown its advantage in dealing with constraints such as state and input limits or road boundaries. Hence, it has been widely utilized for autonomous vehicle control. However, despite the outstanding control performance and robustness of MPC, it requires more computation time, which make it difficult to be implemented in real time. In some studies, linear models are used to simplify the control formulation, but such simplification is not sufficient for the operation at the limits of handling, when the vehicle behaviour becomes highly nonlinear.

In addition to active FWS control, techniques like DYC and torque vectoring (TV) have been extensively discussed in the literature for vehicle control. TV refers to the technique that is able to manipulate the individual torque on each wheel. According to some research, TV can effectively improve the cornering response at the limits of handling by generating a direct yaw moment to stabilise the vehicle. Hence, TV has been applied in several studies for the development of driver assist systems, aiming to improve the vehicle performance and guarantee consistently safe and stable cornering response. In recent years, there are more studies including four wheel steering (4WS) into the actuation formulation for path tracking control design, and demonstrate that 4WS is able to improve the stability and flexibility of the vehicle. To the best of our knowledge, there are few studies on the multi-actuation control including both TV and 4WS in the actuation topology for path tracking purpose, which is part of the novelty of this thesis. In addition, the real time implementation of such a controller is also in the scope.

1.3 Aim and Contributions

The aim of this research is to develop a real-time implementable control strategy for a multi-actuated autonomous electric vehicle, for the purpose of path tracking under the condition of limits of handling. It is worthy to be mentioned that although the control strategy is supposed to be implemented on a vehicle with specific actuation topology, it is developed in a general formulation and thus it can be easily extended to fit other vehicles with different topology.

In order to realise this aim, following contributions has been made in this research:

- develop an optimal control strategy for the specific vehicle actuation topology, with a more suitable formulation between two alternatives for optimal path tracking performance,
- compare the path tracking performance of the vehicle with different level of actuation in simulation,
- validate and modify the control strategy in real-time simulation,
- implement the control algorithm on the vehicle platform and validate the control performance in practical testing.

This research first compares predictive control with different path tracking formulations to understand the most suitable one for the proposed path tracking purpose. Then the path tracking performance with different level of actuation is investigated. According to the high-fidelity simulation results, the vehicle achieves the minimum lateral deviation with the over actuation topology including both torque vectoring and four-wheel steering, which illustrates that the over actuation topology can enhance the path tracking performance by enduing the vehicle with the best stability and flexibility during extreme operation.

Before being implemented on the vehicle, the performance of the proposed control strat-

egy is further assessed in real-time simulation. After evaluating the control performance with different sampling time and prediction horizons, the best setup is identified for compromise between good control performance and the capability of real-time execution. Finally, the control algorithm is implemented on a vehicle platform for practical testing. The testing results demonstrate that the developed controller is capable of autonomous vehicle control at the limits of handling with good path tracking and vehicle stabilisation performance.

1.4 Thesis Outline

This thesis is organised as follows. Chapter 2 discusses the main contribution in literature on path tracking control, vehicle dynamics control at the limits of handling and the application of multi-actuation, with a particular interest on real-time implementation. In Chapter 3, the modelling of the vehicle dynamics and the generation of steady-state references are introduced. Chapter 4 demonstrates the formulation of the optimal control problem based on NMPC as well as the development of the optimal path tracking control strategy, and discusses about the control performance with different path tracking formulation. The control strategy is further studied in Chapter 5 by comparing the path tracking performance with different levels of actuation. Before being implemented for practical testing, the control strategy is evaluated in Chapter 6 for real-time executability. The control performance under various sampling time and prediction horizons are compared to make compromise between short computation time and good prediction of system dynamics. Finally, the developed control algorithm is implemented on the vehicle platform, and Chapter 7 shows the results of the practical testing. In the end, The research outcome is summarised in Chapter 8.

Chapter 2

Literature Review

2.1 Introduction

In this chapter, the relevant context to this research is elaborated including the work in academia as well as in the automotive industry. The literature review is particularly focused on three topics, corresponding to the following three sections in this chapter. The first section is in regards to the path tracking control of autonomous vehicles. Various control methods are discussed and compared to recognise an appropriate approach for this research. In the next section, studies in vehicle dynamic control at the limits of handling are reviewed, including the scenarios of vehicle stability control, and the application in autonomous racing. Here interest lies in how different control approaches can contribute to vehicle stabilisation at extreme conditions, and how the performance is evaluated. At last, the application of multi-actuation is reviewed. This includes the utilisation of TV and RWS, in both driver assistance systems and autonomous vehicles. It is concerned how the application of such multi-actuation techniques can lead to enhanced vehicle performance. In addition, another point of interest is put on the possibility of real-time implementation of the studies, as this is one of the objectives of this research.

2.2 Path Tracking Control of Autonomous Ground Vehicles

Corresponding to the motion control layer in Introduction, path tracking is one of the fundamental tasks of AGVs. Among the various path tracking control methods, geometry-based methods are the most basic ones. This kind of approaches is based on the geometry of the vehicle and path, with the assumption of zero slip angles on the wheels [10]. Coulter raised the pure pursuit steering control algorithm for the path tracking purpose [11]. With only the geometric relationship between vehicle position and a look ahead "goal point", the algorithm determined steering commands by calculating the curvature of the desired arc guiding the vehicle towards the goal point. This algorithm had been proven effective in practical application, but some limitations were addressed, one of which was lack of robustness during rear wheel slip due to a sharp change of curvature at high speed. Another geometric approach for path tracking was often called Stanley controller, named after the autonomous robot vehicle Stanley, which applied the method and won the 2005 DARPA Grand Challenge [12]. The Stanley controller is generally based on a nonlinear feedback function of the cross-track error, which consisted of the heading error and lateral distance error. When the cross-track error became zero, the controller would apply a steering angle aiming to point the front wheels parallel to the path. In comparison with the pure pursuit controller, the Stanley controller was more suitable for higher speed. Apart from purely geometric control methods, there have been approaches based on kinematic vehicle models as well [13, 14]. In [14], a chained form controller was developed based on a kinematic bicycle model. According to the simulation, the controller achieved similar results as the geometric controllers, with reduced performance at high speed compared with the Stanley controller. In summary, the above control methods were developed purely based on vehicle kinematics and path geometry. They were simple and effective for AGV navigation in some conditions, but due to the lack of knowledge on vehicle dynamics, they were less suitable for control at the limits of handling as the derived control

actions can be infeasible or lead to instability.

In addition to the geometric and kinematic control approaches, simple feedback control methods such as PID can also be used for path tracking control. It compares the system output and the desired reference, and applies a control law based on the error. Due to its simple theory and design, PID is broadly used in the industry [15, 16, 17, 18]. In terms of its application for path tracking control, Marino et al. proposed a steering control strategy for autonomous lane keeping [19]. A PID controller was developed to generate the yaw rate reference on the basis of lateral error, and the steering angle was determined by a second PI controller based on the yaw rate tracking error. In [18], a path tracking controller was designed for autonomous driving. In this approach, the steering command was produced by applying the PID control law on the cross-track error of the vehicle. PID control works without a system model, and is generally triggered by the error between the feedback states of the system and the corresponding desired values. This theory makes PID easy to be designed and implemented. However, PID can hardly take into account the physical characteristics or limitations of the system, so the system stability and control performance can not be guaranteed especially when the vehicle is operating with a significant deviation from the reference. This lack of robustness could be compensated by applying an adaptive control law to automatically tune control parameters in response to the changing condition [20], while tuning is another problem with PID. There has not been a theory on the tuning of PID, or criteria for assessing if the optimal performance is achieved. Hence, the tuning of PID can be quite difficult [15, 18].

More researches have been done on the development of dynamic controllers for autonomous path tracking, which incorporate vehicle dynamics in the control design. In [14], geometric, kinematic and dynamic controllers were compared in the aspects such as path requirements, robustness and steady-state error, and it was shown that dynamic controllers were able to achieve better path tracking performance in general. Roselli et al. proposed a path tracking steering controller for lane-keeping at a constant speed [21].

The controller was based on the H-infinity technique, and was formulated with interest on the lateral tracking error and the derivative of a look-ahead error. According to the experimental tests, the H-infinity controller was able to achieve overall smaller lateral error in comparison with a PID controller with feedforward, leading to a smoother action in the corner. Tagne et al. developed a higher-order sliding mode controller for lateral control of AGV [22]. It aimed to stabilise the vehicle and enhance the path tracking performance especially at high speed, while with a special goal of reducing chattering. The developed controller was validated in several scenarios with lateral acceleration up to 7 m/s^2, and showed its robustness to speed variation, curvature change and vehicle parameter uncertainty. In [23], a feedback-feedforward steering controller was designed to maintain vehicle stability and to minimise the path tracking deviation at the limits of handling. The controller took into account the combination of the desired path heading and vehicle sideslip information to produce the feedforward steering command, which was integrated with the feedback component of the control strategy looking at a predicted steady-state path tracking error based on a linear vehicle model. The controller was implemented on an Audi TTS and validated in experimental tests, during which the controller showed good robustness, and the vehicle tracked the reference path with lateral error up to 2m, at the limits of handling indicated by the 9 m/s^2 lateral acceleration.

The above dynamic controllers have shown the advantage of including vehicle dynamics in path tracking control design that enhances the robustness and improves the tracking performance. However, they all used single-track bicycle models for lateral vehicle dynamics. This helps to simplify the control formulation, but due to the fact that the difference between the left and right wheels is neglected, it may not be sufficient in comparison with a more detailed vehicle model for the control of extreme cornering manoeuvre where that difference becomes significant [15]. Hence, full vehicle models have been applied for control design as well [24, 25, 26, 27]. In [25], a control system for AGVs was developed for stability control and path tracking at high speed. The controller was based on a full vehicle model, and on sliding mode control to handle the uncertainties from un-modelled

dynamics and road condition. The proposed controller was evaluated in simulation, during which the vehicle maintained stability when driving along a general path at a constant speed of 50 kph, with lateral tracking error up to 0.06m. In [26], an H-infinity output-feedback control strategy was designed for AGVs. It was supposed to carry out path tracking without the information of the lateral velocity of the vehicle. The simulation results showed that the controller was able to maintain robustness and achieve good path tracking performance within 1 m under the condition of the given uncertainty. Chen et al. presented a hierarchical control strategy for autonomous path tracking without the information of friction condition or external disturbance [27]. A full vehicle model was applied in the control design, while separated into two modules in order to deal with motion control and wheel slip dynamics respectively. These studies demonstrated that with a better knowledge of the vehicle dynamics during high-speed turning, robustness can be maintained even with the presence of uncertainty and disturbance. Hence, the utilisation of full vehicle models in control design can improve the path tracking performance to a higher level.

In addition to the control methods mentioned above for path tracking, optimal control theories have also been introduced to the scope. The linear quadratic control is a typical optimal control formulation. It describes the desired control target as a cost in the form of a quadratic function, and the control law aims to bring the cost function to the minimum based on the linear state feedback [28]. LQR was applied in [29] for autonomous path tracking control based on a bicycle model. The study showed that in comparison with ordinary PD control which could not provide satisfying performance in both response and convergence, the developed LQR controller was able to provide better path tracking performance. Sharp et al. proposed a mathematical model for the steering control of an autonomous vehicle [30], and the work was extended in [31] where an optimal steering control approach for path tracking was developed. The control design was based on LQR, and the path tracking problem was represented as optimal preview control in the local driver view. Different cost functions were attempted with priority on minimising position

error or attitude angle error. The former formulation provided a better path tracking accuracy with more oscillation on the steering, while the latter formulation led to smoother yaw motion with larger path tracking error. Extended from LQR, linear quadratic Gaussian control has also been used for path tracking. Lee et al. proposed an optimal path tracking controller for AGVs [9]. The controller was based on LQG, and thus was able to handle the noise of system states during operation. The controller was implemented on a test vehicle, and provided better performance with less tracking deviation than geometric controllers in both low and high speed conditions. Another LQG path tracking controller was developed in [32]. With the application of an augmented state space model, the path tracking performance was improved compared with conventional LQG controllers.

Since its origin in the late 1970s, MPC has had a significant influence on control engineering [33, 34]. Taking advantage of a dynamic model of the system, MPC is capable of predicting the system response within a finite horizon, and determining a series of the optimal control actions to minimise the gap between the system output and the desired reference. Among various control methods, MPC is found as an outstanding technique. In [35], the path tracking control performance of geometric, LQR and MPC controllers were compared in simulation, and the results showed that MPC achieved the best path tracking performance among the three approaches, with the minimum control effort. Yakub et al. extended the study by comparing the path tracking performance of MPC and linear quadratic control for different speed, tyre-road friction coefficient and control topology including FWS, four-wheel steering (4WS) and FWS with direct yaw moment control (DYC) [36]. A similar conclusion was drawn that MPC was more suitable for multivariable systems. Hence, MPC has been widely used for autonomous vehicle control, including path tracking control. In [37], a linear MPC control structure was presented for path tracking control. Alessandretti et al. developed two MPC control algorithms for path-following and trajectory tracking control of constrained underactuated vehicles [38]. The effect of MPC was evaluated in [39] for autonomous path tracking under crosswind condition. The simulation results demonstrated that MPC was capable of maintaining ro-

bustness given the disturbance produced by side wind. In addition, it was also shown that with a more detailed vehicle model, the tracking performance as well as vehicle stability were considerably improved in conditions with different friction.

Compared with other control approaches, one advantage of MPC is that it is able to handle complicated system models. Linear vehicle models may not be sufficient when the vehicle is operating at the limits of handling, during which the nonlinearity of system dynamics becomes dominant. Thus, nonlinear models are critical to provide more realistic description of vehicle dynamics, which is able to enhance the performance of MPC due to its theory. Falcone et al. developed an NMPC controller for an FWS autonomous vehicle [40]. Despite the application of a bicycle model, the nonlinearities related to lateral vehicle stabilisation were taken into account. Experimental results showed that with the proposed controller, the vehicle was able to track a double lane change trajectory at the speed of 21m/s on icy roads. Berntorp et al. developed an MPC trajectory tracking controller which was adaptive to varying friction conditions [41]. This shows the advantage of MPC in providing the robust control performance with the change of conditions.

Furthermore, MPC has also shown its advantage in dealing with constraints from state and input limits as well as road boundaries [41, 42, 43, 44, 4]. In [42], an NMPC strategy was developed for autonomous path tracking control. The controller was supposed to track the reference path while respecting the state constraints and steering actuator limitation depending on the velocity. The controller was implemented for practical testing, in which the effectiveness of path tracking capability was demonstrated at speed up to 40 kph, while none of the constraints was violated. In [43], a hierarchical NMPC control structure was produced for autonomous vehicle control. The control scheme took into account the acceleration limits of the vehicle determined by the friction circle, as well as the feasible region of driving torque on wheels. Chen et al. developed a controller focusing on path tracking while dealing with cut-in vehicles [44]. In addition to the constraints of vehicle dynamical states and actuator limitations, vehicle position was also included

in the boundaries to provide good path tracking performance while maintaining a safe distance from the cut-in vehicle.

Despite the advantage of MPC, it is computationally expensive for practical implementation. In [45], it was reflected in simulation that the MPC controller with more complicated model was more time consuming and more difficult to tune. As a result, in order to implement MPC in real time, a powerful enough computing platform is required, otherwise simplification should be carried out to ease the computation burden. This could be reduced system complexity [41], or limited application condition [40]. In recent years, MPC based on neural network has been proposed for path tracking problem [46, 47]. Instead of a detailed but complicated system model, a neural network model was applied in the MPC formulation, which can significantly reduce the system complexity. However, this kind of approaches requires initial training, so it is lack of flexibility in different scenarios.

2.3 Vehicle Control at the Limits of Handling

The handling limits of vehicles is often defined as the situation where the vehicle's full dynamical capability is used, namely the available tyre forces are exploited. Operating vehicles at the limits of handling is not only for the best performance, but also for providing a potential solution in emergency scenarios. The first part of this section discusses about the approaches in literature for vehicle stabilisation at the limits of handling, and the second part particularly focuses on the autonomous vehicle control for aggressive driving manoeuvres.

2.3.1 Vehicle Stability Control at the Limits of Handling

Limitations exist in the vehicle handling capability due to the tyre-road friction as well as physical limits of actuators. Professional drivers are capable of operating the vehicle at these handling limits without getting into instability, while less skilled drivers may exceed the limits probably due to emergency manoeuvring or aggressive intension, sometimes leading to an accident. This is the reason why vehicle stability control is vital, and has been massively developed in the industry as part of the driver assist system. Generally speaking, the purpose of vehicle stability control is to guarantee realisation of the intended manoeuvre. Techniques like ABS and ESP have been developed and applied for a few decades in the industry for assisting the driver to maintain vehicle stability. ABS and traction control aims to prevent the tyres from sliding significantly, while ESP aims to fulfill the required yaw moment by braking the individual wheels. In addition to these, Tchamna et al. designed a differential braking control based on sliding mode control [48]. It took yaw rate and side slip angle as control objectives, while taking into account the longitudinal dynamics of the vehicle. Kakalis et al. presented a differential throttle and brake controller to enhance the dynamic behaviour of a RWD sport vehicle [49]. This was realised by generating an asymmetric longitudinal force distribution according to the yaw rate target. These techniques have significantly improved the car stability, but the driver takes full charge of the lateral control, so persistent stability cannot be guaranteed. When it comes to AGV, steering is also included in the control authority, which helps to reach the optimal stability control.

In terms of vehicle statbility control including steering, a gain scheduled active steering controller was developed in [50] for vehicle stability control in extreme handling conditions. The controller applied a rational tyre model which took the saturation of tyre force into account, and aimed to achieve the desired yaw rate and minimal side slip angle. The controller was validated in simulation with different road friction condition, and good vehicle response was observed even if a large yaw rate is required.

More controllers for vehicle vehicle stabilisation have been developed based on MPC. Falcone et al. developed two combined steering and braking MPC controllers with vehicle models of different complexity [45]. Both controllers were tested with a lane change manoeuvre and according to the simulation results, the controller with the complicated model was able to achieve good tracking performance at both low and high speed, while the controller with the simplified model could not stabilise the vehicle at high speed. Beal et al. developed an MPC controller for vehicle stabilisation [51]. The approach took advantage of the estimation of vehicle sideslip as well as tyre-road friction coefficient to understand the vehicle dynamical limitations. An MPC envelope controller based on a linear bicycle model was developed to identify the optimal control actions to guarantee the vehicle states within the range of defined boundaries. Gao et al. presented two control frameworks based on MPC for AGVs to carry out obstacle avoidance on slippery roads [52]. One approach combined path planning and path tracking in one level, while the other approach separated the two tasks in a hierarchical structure. Both controllers applied a full vehicle model for path tracking control, and the obstacle avoidance problem was formulated into a cost function with regards to the distance between the vehicle and the obstacle. Both controllers were validated in simulation, and the hierarchical control framework was able to maintain vehicle stability at higher speed of 55kph while the single layer approach could reach 40kph for maximum. In addition, the single layer controller required longer computation time in higher speed operation. The two layer control required lower computational burden, but it was due to the application of a simplified vehicle model for path planning. Funke et al. presented a control strategy which was capable of vehicle stabilisation and obstacle avoidance during path tracking [8]. The controller utilised MPC so that it could make full use of vehicle's dynamical capability, which was critical in emergency scenarios. In terms of the potential conflict among the three objectives, priority was put on obstacle avoidance over path tracking or vehicle stabilisation so that the accident could be minimised. Lee et al. developed a MPC controller for aggressive cornering manoeuvre, which took into account the effect of a large steering

angle on the longitudinal dynamics of the vehicle [53].

2.3.2 Autonomous Vehicle Control at the Limits of Handling

One scenario of autonomous vehicle control at the limits of handling, is actively controlling the vehicle to carry out aggressive manoeuvres, which is often related to racing purpose. The idea is to make full use of the vehicle dynamical capability, which is essential in racing so that the path could be negotiated within the shortest time. Thommyppillai et al. proposed the development of a car driver model which is able to exploit full capacity of a racing car [54]. The control strategy was based on optimal linear preview control theory, and according to the simulation results, the vehicle was able to track three different paths at a fixed high speed of 30m/s. With adaptive control strategy, the lateral tracking error was significantly reduced. In [55], aggressive autonomous vehicle driving was realised by using smooth curvature polynomial spiral and g-g diagram to generate maximum safe velocity profile to keep the vehicle within its friction limits.

In terms of MPC, Kritayakirana proposed the work on developing autonomous vehicle control at the limits of handling [56]. A longitudinal controller was designed to track a desired speed profile. It included feedforward and feedback sections to take care of the maximum available acceleration per slip circle. A steering controller was developed as well which aimed to exploit the vehicle dynamical capability to track the desired path at the limits of handling. The control framework was implemented on an autonomous Audi TTS and tested at Pikes Peak. The performance was impressive as the vehicle was able to negotiate the path with mixed paved and dirt sections, while reaching the handling limits indicated by the maximum lateral acceleration at 1G. Malmir et al. developed an LTV-MPC controller for the minimum time cornering problem [57]. The controller was supposed to drive the vehicle through a circuit within minimum time. This required to make full use of the available tyre force. Ni et al. developed a path tracking controller for an autonomous electric race car that won the 2017 Formula Student Autonomous

Competition. The controller was based on H infinity control, and managed to operate the vehicle at a random track at the limits of handling. Laurense et al. proposed a controller for autonomous path tracking at the limits of handling [6]. The control scheme consisted of a longitudinal controller which calculated the speed command based on desired yaw rate, and of a steering controller dependent on the slip angle information. This control scheme was effective for the continuous operation of an autonomous vehicle at the limits of handling, with significantly reduced requirement on the friction estimation accuracy.

Alcala et al. proposed a control strategy for autonomous racing based on LPV-MPC [58]. The controller used a bicycle model for vehicle dynamics prediction, and was supposed to track the optimal trajectory provided by the path planning algorithm. The controller was successfully implemented on a RC vehicle and tested in real time. The simulation showed good tracking performance, but in the practical testing, the tracking performance was degraded to some extent potentially due to the simplification of vehicle modelling.

Liniger et al. [59] raised two inspiring control algorithms for the purpose of autonomous racing. The first controller has a hierarchical structure, including two levels carrying out path planning and path tracking separately, while the other one integrates the two tasks by formulating the problem in a single layer, so that path planning and path tracking can be done simultaneously. Both controllers are based on the LTV approximation of the system dynamics, and aim to determine the optimal manoeuvres to maximise the progress on the track, while in addition to that, obstacle avoidance is also included in the scope, and is realised by means of a high-level corridor planner, which generates convex constraints for the controller to find out the optimal path with regards to the maximum progress, while ensuring not hitting the obstacles. This work has been validated with 1:43 scale RC cars in real time. According to the experiment results, both controllers are able to operate the RC car at its handling limits, which is indicated by saturated rear tyre force, while at the same time taking care of the track borders and on-track obstacles. This well demonstrates the advantage of MPC in dealing with complicated constraints.

However, simple linear vehicle model and tyre model are used in this study to reduce the computational complexity of MPC. This works well for a 1:43 scale RC model car, but if the controller is to be implemented on a full-size vehicle, such simplification is no longer sufficient to guarantee good prediction of the vehicle dynamics at the vehicle's handling limits, where the vehicle behaviour becomes highly nonlinear.

In addition to autonomous racing, drifting is another typical manoeuvre to illustrate vehicle operation at the limits of handling. It is an oversteering manoeuvre featuring a large side slip angle, where vehicle is at the margin of stability with a loss of traction force. Velenis et al studies the steady-state equilibria of a RWD vehicle during the drifting manoeuvre with aggressive sideslip, and proposes a novel drift stabilisation controller based on LQR and backstepping control [60]. The controller was successfully validated in simulation, and provided a similar performance as the testing results provided by a real driver. The study was continued by Gonzales et al. who realised autonomous drifting on a 1/10 RC car, using LQR and vision-based state estimation techniques [61]. Apart from LQR, [62] managed to control a real vehicle to autonomously drift along a constant-radius trajectory through lookahead error regulation and novel feedback control law for sideslip stabilisation. The experimental results demonstrated high robustness and good overall performance in terms of path tracking and drifting stabilisation.

Goh et al. developed a controller for autonomous drifting control [63]. The controller was supposed to track a circle path at high side slip angle around 45 degrees, while maintain vehicle stability. The work was extended in [64], where the proposed control strategy was capable of autonomous drifting along a complex path. Zhang et al. proposed a motion control strategy which could track the reference drifting trajectory provided by a higher level path planning algorithm. The control algorithm was implemented on a RC car, and validated in the experiments.

In recent years, neural network based reinforcement learning approaches have been developed and used in academia for autonomous vehicle control at the limits of handling.

[Williams2017] introduced an information theoretic MPC algorithm for aggressive driving control on an off-road track. In this study, multi-layer neural networks were used as the system dynamics model utilised for the MPC formulation, and the experiment tests showed that the 1/5 scale rally car was able to carry out extreme manoeuvres on the off-road track with loose friction and significant disturbance. [Acosta20181] presented an autonomous drifting controller as well based on a data-based neural network approach. In this work, autonomous drifting was achieved through MPC combined with feedforward neural network structure, and the structure was trained using the data obtained from driver-in-the-loop simulation. It was impressive that with this approach, autonomous drifting could be achieved even without any tyre-road friction information, as the controller was provided with a vehicle state database containing the required drifting equilibrium. The application of neural networks ccan satisfy the requirement of high-fidelity system model to some extent, but one of the most disadvantage of this approach is that the controller requires initial training specific to scenarios and manoeuvres, which reduces the flexibility of controller for different scenarios. In addition, [Di2021] also mentioned that this kind of approaches could lead to unstable behaviour of vehicles.

2.4 Multi-Actuation Control of Autonomous Vehicles

2.4.1 Torque Vectoring

To achieve the desired yaw rate is a key criterion to identify the performance of vehicle stability and path tracking purpose. In the previous review of the literature, most studies used only front steering for yaw control. In addition to DYC as well as active FWS control, techniques like torque vectoring (TV) have been extensively discussed in the literature for vehicle control. Typically, TV refers to the differential technique which varies the torque delivered on each wheel. This makes TV different from the traditional technology like ESP that is commonly used in the automotive industry. With the rapid

development of electric vehicles (EVs), the application of independent in-wheel motors provides a more straightforward realisation of TV.

[Smith2017] analysed the effect of TV on the vehicle handling characteristics in minimum time cornering motion. The results showed that TV was able to compensate for the load transfer effects during manoeuvres with high longitudinal or lateral acceleration. Chatzikomis et al. compared a selection of path tracking controllers with and without the TV functionality, and the results confirmed that TV control could effectively improve the cornering response at the limits of handling by generating a direct yaw moment to stabilise the vehicle [65]. Hence, TV has been applied in several studies for the development of driver assist systems, aiming to improve the vehicle performance and guarantee consistently safe and stable cornering response [66, 67, 68]. In [Siampis2015], an MPC control architecture was proposed for electric vehicle stabilisation near the limits of handling. In this study, TV technique was applied on the rear axle with the use of individual wheel motors. The study demonstrated that TV could significantly improve vehicle stability by compensating the insufficient yaw moment required to negotiate the turn. In [67], a TV control algorithm was presented for electric cars with independent wheel drive as well. Based on MPC with a nonlinear vehicle model, the controller aimed to identify the optimal independent torque to be applied on each wheel that helped to achieve the required yaw motion according to the steering input. Simulation results showed that the proposed approach successfully achieved the target with the application of TV. Mikulas et al developed two TV control algorithms for a Formula Student electric racing car. Both controllers were based on MPC, one was LTV while the other was nonlinear. It was addressed that the application of TV could stabilise the vehicle near the limits of handling and achieve a smaller turning radius compared to the formulation with equal torque distribution at the same condition.

Given the improvement in vehicle stabilisation by TV, there is no doubt that AGV control can benefit from the multi-actuation formulation, referring to the integration of TV and

steering. Shim et al. developed an active steering and wheel torque controller based on MPC [69]. The higher level motion planner generated the reference trajectory in the form of double lane change manoeuvre to avoid collision with obstacles, and the path tracking controller aimed to track the generated path. Simulation results suggested the effectiveness of the proposed controller with robustness to changing obstacles. Guo et al. presented a real-time implementable path tracking controller based on MPC [70]. The controller applied an LTV system and was in the hierarchical structure, with the upper level determining the FWS angle as well as required yaw moment, and the lower level achieving that requirement through control allocation. Acosta et al. proposed a multi-actuation controller based on nonlinear vehicle and tyre models for autonomous drift control [71]. The work demonstrated the potential of combined FWS and TV to exploit the vehicle's dynamical capability. However, path tracking was excluded from the MPC strategy, and was carried out by a PID controller, which meant that the vehicle might not be able to follow complex path properly. In addition, the controller might not be able to be implemented in real-time. Ni et al. proposed a control strategy for path tracking of an AGV with TV functionality [72]. Within the control algorithm, a feedforward-feedback controller was developed to identify the required steering angle for the desired path, and a yaw moment controller based on SMC was used to prevent the vehicle from losing stability at the handling limits.

[Acosta2018] introduced a hierarchical autonomous drift control system to fully exploit the dynamical capability of a multi-actuated ground vehicle. The high-level controller included a path following controller and a drift reference generator, and the low-level controller utilises MPC to identify the optimal steering input and optimal torque on individual wheels to satisfy the vehicle state reference. The control performance was evaluated in simulation, and it was shown that with the controller, the vehicle was able to drift along tight paths while maintaining stability at high side slip angles.

2.4.2 Rear Wheel Steering

In addition to the FWS control, RWS has also been introduced to the industry. After its commercial debut in 1985 on the Nissan R31 Skyline [73], RWS has been expanded for application among various manufacturers. One of the advantages to apply RWS is that it can improve the manoeuvrability of vehicles. It has been stated that a vehicle with 4WS system can achieve a significantly smaller turning radius - 21% smaller compared to a vehicle with FWS only [74]. Thus, RWS has been widely used in large engineering vehicles. For instance, electrohydraulic rear axle steering systems have been applied for agricultural vehicles [75]. Furthermore, RWS can also improve the stability and dynamical response of vehicles. At high speeds, RWS can actively reduce the over-steering condition by applying a steering on rear wheels in the same direction as the front ones [74]. This is typically realised with a controller dependent on forward velocity mapping [73]. Due to the increased complexity in chassis structure, nowadays the RWS technique tends to be available more often in high-end sports cars such as Ferrari F12TDF, Lamborghini Urus, BMW 850 CSi, Porsche 911 GT3 and so forth, while there are normally several degrees available in the turning of rear wheels.

The effect of RWS has been extensively researched in the academia. Alexandru developed a mechanical integral steering system focusing on rear wheels, and the dynamical analysis showed that the application of RWS could significantly reduce body oscillation and improve stability during a single lane change manoeuvre at high speed [76]. Li et al. proposed a hierarchical control strategy which integrated direct yaw moment and active rear steering to enhance vehicle stability [77]. Sliding mode control was utilized to generate the required yaw moment and rear steering commands. The controller was proven effect in simulation as the stability as well as handling performance of the vehicle were improved in both slalom and double lane change manoeuvres. Zhang et al. proposed a pulsed active rear steering strategy which aimed to prevent sport utility vehicles from rolling over [78]. Instead of a constant rear steering angle, a pulsed angle was applied

regarding to the front steering angle provided by the driver, and the pulsation was based on a rollover coefficient regarding to the load transfer on left and right wheels. The controller was implemented in practical testing, and it proved that the proposed controller was able to stabilise the vehicle at extreme conditions. The work was extended in [79], where the pulse active rear steering system was applied for vehicle yaw stability control. In [39], it was concluded that the introduction of RWS were beneficial to vehicle control by reducing turning radius at low speed, enhancing handling at high speed and improving path tracking performance. He et al. proposed an H infinity coordination control scheme based on active front and rear steering to improve the lateral motion control performance of autonomous vehicle at the limits of handling [80].

Given the advantage of RWS in vehicle stabilisation as well as flexibility improvement, RWS has also been utilised for the purpose of path tracking. [Hang2021] proposed a 4WS controller for the path tracking purpose of an AGV. Taking advantage of LQR and a linear parameter variant system model, the vehicle was able to track the desired path aiming minimum lateral and heading error. Good robustness was achieved under different road friction condition. Zhang et al. proposed a path tracking control strategy for active rear steering vehicles assisting the driver's FWS commands for tracking a desired trajectory [81]. MPC was applied for the control design, and the work demonstrated that including RWS could significantly improve the tracking performance and reduce the workloads on the FWS.

By combining TV and RWS can give vehicle more flexibility and potential to stay stability. In [44], a path tracking controller for four-wheel drive (4WD) and 4WS vehicles was developed. The controller was based on linear vehicle and tyre models, and a similar hierarchical structure. The disadvantage of such formulation was the exclusion of steering from control allocation, leading to suboptimality in the results. What's more, as the higher levels were based on linear system, it was hard to guarantee feasibility, especially in extreme conditions. This could hinder the vehicle from operating at the limits

of handling. To the best of the author's knowledge, there has been less work on the over actuation formulation combining TV and RWS in control authority for the path tracking purpose, which is the novelty of this thesis.

2.5 Summary

According to the literature, there has been numerous research work done for autonomous vehicle control. Among the various control methods used for path tracking, MPC is the most outstanding approach. Based on a good knowledge of the system dynamics, MPC is able to improve path tracking performance as well as system stability. Furthermore, It is compatible with complicated system dynamics and constraints, which makes MPC the most suitable approach for this research. However, despite the advantages of MPC, the application of a complicated formulation can lead to high computational burden, which makes it more difficult for MPC compared with other control methods to be implemented in real time.

In regards to vehicle control at the limits of handling, work has been done on vehicle stability control as well as aggressive autonomous driving. It is shown that vehicle control at the limits of handling raises critical requirement on taking into account the physical limitations of vehicle dynamics, and thus MPC has again shown its advantage in limits of handling control. Furthermore, the application of multi actuation can extend the dynamical capability of vehicles, and enhance the performance at extreme conditions. It has been studied that techniques like TV can help to achieve the desired yaw motion while stabilising the vehicle at its handling limits by manipulating the driving torque on each wheel individually. Besides, the introduction of RWS can also improve the flexibility and stability of the vehicle. These techniques have been developed and applied in academia and the industry, but according to the literature review, less work has been done on integrating both TV and RWS in an over actuation formulation for autonomous vehicle control,

especially for real-time operation at the limits of handling, which defines the novelty of this thesis.

Aiming at the research gap, this thesis proposes a control strategy for autonomous vehicles at the limits of handling. The control strategy aims to exploit the entire dynamical capability of the multi actuation formulation with TV and 4WS, to reach the limits of handling. The controller is able to take into account the physical limitations of the vehicle, and to be implemented in real time.

Chapter 3

Vehicle Dynamics Modelling

3.1 Introduction

This chapter introduces the modelling of vehicle dynamics and tyre force for the development of the optimal path tracking control strategy, followed by the methodology of reference generation on vehicle states and position for the controller.

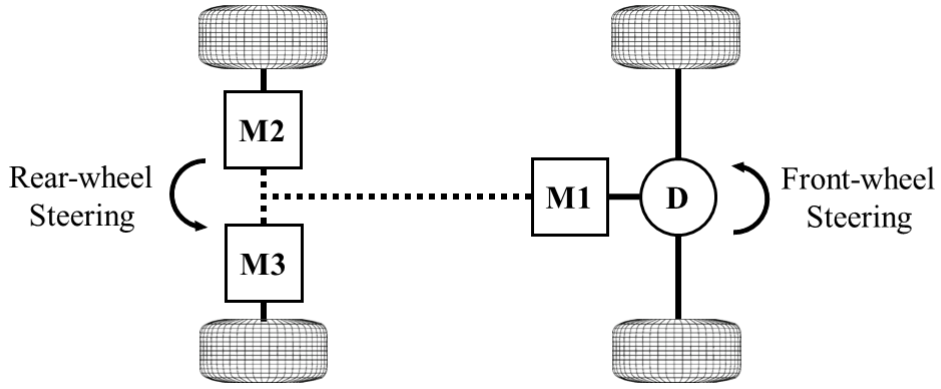


Figure 3.1: Actuator topology of the case study vehicle. M1. M2 and M3 refers to the three motors installed on the vehicle. D refers to the differential system through which M1 drives the front axle of the vehicle.

As mentioned in Chapter 1, the autonomous vehicle control algorithm developed in this research is supposed to be implemented on a vehicle platform. The platform is provided by Delta Motorsport, which is a company being expert at battery systems and platform

master control. The platform has got in-wheel hub motors on each of the rear wheels, while the front wheels are driven by a third motor through the differential system. With the individual rear motors, TV can be carried out for enhanced performance. Furthermore, the platform is equipped with 4WS capability, which improved the flexibility of the platform as well. Figure 3.1 shows the actuation topology of the platform, and more details of the platform are introduced in Chapter 7.

3.2 Vehicle Modelling

Figure 3.2 shows the coordinate frame in which the equations of motion (EOM) are expressed. It is called ”intermediate axis system” [82], which is a local coordinate system that originates at the vehicle’s COG. As a common practice to reduce inconsequential system complexity, the following assumptions are made:

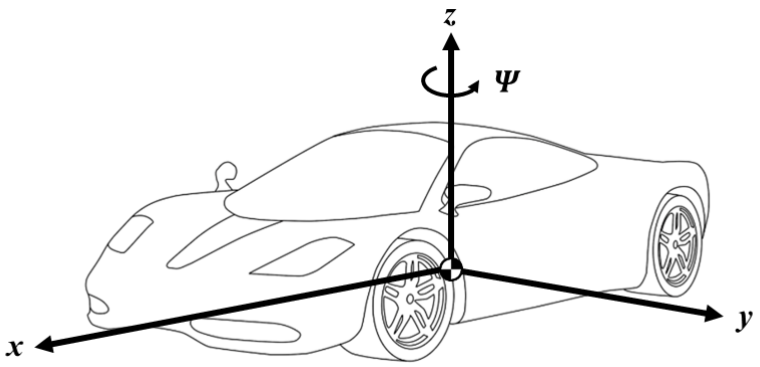


Figure 3.2: Intermediate coordinate frame.

- The vehicle travels on a horizontal plane,
- the pitch, roll and heave motion are neglected,
- the Ackerman Principle is neglected, namely the front wheels or rear wheels steers with the same angle,
- the wheel dynamics are neglected,

- the camber angles and toe angles of the wheels are zero,
- the rolling resistance is neglected,
- the aerodynamic force is neglected.

3.2.1 Equations of Motion

A three degree-of-freedom (DOF) nonlinear vehicle model is applied in this research as the modelling of system dynamics for control design. Compared with vehicle models including higher degrees of freedom, such as a 7DOF model as used in [60], the 3DOF model omits less essential information regarding this research such as wheel dynamics, which reduces the system complexity and benefits real-time implementation of the controller. The EOM is derived with Newton's Second Law in the longitudinal and lateral direction of the vehicle's frame:

$$\begin{aligned} ma_x &= \Sigma f_x, \\ ma_y &= \Sigma f_y, \end{aligned} \tag{3.1}$$

where a_x and a_y refer to the vector components of the vehicle acceleration in longitudinal and lateral direction, while Σf_x and Σf_y represent the total force applied on the vehicle in the relative direction. The yaw motion of the vehicle can be described with Euler's rotation equations:

$$I\alpha = \Sigma M, \tag{3.2}$$

where α is the angular acceleration of the vehicle, and ΣM is the total yaw moment applied on the vehicle. Based on these theorems, the EOM of the two-track vehicle model are given as follows:

$$\begin{aligned} m(\dot{V}_x - V_y r) = {} & (f_{FLx} + f_{FRx})\cos\delta_F - (f_{FLy} + f_{FRy})\sin\delta_F \\ & + (f_{RLx} + f_{RRx})\cos\delta_R - (f_{RLy} + f_{RRy})\sin\delta_R \end{aligned} \tag{3.3}$$

$$\begin{aligned} m(\dot{V}_y + V_x r) = {} & (f_{FLx} + f_{FRx})\sin\delta_F - (f_{FLy} + f_{FRy})\cos\delta_F \\ & + (f_{RLx} + f_{RRx})\sin\delta_R - (f_{RLy} + f_{RRy})\cos\delta_R \end{aligned} \tag{3.4}$$

$$\begin{aligned} I_z\dot{r} = {} & l_F \cdot (f_{FLx} + f_{FRx})\sin\delta_F + l_F \cdot (f_{FLy} + f_{FRy})\cos\delta_F \\ & - l_R \cdot (f_{RLx} + f_{RRx})\sin\delta_R - l_R \cdot (f_{RLy} + f_{RRy})\cos\delta_R \\ & - w_L \cdot (f_{FLx}\cos\delta_F - f_{FLy}\sin\delta_F) - w_L \cdot (f_{RLx}\cos\delta_R - f_{FLy}\sin\delta_R) \\ & + w_R \cdot (f_{FRx}\cos\delta_F - f_{FRy}\sin\delta_F) + w_R \cdot (f_{RRx}\cos\delta_R - f_{RRy}\sin\delta_R) \end{aligned} \tag{3.5}$$

In the above equations, V_x and V_y are the longitudinal and lateral velocity at COG, and r is the yaw rate. The vehicle's mass is represented by m, and I_z is the vehicle's moment of inertia about the vertical axis through COG. The dimensions of the vehicle are given by l_F and l_R, which stand for the distances from COG to the front and rear axle, and by w_L and w_R, which stand for the left and right portions of the track width divided by COG. The longitudinal and lateral tyre forces are denoted by F_{ijk} $(i = F,R,\ j = L,R,\ k = x,y)$. Finally, δ_F and δ_R denotes the steering angles respectively on the front and rear wheels. Figure 3.3 shows a diagram of the vehicle model.

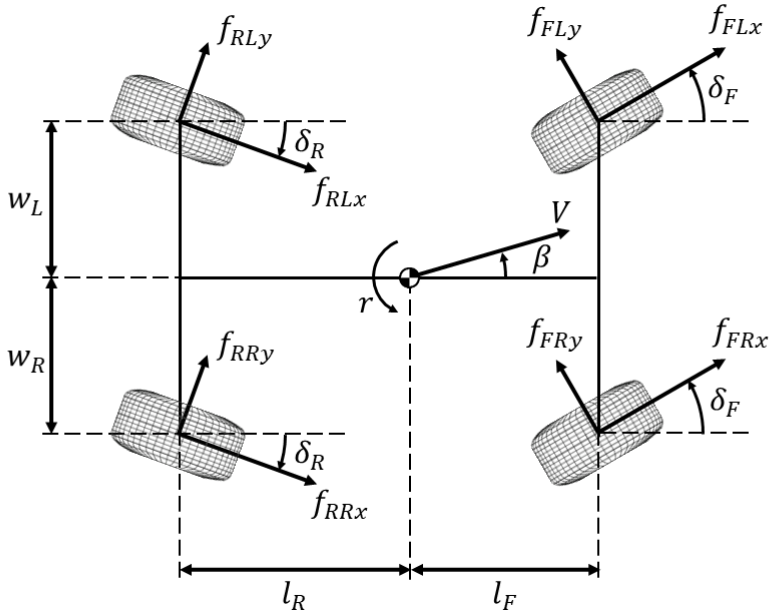


Figure 3.3: EOM diagram of the case study vehicle.

3.2.2 Load Transfer

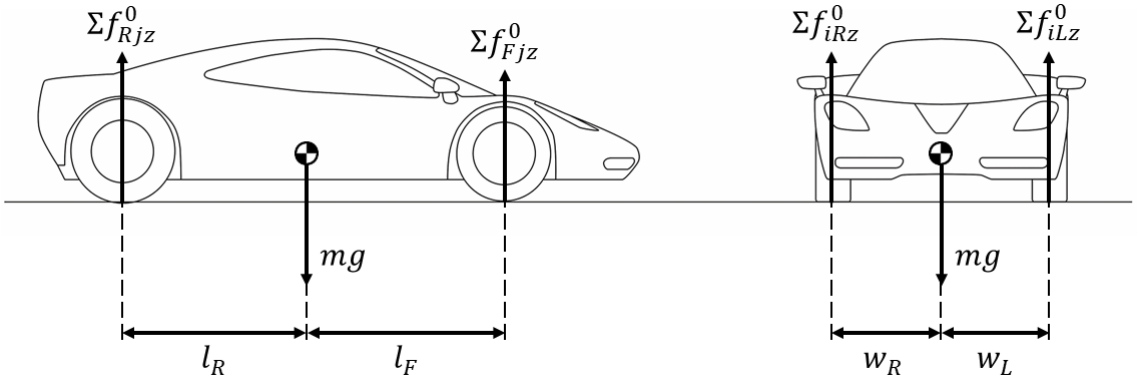


Figure 3.4: Static load distribution.

Figure 3.4 shows the static load distribution on the wheels, which can be calculated as:

$$\begin{aligned} f^0_{FLz} &= \frac{mgl_R w_R}{(l_F+l_R)(w_L+w_R)},\ f^0_{FRz} = \frac{mgl_R w_L}{(l_F+l_R)(w_L+w_R)}, \\ f^0_{RLz} &= \frac{mgl_F w_R}{(l_F+l_R)(w_L+w_R)},\ f^0_{RRz} = \frac{mgl_F w_L}{(l_F+l_R)(w_L+w_R)}, \end{aligned} \tag{3.6}$$

where f^0_{ijz} represents the static load on each tyre. However, the actual tyre load could differ from the static distribution due to longitudinal and lateral acceleration, which is called load transfer effect. The difference can be significant during operation at the limits of handling with large vehicle acceleration, so it is vital to take the load transfer effect in account. Figure 3.5 shows the dynamic load distribution on the wheels considering load transfer, and based on the assumptions of a fixed roll axis and a rigid vehicle body, the change in tyre load can be calculated as:

$$\begin{aligned} \Delta f^x_L &= \frac{mhw_R}{(l_F+l_R)(w_L+w_R)} \cdot a_x,\ \Delta f^x_R = \frac{mhw_L}{(l_F+l_R)(w_L+w_R)} \cdot a_x, \\ \Delta f^y_F &= \frac{mhl_R}{(l_F+l_R)(w_L+w_R)} \cdot a_y,\ \Delta f^y_R = \frac{mhl_F}{(l_F+l_R)(w_L+w_R)} \cdot a_y, \end{aligned} \tag{3.7}$$

where Δf^x_j and Δf^y_i represent the change of tyre load due to a_x and a_y respectively, and h denotes the height of COG from the ground.

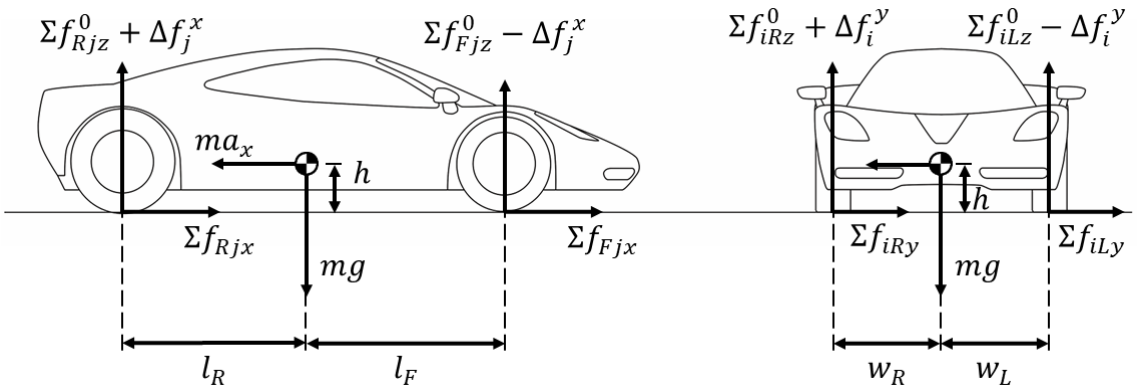


Figure 3.5: Dynamic load distribution during to longitudinal and lateral acceleration.

As a sum of the static load and the transferred weight. The total vertical load at each wheel F_{ijz} are given by:

$$\begin{aligned} f_{FLz} &= f^0_{FLz} - \Delta f^x_L - \Delta f^y_F,\ f_{FRz} = f^0_{FRz} - \Delta f^x_R + \Delta f^y_F, \\ f_{RLz} &= f^0_{RLz} + \Delta f^x_L - \Delta f^y_R,\ f_{RRz} = f^0_{RRz} + \Delta f^x_R + \Delta f^y_R, \end{aligned} \tag{3.8}$$

3.2.3 Tyre Model

Vehicle dynamics are significantly impacted by the forces and moments from the tyre-road contact, and as a result, the handling limits of a vehicle is highly related to the physical limitations of the available tyre force. Hence, an appropriate tyre force model is crucial for the operation at the limits of handling.

Assuming that the tyre does not exceed the adhesion limit in the longitudinal direction, which can be enforced through an appropriate constraint in the controller development, the wheel rotational dynamics can be neglected. In this case, the longitudinal tyre force is proportional to the applied driving or braking torque on the wheel. As introduced in the previous section, the front wheels are driven by a single motor through the differential, it is assumed that the torque on the front axle is evenly distributed on the two front wheels. By introducing the three control inputs T_F, T_{RL} and T_{RR}, the longitudinal tyre force on

each wheel can be calculated as follows:

$$f_{Fjx} = \frac{T_F/2}{R_w} \tag{3.9}$$

$$f_{Rjx} = \frac{T_{Rj}}{R_w} \tag{3.10}$$

The lateral tyre force is related to the slip angle. The slip angle of a tyre is defined as the angle between the orientation of the tyre and the orientation of the velocity vector of the wheel [83]. The side slip angles on the front and rear tyres can be calculated by the following equations. It is assumed that the side slip angles are the same at left and right tyres.

$$\alpha_F = \arctan\frac{V_y + l_F \cdot r}{V_x} - \delta_F \tag{3.11}$$

$$\alpha_R = \arctan\frac{V_y - l_R \cdot r}{V_x} - \delta_R \tag{3.12}$$

And the lateral tyre force on individual wheels can be calculated by the simplified Pacejka's Magic Formula tyre model [84]

$$f_{ijy} = -f_{ijz} \cdot D\sin(C\arctan(B\alpha_i)) \tag{3.13}$$

3.2.4 Friction Circle

Assuming that the tyre-road friction coefficient μ is independent of the vertical tyre load, the total available tyre force can be calculated as:

$$f_{ij,max} = \mu f_{ijz}. \tag{3.14}$$

The tyre force on each wheel consists of two vector components in the longitudinal and lateral direction:

$$f_{ij} = \sqrt{{f_{ijx}}^2 + {f_{ijy}}^2}. \tag{3.15}$$

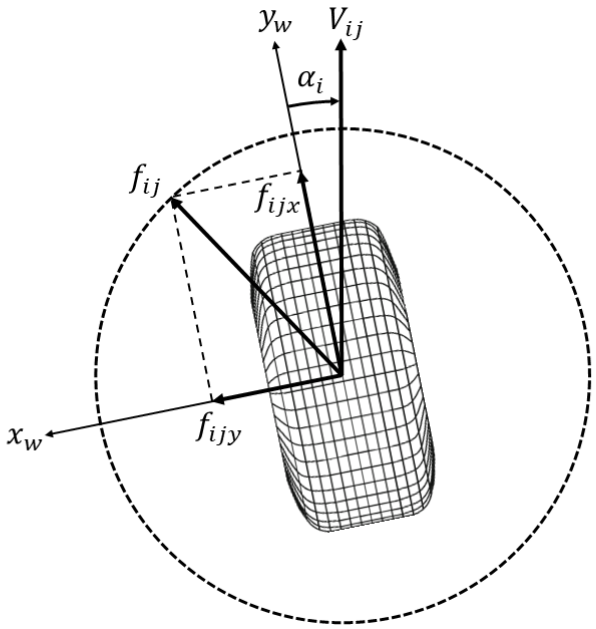


Figure 3.6: Friction circle.

Figure 3.6 provides a geometric view of Equation 3.15. Given a particular friction coefficient and vertical tyre load, the total available tyre force remains constant. Thus, for a specific longitudinal tyre force value, the maximum lateral tyre force is specific so that the total tyre force doesn't exceed its limitation.

$$f_{ijy,max} = \sqrt{{f_{ij,max}}^2 - {f_{ijx}}^2}. \tag{3.16}$$

The limitation is often called friction circle. The friction circle indicates the connection between longitudinal and lateral tyre force, and thus is an important concept to take in account for tyre modelling, especially for the limits of handling. Based on this, Equation 3.13 can be reformulated as:

$$f_{ijy} = -f_{ijy,max} \cdot \sin(C \arctan(B\alpha_i)) \tag{3.17}$$

3.3 Steady-State Reference Generation

In normal practice of MPC control design, when constructing the cost function, penalties are applied directly on the control inputs in order to minimise control efforts from a perspective of energy saving. However, minimal steering action is not expected in this research, as the vehicle is supposed to be operating at the limits of handling condition. Hence, in addition to the target vehicle states including V_x, V_y, r as well as the position information, references of the steering inputs δ_F and δ_R are also provided to the controller.

A bicycle model is used for the reference generation of the steering inputs. Assuming that the turning radius is much larger than the wheelbase of the vehicle, small angles can be assumed, and thus the difference between inner and outer wheels can be negligible [85]. A bicycle model can be obtained by representing left and right wheels with a single one, with a cornering force equivalent to both wheels.

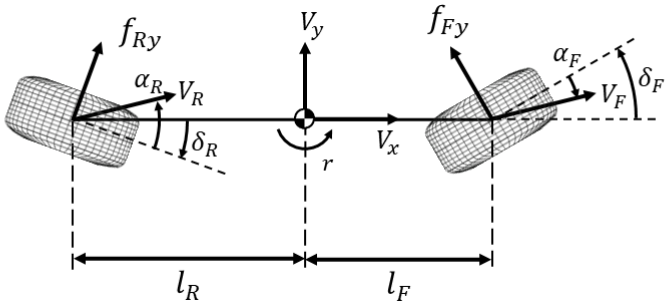


Figure 3.7: Bicycle model.

Figure 3.7 shows a diagram of the bicycle model for reference generation. With the focus on lateral dynamics only, the EOM are given in the state space form by:

$$\begin{bmatrix} \dot{V}_y \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -\frac{(C_F+C_R)}{mV_x} & -V_x - \frac{(l_F C_F - l_R C_R)}{mV_x} \\ -\frac{(l_F C_F - l_R C_R)}{I_z V_x} & -\frac{(l_F{}^2 C_F + l_R{}^2 C_R)}{I_z V_x} \end{bmatrix} \cdot \begin{bmatrix} V_y \\ r \end{bmatrix} + \begin{bmatrix} \frac{C_F}{m} & \frac{C_R}{m} \\ \frac{l_F C_F}{I_z} & \frac{-l_R C_R}{I_z} \end{bmatrix} \cdot \begin{bmatrix} \delta_F \\ \delta_R \end{bmatrix}, \tag{3.18}$$

where C_F and C_R are the cornering stiffness of the front and rear tyres respectively. By assuming a steady-state condition where $\dot{V}_y$ and $\dot{r}$ are zero, Equation 3.18 can be transformed as the following:

$$\begin{bmatrix} \delta_{F,SS} \\ \delta_{R,SS} \end{bmatrix} = - \begin{bmatrix} \frac{C_F}{m} & \frac{C_R}{m} \\ \frac{l_F C_F}{I_z} & \frac{-l_R C_R}{I_z} \end{bmatrix}^{-1} \cdot \begin{bmatrix} -\frac{(C_F+C_R)}{mV_{x,SS}} & -V_{x,SS} - \frac{(l_F C_F - l_R C_R)}{mV_{x,SS}} \\ -\frac{(l_F C_F - l_R C_R)}{I_z V_{x,SS}} & -\frac{(l_F{}^2 C_F + l_R{}^2 C_R)}{I_z V_{x,SS}} \end{bmatrix} \cdot \begin{bmatrix} V_{y,SS} \\ r_{SS} \end{bmatrix}, \tag{3.19}$$

where $V_{x,SS}$, $V_{y,SS}$ and r_{SS} are the steady states of the vehicle according to the references, while $\delta_{F,SS}$ and $\delta_{R,SS}$ are the corresponding steering inputs. With Equation 3.19, references of the steering inputs can be calculated given the reference velocity, side slip angle and yaw rate. Providing such reference inputs to the controller helps to normalise the variation of steering actions during steady-state condition, and thus aids the stabilisation of the vehicle.

Chapter 4

MPC Control Formulations

4.1 Introduction

In this chapter, controllers are developed for the purpose of path tracking at the limits of handling by applying MPC.

MPC doesn't represent a specific control strategy, instead it refers to a wide variety of control methods that make explicit use of a system model for control development [34]. The basic idea of MPC, is to take advantage of a dynamic model to predict system states, and obtain the optimal control action by minimising an objective function. Figure 4.1 shows a scheme of discrete MPC. At current time k, the system state is measured. The future system state within a prediction horizon is then predicted based on the system dynamic model, the measurement state and predicted control input. In this procedure, different control inputs are attempted to find out the one leading to the most ideal output prediction. Once the series of control actions are determined for this time step, only the first control input will be applied to the system. After that, another measurement will be taken on the system states and the procedure will be repeated at next time $k+1$.

Different with the optimal control methods such as LQR, which normally solve a QP

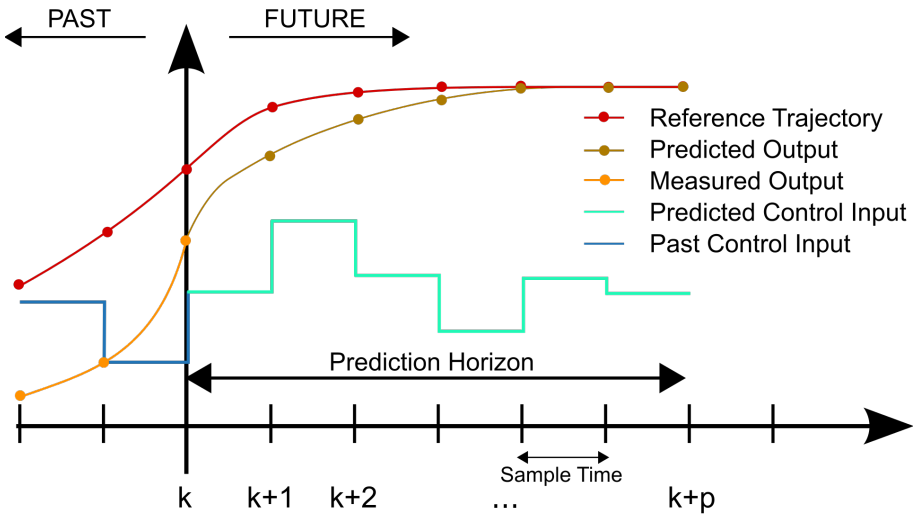


Figure 4.1: Scheme of discrete MPC principle.

problem in an infinite time domain, MPC carries out optimisation within a finite time domain. The prediction horizon of MPC rolls ahead along with MPC works in so called a receding horizon, and this is why MPC is also called receding horizon control. Working in the receding horizon allows MPC to effectively respond to the change of system condition or to external disturbance, which improves the system robustness.

Another great advantage of MPC compared with classic control and optimal control methods like PID, LQR, sliding mode control and so on, is that it is able to take into account the constraints, even the complicated ones can be dealt with properly. This helps to include the limitation of actuators and vehicle to ensure that the control commands are feasible and suitable.

According to the principle of MPC shown above, it is obvious that the dynamic model plays an important role in the MPC control. It determines the precision of the prediction of system behaviour within the horizon, and thus have a great impact on the final control performance. However, a high fidelity model often means high complexity, which could lead to potential long computation time. In order to realise real-time control, it is critical to find a balance between control performance and computation time.

The vehicle and tyre model discussed in Chapter 4 is used as the internal model for MPC

controller design. Two MPC controllers are developed and discussed in this chapter and the difference lays in the fact that they apply different road models in the formulation for path tracking. One controller uses Cartesian Coordinate system, while the other controller applies so called ”curvilinear coordinate system” which cares more about the relative position of the vehicle with regards to the reference path. The control performance of the two controllers are compared to see which is more suitable for the dynamic model used and for the specific path tracking purpose.

4.2 Cartesian Coordinate Formulation

For the path tracking purpose, it is fundamental to include the vehicle position in the model. In this section, global coordinates are used to identify the vehicle position. This is a straightforward formulation as the required position information can be easily obtained from localisation technology like GPS sensors. The derivatives of the vehicle position as well as yaw angle Ψ can be calculated as:

$$\dot{X} = V_x \cos(\Psi) - V_y \sin(\Psi) \tag{4.1}$$

$$\dot{Y} = V_x \sin(\Psi) + V_y \cos(\Psi) \tag{4.2}$$

$$\dot{\Psi} = r \tag{4.3}$$

The nonlinear continuous-time system can be described as:

$$\dot{x_t} = fc(x_t,\ u_t), \tag{4.4}$$

where x_t stands for the state vector $[V_x, V_y, r, X, Y, \Psi]^T$ and u refers to the control input

vector $[\delta_F, \delta_R, T_F, T_{RL}, T_{RR}]^T$.

The controller development is based on NMPC, and is realised in the sampled-data framework by discretizing the nonlinear continuous-time system with explicit Runge-Kutta 4th order method. The main purpose of the controller is to follow the reference path at the reference velocity, and the discrete NMPC problem is formulated as

$$\begin{aligned}
&\min_{x,u} \sum_{k=0}^{N-1} (x_{k+1} - x_{ref,k+1})^T Q (x_{k+1} - x_{ref,k+1}) + (u_k - u_{ref,k})^T R (u_k - u_{ref,k}) \\
&st.\ \ x_0 = x_{initial} \\
&\quad x_{k+1} = f_d(x_k, u_k), k = 0, \cdots, N-1 \\
&\quad x_{min,k} \leq x_k \leq x_{max,k}, k = 0, \cdots, N-1 \\
&\quad u_{min,k} \leq u_k \leq u_{max,k}, k = 0, \cdots, N-1
\end{aligned} \tag{4.5}$$

where N is the prediction horizon steps, and Q, R are the weighting matrices of the state and control input vectors respectively. f_d represents the discrete-time system derived from f_c.

The reference states are calculated with the bicycle model introduced in Chapter 3, according to the actual vehicle status. The reference path is parametrised by the arc length S along the path from the origin point, where $S \in [0, L]$ and L is the total length of the path. With this parameterisation, the position $X_{ref}(S)$, $Y_{ref}(S)$ of any point on the reference path can be obtained by carrying out spline interpolation according to the argument S. In addition, the tangential angle of the path at the point can be obtained as

$$\Psi_{ref}(S) = \arctan \frac{\partial Y_{ref}(S)}{\partial X_{ref}(S)}, \tag{4.6}$$

and it is used as the reference yaw angle of the vehicle. This parameterisation takes advantage of the known waypoints on the reference path and provides an accurate interpolation within them [59].

For path tracking purpose, the relative position of the vehicle with regards to the reference path is required. Point $X_{ref}(S_0), Y_{ref}(S_0))$ is proposed as the projection of the vehicle position on the reference path, and S_0 can be obtained by solving the optimisation problem

$$S_0 = \min_S \ [X - X_{ref}(S)]^2 + [Y - Y_{ref}(S)]^2. \tag{4.7}$$

S_0 can be used to denote the progress of the vehicle along the reference path, and the distance between the vehicle and this projection point refers to the lateral deviation of the vehicle from the path.

For the discrete objective function, a total of N waypoints are required to generate Z_{ref}. The waypoints are supposed to follow the projection point $(X_{ref}(S_0), Y_{ref}(S_0))$, with an interval of ΔS,

$$\Delta S = V_{ref} \cdot t_s, \tag{4.8}$$

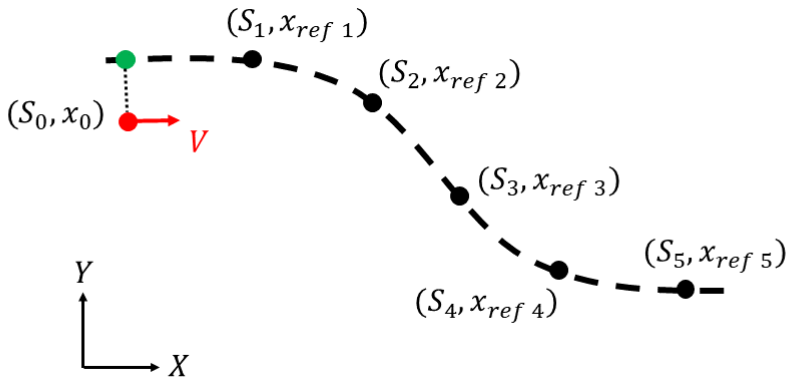


Figure 4.2: Diagram of the reference waypoints. The red point represents the vehicle's position, and the green point stands for the projection of vehicle position on the reference path. The interval of the waypoints $\Delta S = V_{ref} t_s$, where V_{ref} is the reference velocity and t_s is the sampling time of the controller.

where V_{ref} is the reference velocity and t_s is the sampling time of the discrete-time system. Figure 4.2 shows a diagram of the generated waypoints based on the vehicle's position, the projection waypoint and the reference velocity. The reference Cartesian coordinates and yaw angle are then evaluated by carrying out spline interpolation based on the argument S. In order to reduce the computation time, the solving of this optimisation problem is carried

out in a ”local” range, which means only the points within a specific range are included in the problem, and the range is supposed to be within a short distance proportional to velocity to the vehicle's last position. This could massively increase the efficiency of localisation while maintaining accurate solution under assumption that the vehicle does not deviated far from the reference path.

4.3 Curvilinear Coordinate Formulation

Another way to identify the vehicle position, is to translate it as the relative position with regards to the reference path, which is exactly what a curvilinear coordinate system does. Figure 4.3 shows a diagram of the curvilinear coordinate system used in this formulation. Similar to the Cartesian coordinate system introduced in the last section, this also requires three degrees of freedom to describe the vehicle position. It interprets the vehicle position into S, ε_y and ε_Ψ, where S refers to the distance that the vehicle has travelled along the reference path, ε_y indicates the lateral deviation of the vehicle from the reference path, which can also be identified as the tracking error, and ε_Ψ is the difference between the vehicle's yaw angle and the tangent of the reference path at projected point of the vehicle's position on the path.

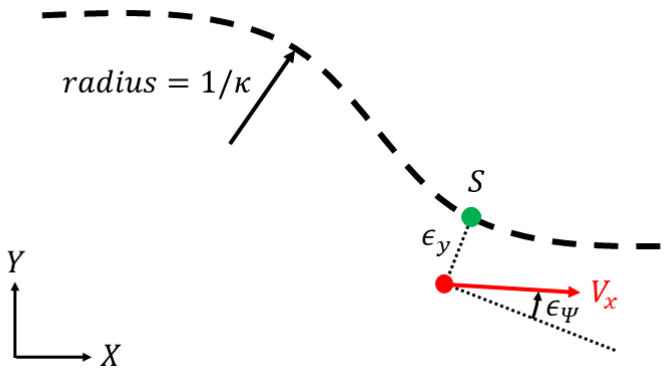


Figure 4.3: Diagram of the curvilinear coordinate system used in the control system.

The derivatives of the curvilinear coordinates can be calculated with Equation 4.9:

$$\begin{aligned} \dot{S} =& \frac{V_x \cos(\varepsilon_\Psi) - V_y \sin(\varepsilon_\Psi)}{1 - \kappa \varepsilon_y} \\ \dot{\varepsilon}_y =& V_x \sin(\varepsilon_\Psi) + V_y \cos(\varepsilon_\Psi) \\ \dot{\varepsilon}_\Psi =& r - \kappa \dot{S} \end{aligned} \tag{4.9}$$

The nonlinear continuous-time system can be described again by Equation 4.4, with the state vector x_t replaced by $[V_x, V_y, r, S, \varepsilon_y, \varepsilon_\Psi]^T$, while the control input vector u remains the same as $[\delta_F, \delta_R, T_F, T_{RL}, T_{RR}]^T$. The discrete NMPC problem can be constructed in the same formulation as given by Equation 4.5.

Figure 4.4 shows the diagram of the complete control algorithm. The parameterisation of the desired path is completed offline prior to the simulation. Since there is a QP problem within the NMPC formulation to be solved, a solver is required. Due to the target of applying the controller in real-time operation, it is necessary to ensure that the solver is able to solve the QP problem in short time as well as to provide the correct optimal solution. In this project, the NMPC solver is built with a tool called Forces Pro [86, 87], which can generate fast executable optimisation solvers based on highly customised optimisation problem.

4.4 Simulation Results

This section demonstrates the simulation results of the vehicle's path tracking performance with different control formulation and configurations. The main purpose of the controller is to track a reference path at a reference velocity. First the controllers with Cartisian coordinate formulation (CTSNF) and curvilinear coordinate formulation (CVLRF) are compared in two scenarios, including a double lane change and a double U-turn. Then the control performance is further investigated to evaluate the effect of steady-state refer-

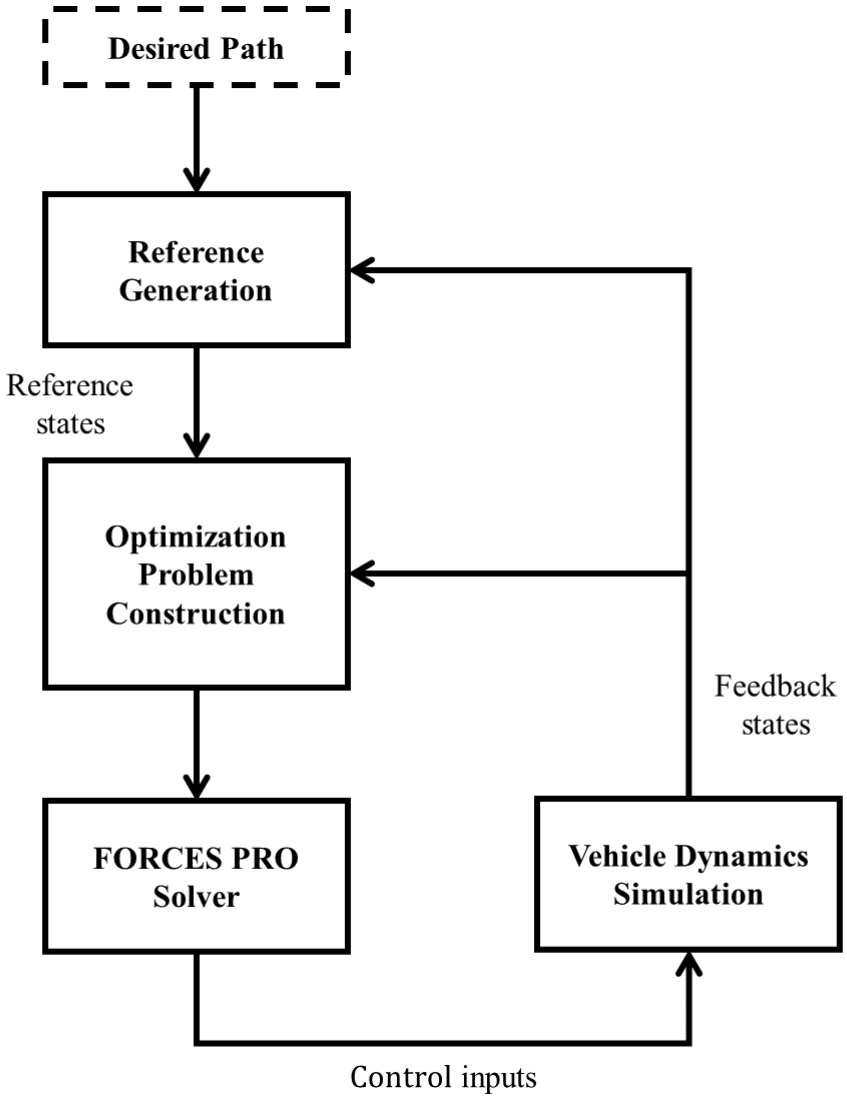


Figure 4.4: Diagram of the control system.

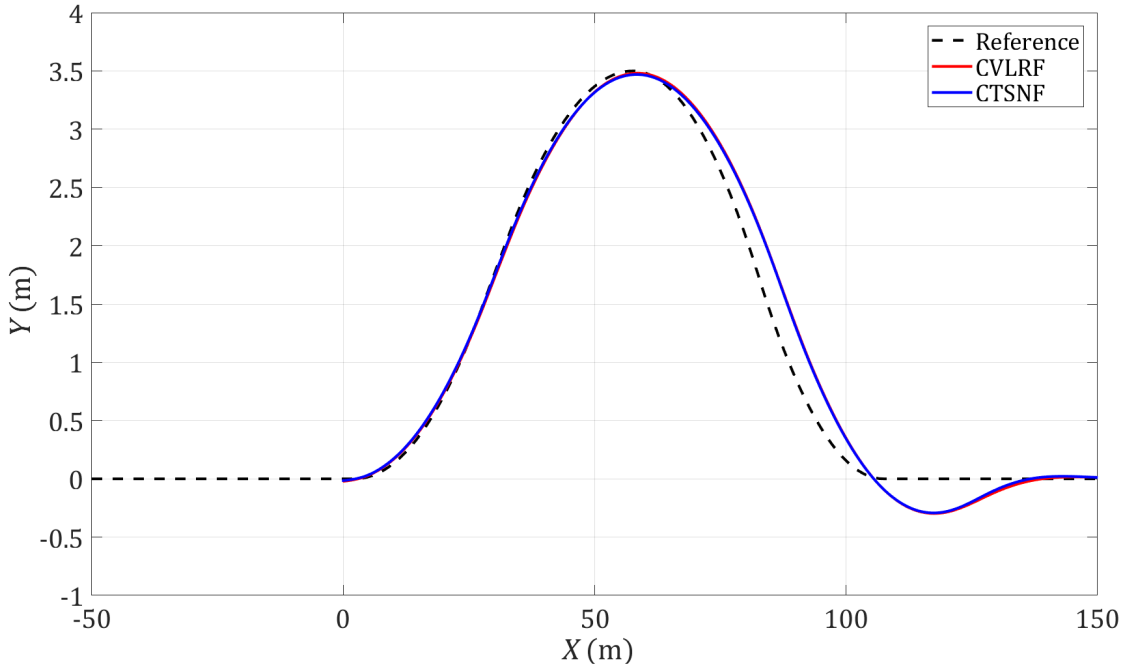
ence (SSR). The simulation is carried out within IPG CarMaker, which is a software that provides high fidelity vehicle dynamics simulation. The simulation sessions are run on a workstation laptop, and its specification can be found in Appendix D. In the simulation, a good tyre-road friction is assumed ($\mu = 1.16$), while the parameters of the vehicle are listed in Appendix A.

4.4.1 Comparison of Different Path Tracking Formulation

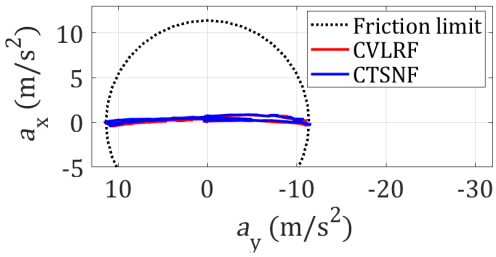
Double Lane Change Scenario

The two controllers with different path tracking formulations are first evaluated in a double lane change manoeuvre. This is a common manoeuvre in daily driving for overtaking and obstacle avoidance, and thus is widely used in automotive industry for testing vehicle dynamics and handling response. A reference path is designed according to the dimensions of the double lane-change track defined in ISO 3888-1-2018, the detailed layout of which can be found in Appendix B.2. The vehicle is expected to track a constant reference velocity throughout the double lane change manoeuvre. The purpose of the simulation is to find out the maximum entry speed allowed by both controllers to carry out a qualified double lane change. Here a qualified double lane change is defined as tracking the reference path within an lateral tracking error of 0.3m at the exit of the double lane change. Given the track width as well as the vehicle dimension, the 0.3m limitation would guarantee that the vehicle won't hit the boundary of the track.

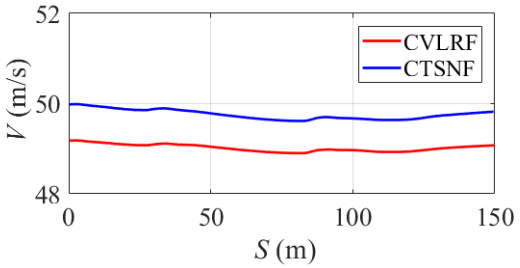
Figure 4.5 shows the states of the vehicle during the double lane change manoeuvre under both control formulations. According to the simulation, the CVLRF controller was able to maintain the vehicle within the 0.3m lateral error at 49.2 m/s, while the CTSNF controller manages to maintain the lateral error within 0.3m at the velocity up to 50.0 m/s, which is shown in Figure 4.5c. Figure 4.5a shows the trajectory of the vehicle and in Figure 4.5b, it can be seen that the vehicle has reached the acceleration limit, indicating the limits of



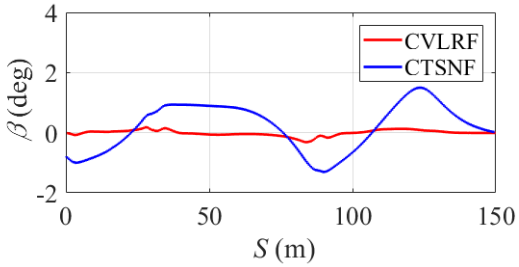
(a) Vehicle trajectory



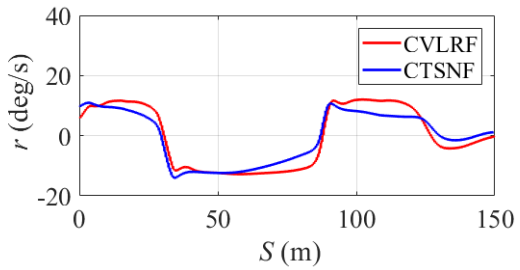
(b) g-g diagram



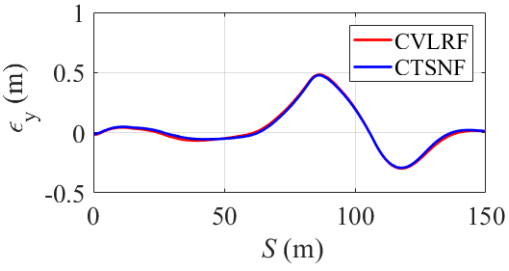
(c) Velocity



(d) Side slip angle

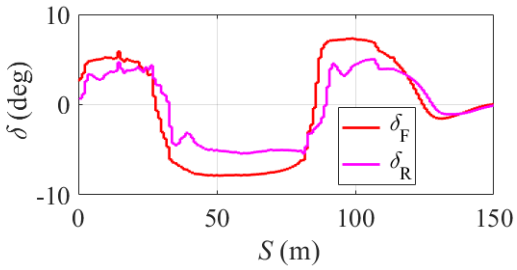


(e) Yaw rate

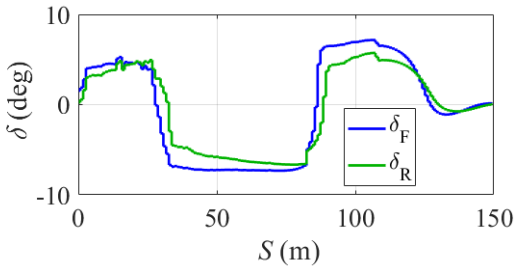


(f) Lateral tracking error

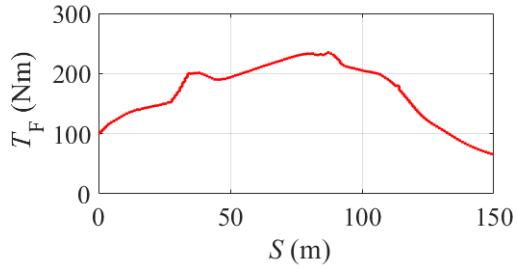
Figure 4.5: Vehicle states in the double lane change scenario with CVLRF and CTSNF controllers.



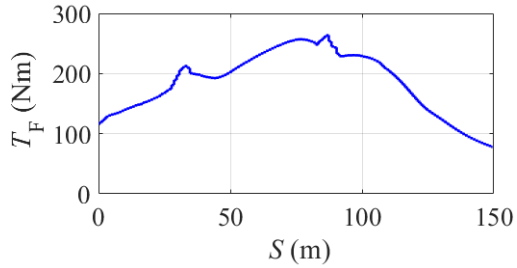
(a) Steering commands



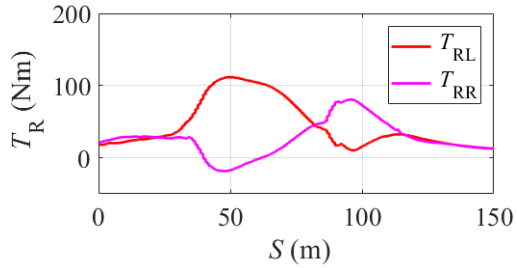
(b) Steering commands



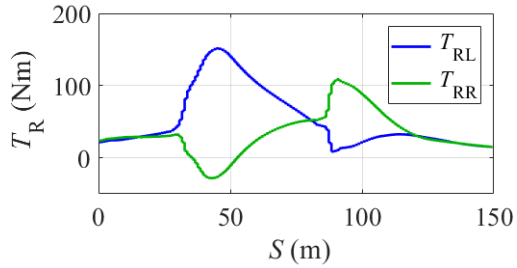
(c) Front wheel torque commands



(d) Front wheel torque commands



(e) Rear wheel torque commands



(f) Rear wheel torque commands

Figure 4.6: Control commands in the double lane change scenario with CVLRF and CT-SNF controllers.

Table 4.1: Summary of path-tracking performance of CVLRF and CTSNF controllers in the double lane change scenario.

	CVLRF	CTSNF
Max V (m/s)	49.2	50.0
RMS ε_y (m)	0.162	0.182
Max $\lvert\varepsilon_y\rvert$ (m)	0.483	0.478
Average t_{solve} (s)	0.033	0.029
Max t_{solve} (s)	0.055	0.060

handling.

Table 4.1 shows a summary of some key performance indicators of path-tracking performance with CVLRF and CTSNF controllers in the double lane change scenario. It can be seen that the CTSNF controller is able to reach higher velocity and maintain smaller lateral tracking error while taking lower computation time, but the advantage is not significant. On the other hand, the maximum computation time of the CTSNF controller is actually higher than the CVLRF controller, suggesting that optimisation on the formulation is required on the CTSNF controller for more consistent performance.

Double U-turn Scenario

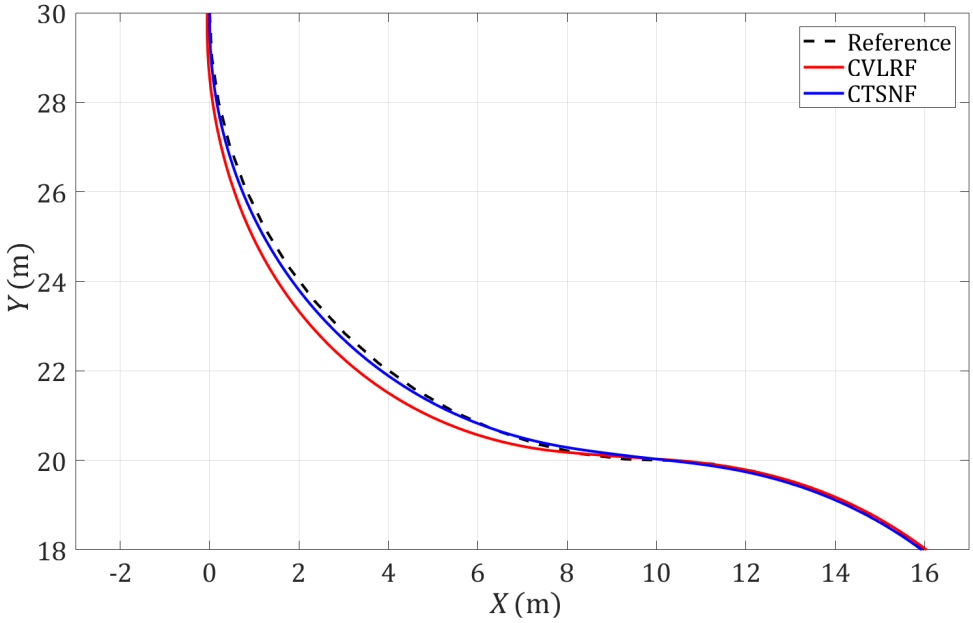
The performance of the two formulations are tested in a double U-turn scenario as well. The reference path has two semicircle turns with the same radius of 10m, and two straights on the beginning and end of the double U-turn. A diagram of the reference path is shown in the Appendix B.1. Different from the double lane change scenario, the double U-turn track has in-continuous curvature profile, which could be a challenge for the controllers to track the path with minimum error. A reference velocity of 10m/s is provided to both controllers for path tracking, and their performance is then compared. A particular focus is put on the vehicle stability as well as tracking performance at the joining point of the two semicircle turning course, where the maximal change of curvature takes place.

In the simulation, the vehicle reaches its handling limits again with both controllers, as demonstrated by Figure 4.7b. It can be noticed that the vehicle's acceleration is deviated more from the lateral direction with the CVLRF controller. Figure 4.7c shows the velocity of the vehicle, and it can be seen that a drop of velocity happens with the CVLRF controller in midway of the path, where the harsh change of direction happens. In contrast, the CTSNF controller is able to track the reference velocity with a maximal deviation of 0.2 m/s, which is much better than the 0.7 m/s deviation with the CVLRF controller. As shown in Figure 4.7d, the CVLRF controller achieves smaller sideslip angle than the CTSNF, while in Figure 4.7e, the yaw rate of the vehicle is smoother with the CTSNF controller.

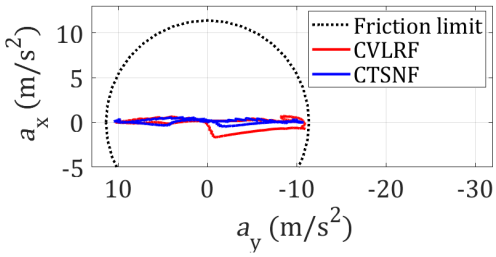
In terms of the path tracking performance, Figure 4.7f shows the lateral tracking error of the vehicle with both controllers. The CTSNF controller is able to maintain the lateral tracking error within 0.12 m over the entire path, while a larger error up to 0.43 m happens with the CVLRF controller. Together with Figure 4.7a showing the vehicle trajectory, it can be told that the drawback of path tracking performance with the CVLRF controller happens right after the change of direction in the midway of the double U-turn path. The CVLRF doesn't respond promptly enough to the direction change, leading to a larger deviation from the reference path.

Figure 4.8 shows the control commands from both controllers during the double U-turn scenario. From Figure 4.8a and Figure 4.8b it can be seen that opposite front and rear steering angles are commanded by both controllers during the turning. The system

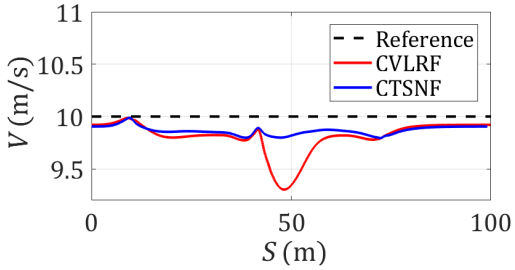
Table 4.2 shows a summary of the path-tracking performance of both CVLRF and CTSNF controllers in the double U-turn scenario. Here ε_V refers to the tracking error of velocity. In the double U-turn scenario, the CTSNF controller has shown its advantage over the CVLRF controller, with obviously smaller tracking error of both reference velocity and path. The average computation time of the CTSNF controller is also lower than the CVLRF controller, while the maximum value is still higher.



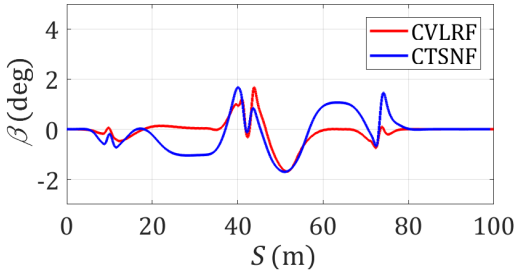
(a) Vehicle trajectory



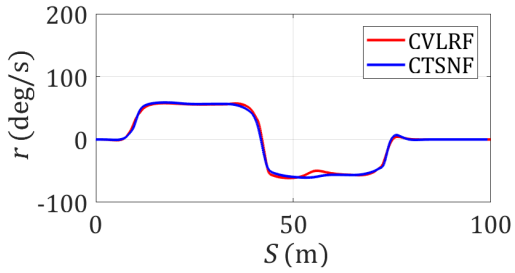
(b) g-g diagram



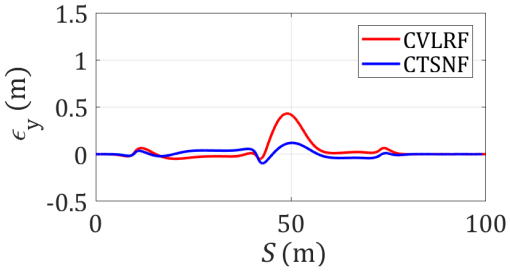
(c) Velocity



(d) Side slip angle

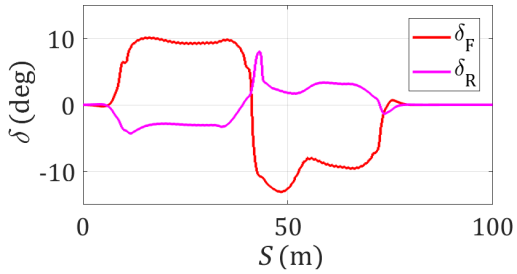


(e) Yaw rate

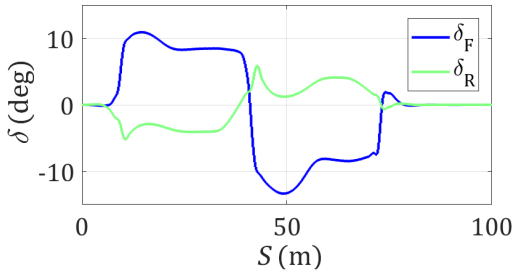


(f) Lateral tracking error

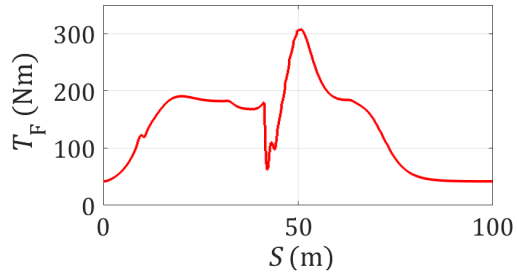
Figure 4.7: Vehicle states in the double U-turn scenario.



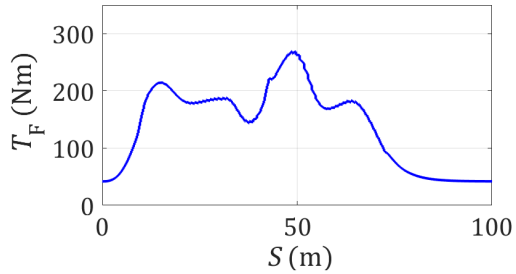
(a) Steering commands of CVLRF controller.



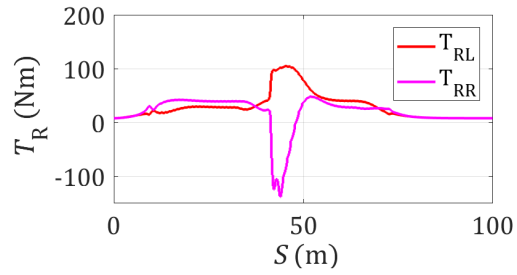
(b) Steering commands of CTSNF controller.



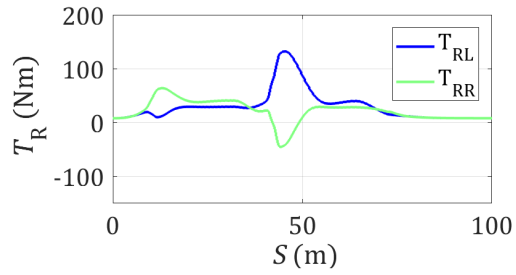
(c) Front wheel torque commands of CVLRF controller.



(d) Front wheel torque commands of CTSNF controller.



(e) Rear wheel torque commands of CVLRF controller.



(f) Rear wheel torque commands of CVLRF controller

Figure 4.8: Control commands in the double U-turn scenario.

Table 4.2: Summary of path-tracking performance of CVLRF and CTSNF controller in the double U-turn scenario.

	CVLRF	CTSNF
RMS ε_V (m/s)	0.251	0.139
Max $\lvert\varepsilon_V\rvert$ (m/s)	0.698	0.206
RMS ε_y (m)	0.117	0.040
Max $\lvert\varepsilon_y\rvert$ (m)	0.432	0.120
Average t_{solve} (s)	0.033	0.029
Max t_{solve} (m)	0.056	0.060

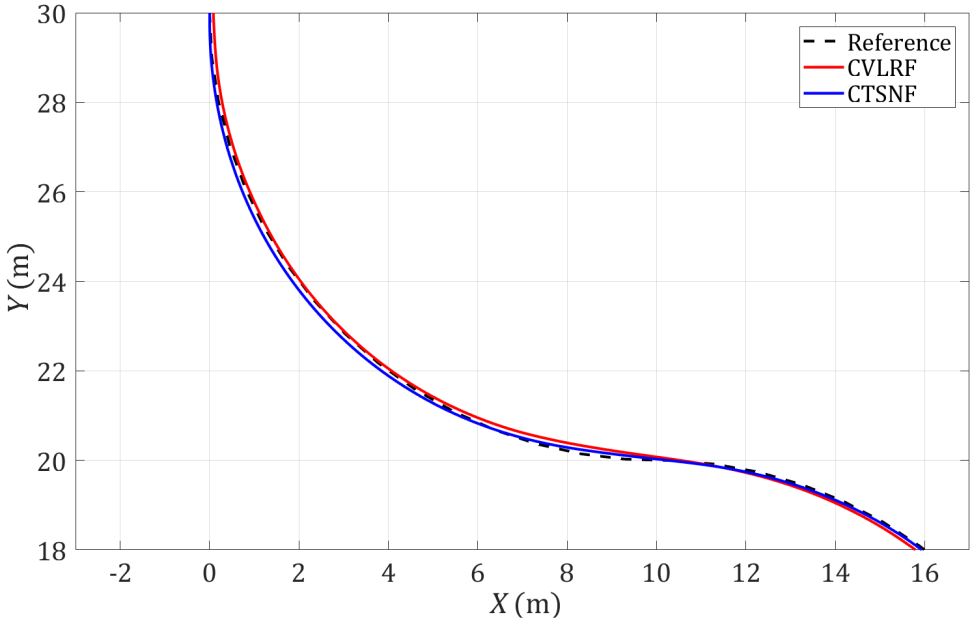
4.4.2 Evaluation of the Steady-State Reference

The effect of the steady-state reference on the path tracking performance is also evaluated. The steady-state reference is removed from the MPC formulation, and the performance is compared with the CTSNF controller in the double U-turn scenario.

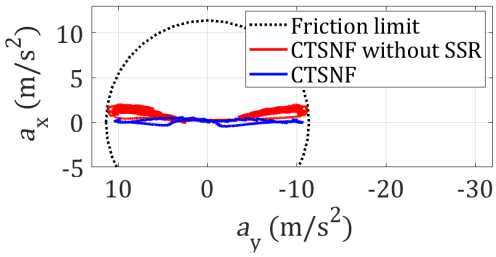
It can be told from Figure 4.9a and Figure 4.9f that without the SSR, The control performance of the CTSNF controller is worse. More oscillations can be observed during the entire tracking course. The velocity of the vehicle is shown in Figure 4.9c, and it can be seen that the CTSNF controller without SSR fails to track the reference velocity as good as the formulation including SSR. Similar circumstances are reflected in Figure 4.9d and Figure 4.9e, where oscillation takes place in the side slip angle as well as yaw rate of the vehicle.

The drawback in control performance without SSR is also reflected in the control commands shown in Figure 4.11. The rear wheel steering angle fluctuates significantly during turning without SSR, while with SSR the steering commands are much smoother. With regards to the torque commands, oscillation can also be noticed without SSR.

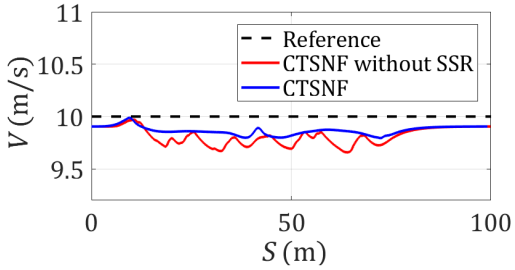
Figure 4.11 shows the computation information of the Forces Pro solvers during simulation. It can be seen in Figure 4.11a that exitflag other than 1 takes place without the



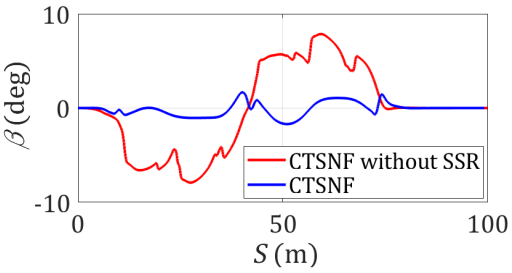
(a) Vehicle trajectory



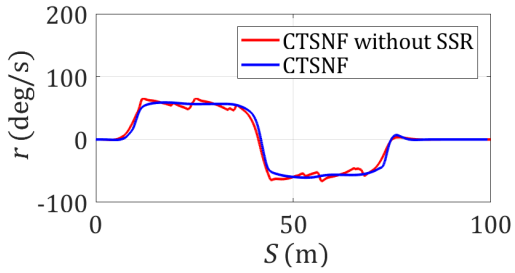
(b) g-g diagram



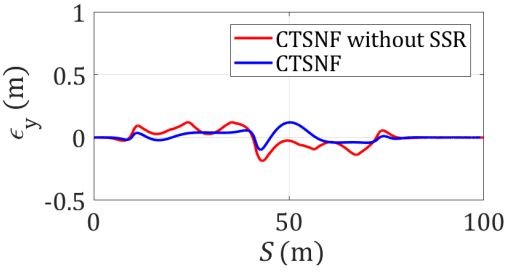
(c) Velocity



(d) Side slip angle

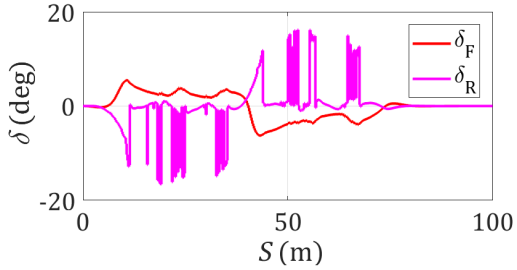


(e) Yaw rate

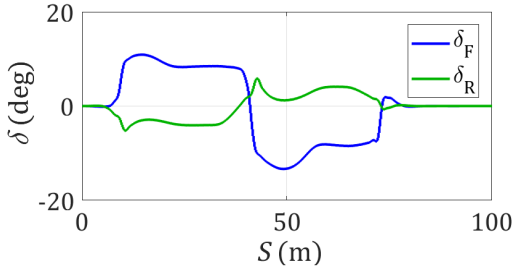


(f) Lateral tracking error

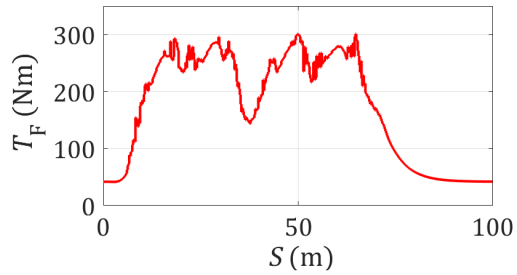
Figure 4.9: Vehicle States in U-turn



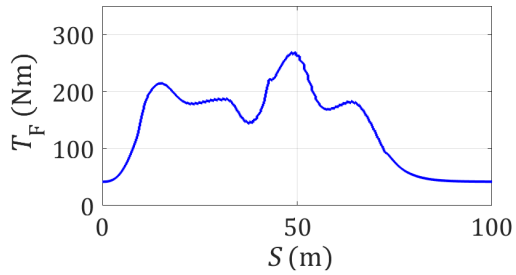
(a) Steering commands of CTSNF controller without SSR.



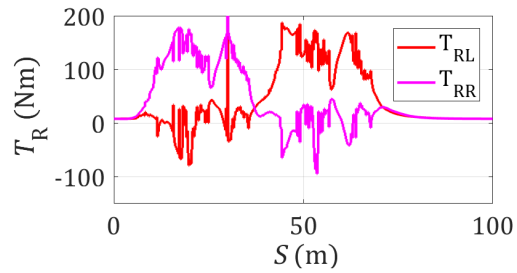
(b) Steering commands of CTSNF controller.



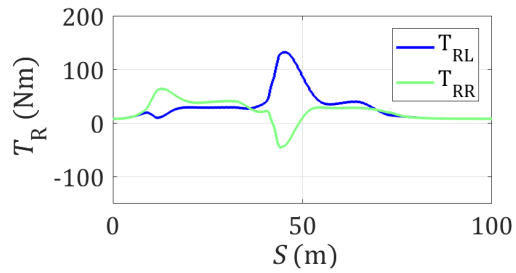
(c) Front wheel torque commands of CTSNF controller without SSR.



(d) Front wheel torque commands of CTSNF controller.

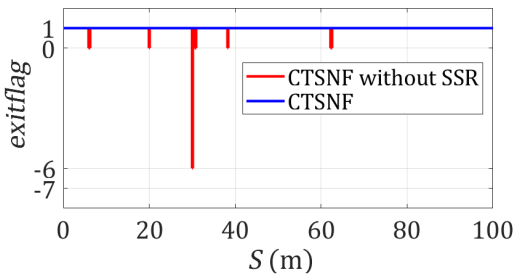


(e) Rear wheel torque commands of CTSNF controller withoutSSR.

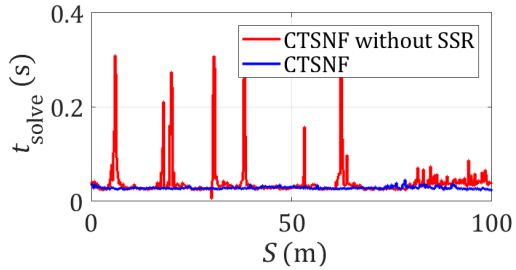


(f) Rear wheel torque commands of CTSNF controller without SSR.

Figure 4.10: Control commands in double-U-turn.



(a) Exitflag.



(b) Solve time.

Figure 4.11: Forces Pro solver information.

introduction of SSR, indicating that the solver fails to find he optimal solution of the MPC problem. In addition, the solve time of the controller without SSR cannot maintain at low level. From time to time the solve time is dramatically high, indicating that the solver is struggling with the optimisation problem. This shows that the application of SSR in the formulation helps the solver the find out optimal solutions more quickly, and thus contributes to improving the robustness and performance of the system.

4.5 Summary

In this chapter, two controllers are proposed based on MPC and different coordinate system formulation for path tracking. The MPC strategies are based on the 3DOF two-track vehicle model introduced in the previous chapter. The cost functions for both controllers are constructed aiming for minimising the path tracking error, while taking as reference the steady-state condition obtained by a bicycle model.

The controllers are tested in the simulation. Two different scenarios are used to validate the control performance, and both controllers have shown their capability in path tracking control at the limits of handling. In the double lane change scenario, the CTSNF controller is able to achieve higher speed up to 50 m/s while maintaining the lateral tracking error within the safety margin of 0.3 m. In comparison, the CVLRF controller reaches 49.2 m/s to guarantee the similar tracking error.

In the double U-turn scenario, the CTSNF provides better performance than the CVLRF controller upon the harsh change of direction in midway of the path. When the dirction change is required, the CTSNF controller manages to maintain the vehicle's velocity close to the reference, while a drop of velocity can be noticed with CVLRF controller. At the same time, the CTSNF controller also does a better job than the CVLRF controller in minimising the lateral tracking error. In addition, there is less oscillation in the yaw rate of the vehicle with the CTSNF controller. By studying the control commands from the

controllers, both controllers command opposite front and rear steering during turning. TV is applied by both controllers in response of the rapid change of direction, where a large yaw moment is required in order to prevent the vehicle from deviating from the reference path by a large error.

The CTSNF controller takes less time on average than the CVLRF controller during operation, which is critical for real time implementation. However, both controllers take longer time for execution than the sampling time, so further optimisation is required in order to implement them in real time.

More simulation has been carried out of the CTSNF controller to evaluate its performance without and without SSR. It is proven that the application of SSR can significantly improve the system stability as well as path tracking performance.

According to the simulation, the CTSNF controller stands out in terms of the path tracking performance. However, it cannot be concluded that the CVLRF controller is worse than the CTSNF controller. The performance of the CVLRF controller may be improved by further tuning the parameters or modification on the formulation. Based on the current configuration, the CTSNF provides a better performance, and thus is more suitable for this research. Hence, it is applied and will be further studied in the following parts of this thesis.

Chapter 5

Path Tracking Control with Different Levels of Actuation

5.1 Introduction

In this chapter, the effect of the application of multi-actuation on the performance of autonomous vehicle control is evaluated. Four controllers based on different level of actuation are proposed, including front-wheel steering only (FWS), four-wheel steering (4WS), front-wheel steering with TV (FWS-TV) and four-wheel steering with TV (4WS-TV). They are tested in the double U-turn as well as double lane change scenario, and the control performance is compared.

5.2 Formulation of Different Levels of Actuation

With regards to the actuation configuration, there are four formulations to be investigated, including FWS, FWS-TV, 4WS and 4WS-TV. For the FWS and FWS-TV formulations, the RWS is not supposed to be included in the functionality, and thus remains zero. For the

Table 5.1: System Configuration of the Controllers

Controller	Steering	Driving Torque
FWS	FWS only	Same on each wheel
FWS-TV	FWS only	TV applied on rear wheels
4WS	FWS+RWS	Same on each wheel
4WS-TV	FWS+RWS	TV applied on rear wheels

FWS and 4WS formulations, it is assumed that the driving torque delivered on each wheel equals the same, which means the torque from the three motors have the relationship:

$$\frac{T_F}{2} = T_{RL} = T_{RR} = T_w, \tag{5.1}$$

where T_w is the actual control input in the FWS and 4WS formulations. Table 5.1 shows the actuation configuration of each formulation, and the control input vector U of each of them is shown as follows,

$$\begin{aligned} u_{FWS} &= [\delta_F, T_w]^T \\ u_{4WS} &= [\delta_F, \delta_R, T_w]^T \\ u_{FWS-TV} &= [\delta_F, T_F, T_{RL}, T_{RR}]^T \\ u_{4WS-TV} &= [\delta_F, \delta_R, T_F, T_{RL}, T_{RR}]^T \end{aligned} \tag{5.2}$$

5.3 Simulation Results

In this section, the simulation results of the path tracking performance with the four controllers are demonstrated. The controllers are tested in two scenarios, double U-turn and double lane change.

5.3.1 Double Lane Change Scenario

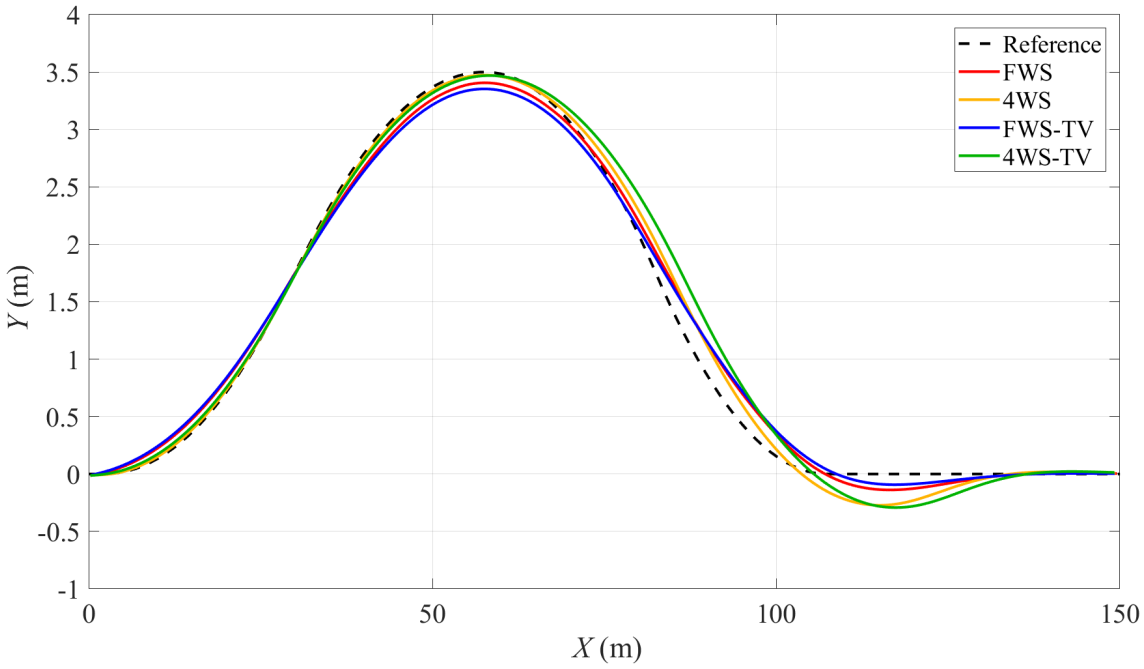
The four controllers are tested in the double lane scenario first. The reference velocity for each controller keeps increasing by 0.1m/s until the lateral deviation on exit exceeds 0.3m.

Figure 5.1c shows the highest velocity that each controller can achieve during the simulation. The FWS controller can only carry out the double lane change manoeuvre up to 44.6m/s, while the FWS-TV controller handles the same task up to 46.3m/s. The 4WS and 4WS-TV controllers are able to achieve higher velocity, with the former up to 48.1 m/s and the latter up to 50m/s. Looking into Figure 5.1d and Figure 5.1e, it can be seen that the two controllers with rear steering helps the vehicle to pass the double lane change course with smaller side slip angle as well as smaller yaw rate.

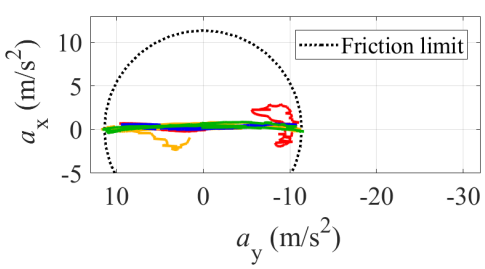
Figure 5.2 shows the commands of the four controllers during the double lane change. It can be seen that there is again quite a few oscillation in the commands of the FWS controller. In terms of the steering commands, it is noticed that both 4WS and 4WS-TV controllers requires front and rear steering in the same direction, so that less tyre force in the lateral direction is required for the lateral motion of the vehicle.

Figure 5.3b shows the computation time costed by the four controllers, which is generally similar as the results in the double U-turn scenario. The FWS controller requires generally the shortest time for computation, while it takes the most time for the 4WS-TV controller to operate. All of the four controllers take longer time than the sampling time of 0.02 s for computation, which means that none of them can be directly implemented in real time, and further simplification is required for doing so.

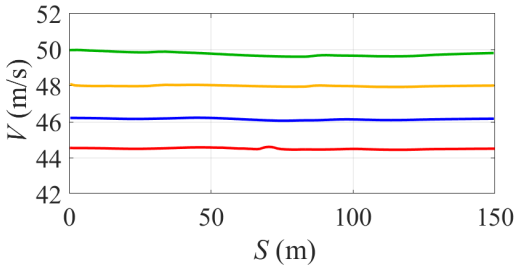
Table 5.2 shows a summary of the path-tracking performance with different levels of actuation in the double lane change scenario.



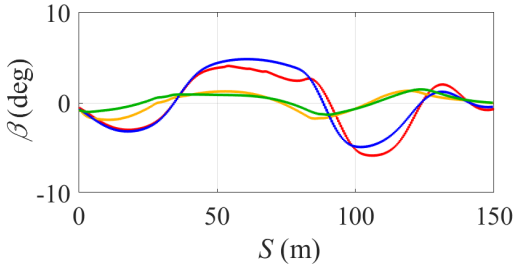
(a) Vehicle trajectory



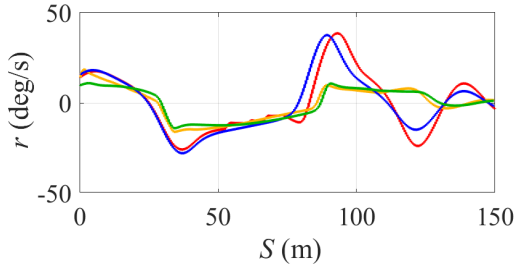
(b) g-g diagram



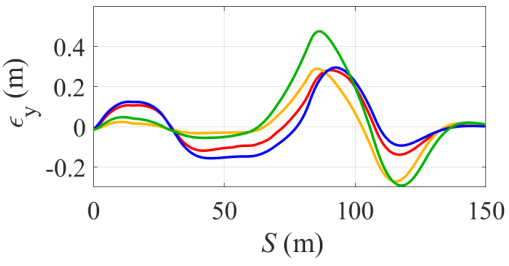
(c) Velocity



(d) Side slip angle

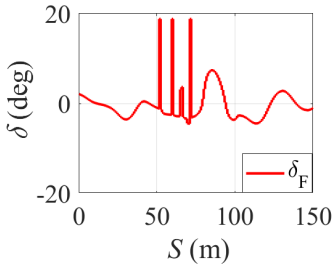


(e) Yaw rate

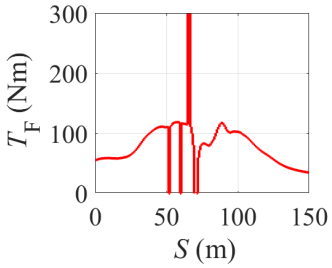


(f) Lateral tracking error

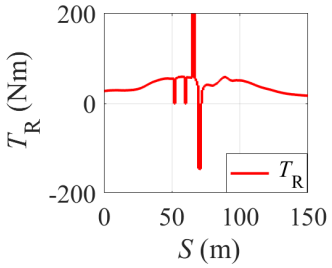
Figure 5.1: Vehicle States in double lane change.



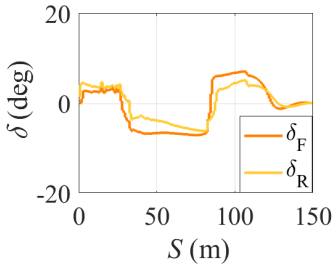
(a) Steering commands of FWS controller.



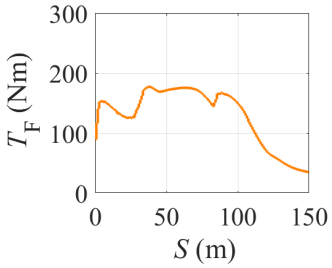
(b) Front wheel torque commands of FWS controller.



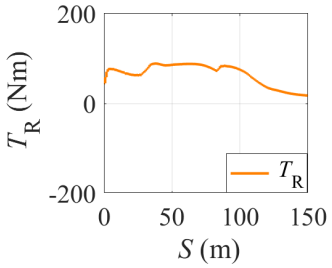
(c) Rear wheel torque commands of FWS controller.



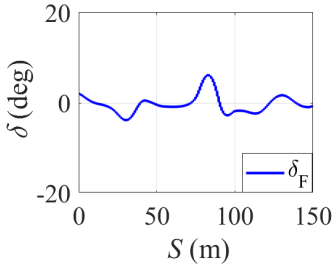
(d) Steering commands of 4WS controller.



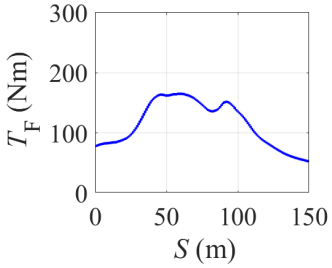
(e) Front wheel torque commands of 4WS controller.



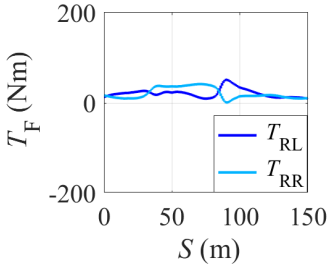
(f) Rear wheel torque commands of 4WS controller.



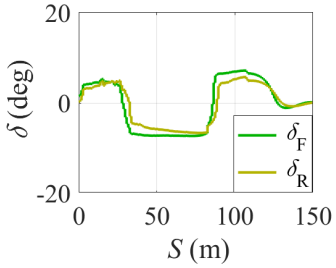
(g) Steering commands of FWS-TV controller.



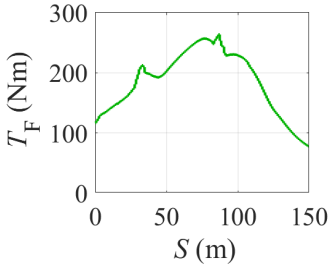
(h) Front wheel torque commands of FWS-TV controller.



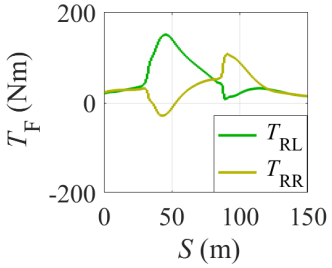
(i) Rear wheel torque commands of FWS-TV controller.



(j) Steering commands of 4WS-TV controller.

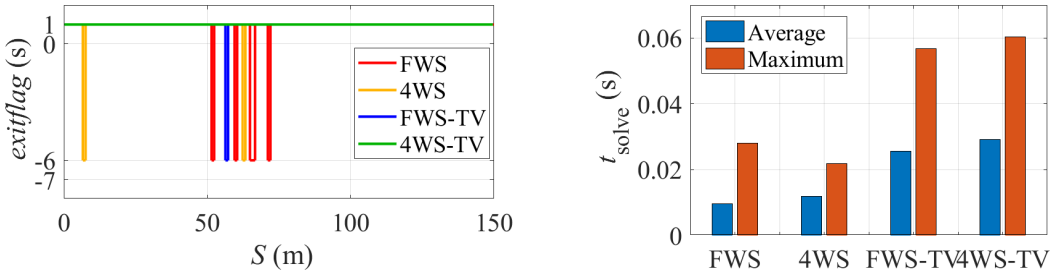


(k) Front wheel torque commands of 4WS-TV controller.



(l) Rear wheel torque commands of 4WS-TV controller.

Figure 5.2: Control commands in double lane change.



(a) Exitflag.

(b) Solve time.

Figure 5.3: Forces Pro solver information.

Table 5.2: Summary of path-tracking performance with different levels of actuation in the double lane change scenario.

	FWS	4WS	FWS-TV	4WS-TV
Max V (m/s)	44.6	48.1	46.3	50.0
RMS ε_y (m)	0.119	0.123	0.128	0.182
Max $\lvert\varepsilon_y\rvert$ (m)	0.283	0.291	0.297	0.478
Average t_{solve} (s)	0.010	0.012	0.026	0.029
Max t_{solve} (m)	0.028	0.022	0.057	0.060

5.3.2 Double U-turn Scenario

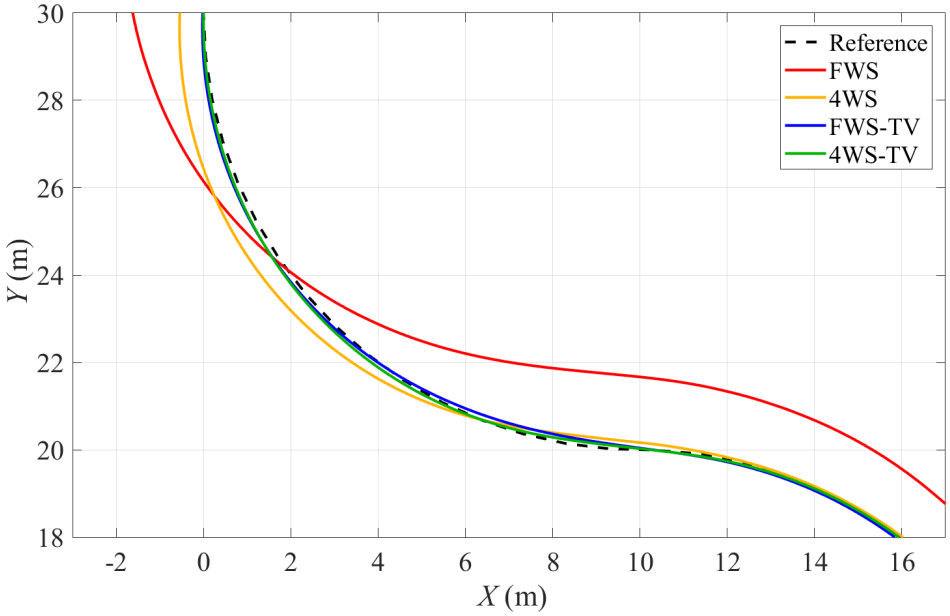
Next, the controllers are tested in the double U-turn scenario. The controllers are supposed to track the reference path at a constant reference velocity which is given as 10m/s. Figure 5.4b shows the acceleration of the vehicle. It can be seen that the vehicle is quite close to the friction limit with all the four controllers, indicating the operation at the limits of handling on the double U-turn path. However, evident oscillation in acceleration can be observed with the FWS and 4WS controllers. Figure 5.4c shows the velocity tracking performance of the vehicle. The FWS and 4WS controllers has larger variation in velocity tracking. In comparison, the FWS-TV and 4WS-TV controllers do a better job in maintaining the velocity during the sharp turn.

Figure 5.4d shows the side slip angle of the vehicle. It can be seen that the vehicle tends to have a smaller side slip angle with the 4WS and 4WS-TV controllers, while a larger

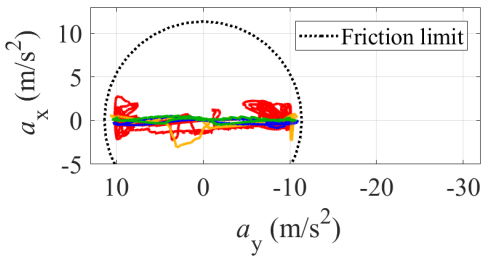
side slip angle can be noticed with the other two control formulation without rear steering. This proves that the introduction of RWS in the control formulation helps to manipulate the vehicle attitude control, as the cost function includes a reference side slip angle of 0°. Figure 5.4e shows the yaw rate of the vehicle, and a larger overshoot can be seen with the FWS controller after the double U-turns.

The vehicle trajectories are shown in Figure 5.4a. The vehicle deviates significantly from the reference path with the FWS controller, which shows that the FWS controller is not able to handle the turning at the extreme condition. The performance of the 4WS controller is acceptable, but its performance gets worse after the change of direction. In comparison, the FWS-TV and 4WS-TV controllers have achieved better path tracking performance, especially keeping close to the reference path after the sharp change direction. Figure 5.4f shows the lateral deviation ε_y of the vehicle from the reference path with all four controllers. The vehicle has got the largest lateral deviation up to 3m with the FWS controller on the double U-turns. The application of either 4WS or TV can significantly reduce the lateral tracking error, but the utilisation of TV provides a greater improvement. The 4WS controller has the lateral deviation up to 0.6m, while the FWS-TV controller maintains the tracking error within 0.15m. Among the four controllers, the 4WS-TV controller achieves the best path tracking performance with the smallest lateral tracking error. By comparing the FWS-TV and 4WS-TV controllers in particular, it can be seen that the vehicle has smaller steady-state error during turning, and the error gets back to zero quicker after the change of direction. This proves the advantage of RWS in addition to TV for autonomous vehicle path tracking, particularly when close to the limits of handling. RWS improves the vehicle's flexibility and potential to deal with emergency scenarios that require sharp turning.

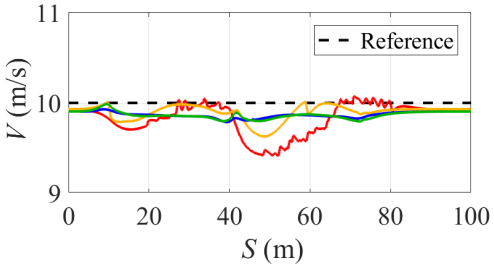
Figure 5.5 shows the control commands of the controllers. There is large fluctuation in the commands of the FWS controller, showing that it is quite struggling in the path tracking control of the vehicle. In terms of the steering commands, both the 4WS and 4WS-



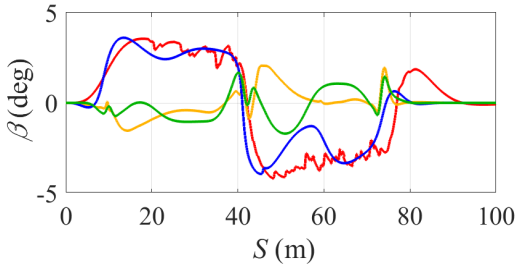
(a) Vehicle trajectory



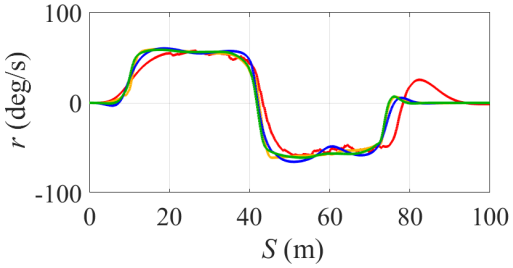
(b) g-g diagram



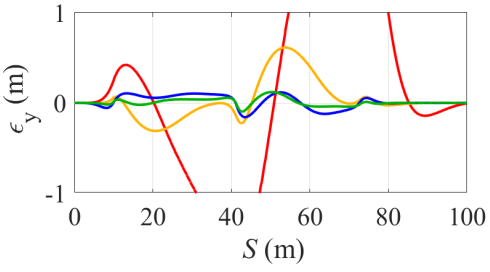
(c) Velocity



(d) Side slip angle

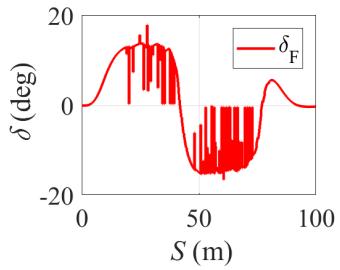


(e) Yaw rate

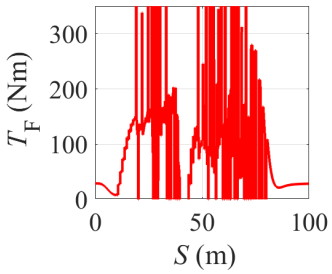


(f) Lateral tracking error

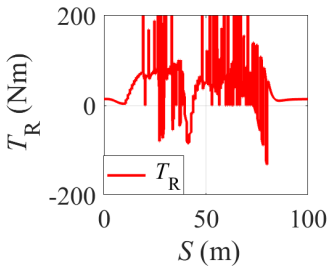
Figure 5.4: Vehicle States in U-turn



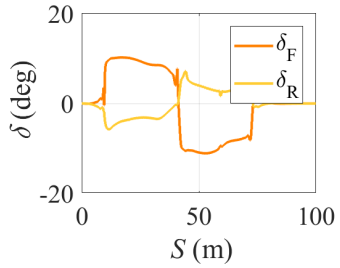
(a) Steering commands of FWS controller.



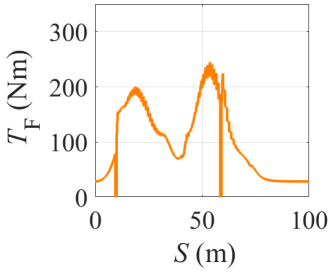
(b) Front wheel torque commands of FWS controller.



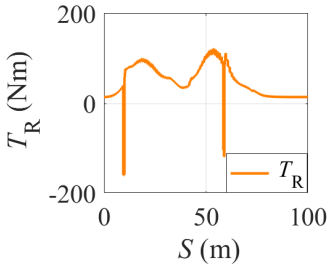
(c) Rear wheel torque commands of FWS controller.



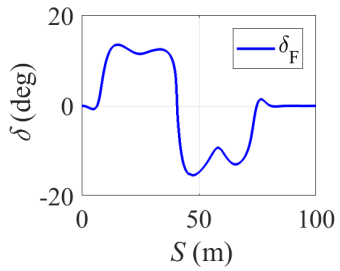
(d) Steering commands of 4WS controller.



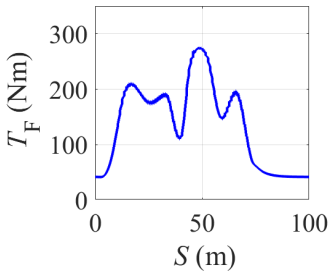
(e) Front wheel torque commands of 4WS controller.



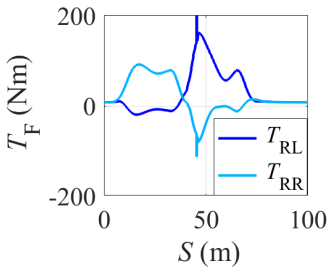
(f) Rear wheel torque commands of 4WS controller.



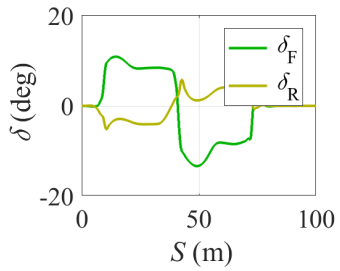
(g) Steering commands of FWS-TV controller.



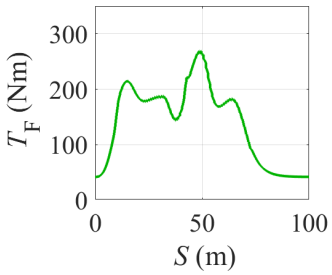
(h) Front wheel torque commands of FWS-TV controller.



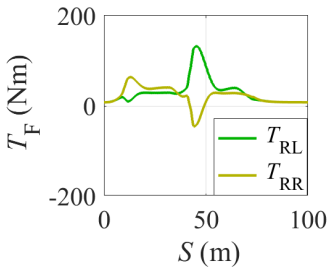
(i) Rear wheel torque commands of FWS-TV controller.



(j) Steering commands of 4WS-TV controller.

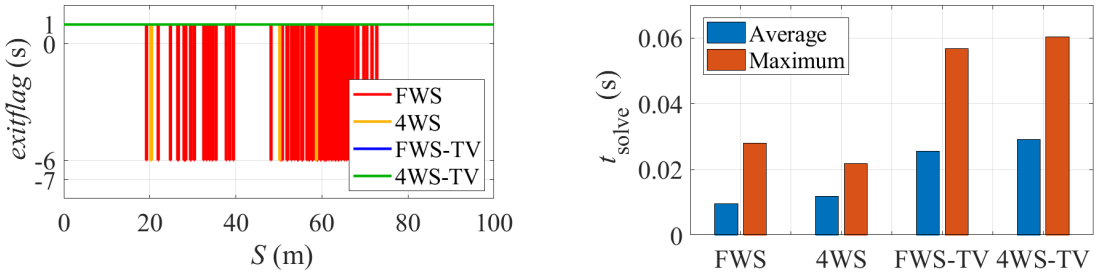


(k) Front wheel torque commands of 4WS-TV controller.



(l) Rear wheel torque commands of 4WS-TV controller.

Figure 5.5: Control commands in double-U-turn.



(a) Exitflag.

(b) Solve time.

Figure 5.6: Forces Pro solver information.

TV controllers have front and rear steering angle in opposite direction, so that a large yaw moment is generated to handle the required yaw rate. In addition, because of the application of rear steering, they both require less front steering angle than the FWS-TV controller.

With regards to the torque commands, TV can be observed with both FWS-TV and 4WS-TV controllers, which generates a yaw moment during the turning and helps to improve the path tracking performance. Hence, there is less oscillation in the torque commands of the FWS-TV and 4WS-TV controllers, indicating that TV helps to maintain the vehicle's stability by manipulating the wheel torque individually. In addition, by comparing the FWS-TV and the 4WS-TV controllers, it can be seen than the 4WS-TV controller requires less torque difference on the rear wheels at the changge of direction. This is because of the presence of rear steering, which contributes to the vehicle stabilisation.

Figure 5.6a shows the exitflag of the controllers during the simulation. Most of the bad exitflag comes from the FWS controller, which shows that the required double U-turn scenario is too challenging for the FWS-only formulation to handle, and explains the oscillation in the control commands. The 4WS controller is better, but there is still some bad exitflag at the points with curvature changing. In comparison, the FWS-TV and 4WS-TV controllers have provided a reliable performance throughout the entire path.

Figure 5.6b shows the computation time of each controller. Here the solve time refers to the time that FORCES PRO takes to solve the optimisation problem. In general, the

Table 5.3: Summary of path-tracking performance with different levels of actuation in the double U-turn scenario.

	FWS	4WS	FWS-TV	4WS-TV
RMS ε_V (m/s)	0.244	0.138	0.136	0.139
Max $\lvert\varepsilon_V\rvert$ (m/s)	0.586	0.373	0.213	0.206
RMS ε_y (m)	1.453	0.229	0.071	0.040
Max $\lvert\varepsilon_y\rvert$ (m)	3.028	0.614	0.158	0.120
Average t_{solve} (s)	0.010	0.012	0.026	0.029
Max t_{solve} (m)	0.028	0.022	0.057	0.060

computation time is related to the number of state and control variables in the formulation. The FWS controller has the shortest computation time, while the 4WS has the second shortest computation time. Despite the better path tracking performance, the FWS-TV and 4WS-TV require longer time over 0.02 s on average for computation due to the system complexity.

Table 5.3 shows a summary of the path-tracking performance with different levels of actuation in the double U-turn scenario.

5.4 Summary

This chapter compares the path-tracking performance with different levels of actuation available. By comparing the simulation results of the four controllers in double lane change and double U-turn scenarios, it is obvious that both 4WS and TV is able to improve the path tracking performance of the vehicle. TV improves the vehicle's response and stability in turning by generating a yaw moment directly. In addition to that, 4WS is able to manipulate the vehicle's motion with the direction of FWS and RWS angles, thus the vehicle's flexibility is increased. The 4WS-TV provides the best path tracking performance among the four formulations, but at the same time it requires more computation

time. Hence, further investigation is required on simplification for real-time operation. In addition, it is worth mentioning that along with the improved performance brought by the increased levels of actuation, practical disadvantages such as increased hardware complexity and higher manufacturing cost also arise. Such influence is not taken into account in this research, but is definitely required to be considered about when the multi-actuation system is to be adopted in mass production.

Chapter 6

MPC Sampling Time and Horizon for Real-time Execution

6.1 Introduction

The developed 4WS-TV controller has been validated in the previous simulation and has proven its capability in the path tracking control as well as stabilisation of autonomous vehicles. However, due to the system complexity, it takes longer time than the sampling time for computation, which prevents the controller from being implemented in real time. In this chapter, the proposed 4WS-TV controller is to be simplified for real time operation by looking for the most suitable configuration of sampling time and horizon. Once a compromise is made between low computational complexity and good control performance, the controller will be implemented in a real time target machine for practical validation.

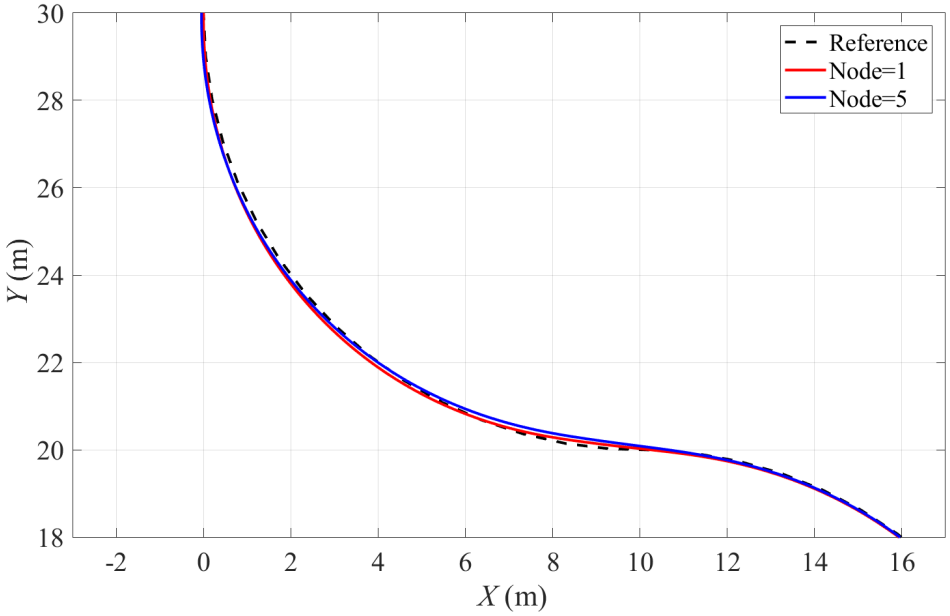
6.2 Changing Control Horizon

First of all, it is tested to reduce the frequency of control actions. This is realised by increasing the sampling time of the controller, while taking more iterations within the sampling time to compensate the loss of accuracy in system state estimation. Here the sampling time is increased from 0.02s to 0.1s, while setting up 5 nodes in the integration (N5), so that the control action is updated every 0.1s, while the internal integration of system states is equivalent to the formulation with sampling time of 0.02s (N1).

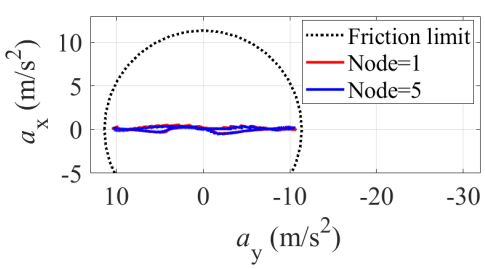
The performance of the N5 controller is tested in the double U-turn scenario, and compared with the N1 controller. Figure 6.1 shows the vehicle states with the two controllers during the simulation, and it can be seen that the two controllers have achieved similar performance in terms of the tracking of velocity and reference path. As expected, there is drawback in the path tracking performance with the N5 controller, where the N1 controller has smaller lateral tracking error. While on the other hand, it can been seen that the N5 controller does a slightly better job in tracking the reference velocity. From Figure 6.1a and 6.1f, it can be seen that with a longer control horizon, the N5 controller cannot respond promptly to rapid changes of curvature, so that the vehicle tends to cut more corners and to go wide on both of the U-turns, which could lead to a larger turning radius allowing a higher velocity.

Figure 6.2 shows the control commands of both controllers. Similar control commands are requested by the two controllers, while there is less torque differential on the rear axle with the N5 controller.

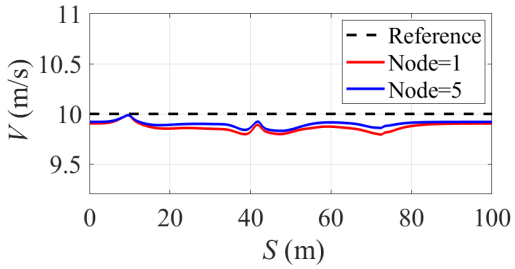
Figure 6.3 shows the computation time of the two controllers, and it can be seen that the N5 controller massively reduces the computation burden. The sampling time of the N5 controller is 100ms, and it has a mean computation time of 8.7ms. In comparison, the N1 controller has a sampling time of 20ms, while it costs around 28.5ms on average for computation. In order to implement the controller in real time, it is essential to assure



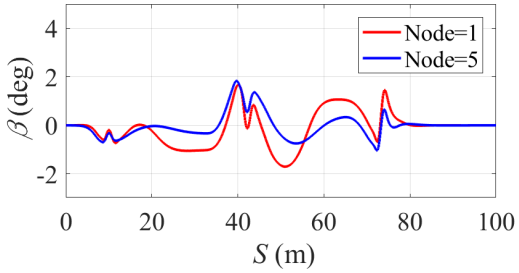
(a) Vehicle trajectory



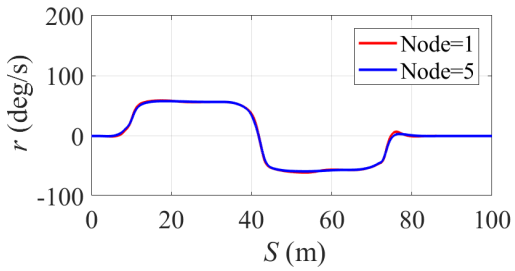
(b) g-g diagram



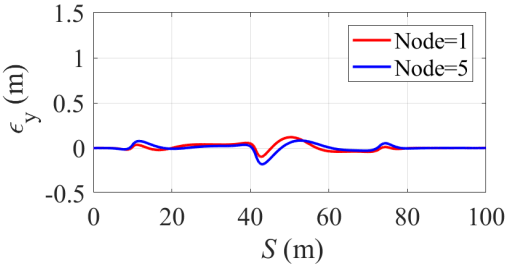
(c) Velocity



(d) Side slip angle

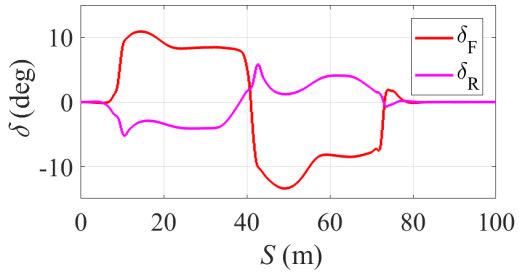


(e) Yaw rate

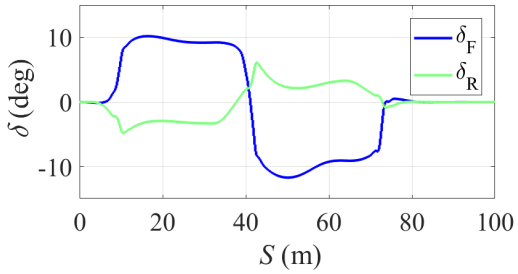


(f) Lateral tracking error

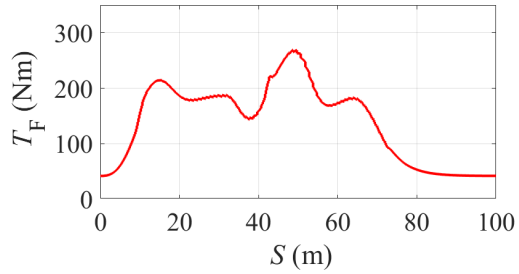
Figure 6.1: Vehicle States in U-turn



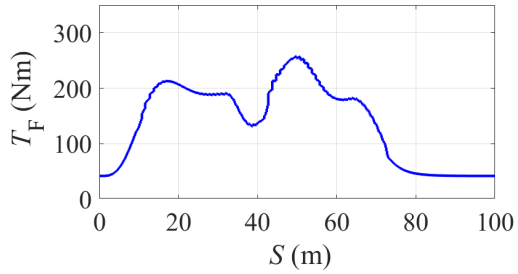
(a) Steering commands of N1 controller.



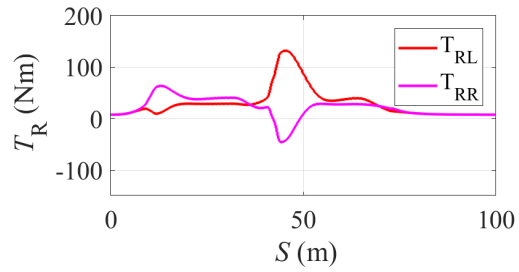
(b) Steering commands of N5 controller.



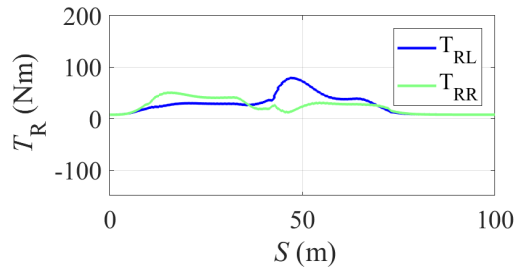
(c) Front wheel torque commands of N1 controller.



(d) Front wheel torque commands of N5 controller.



(e) Rear wheel torque commands of N1 controller.



(f) Rear wheel torque commands of N5 controller.

Figure 6.2: Control commands in double-U-turn.

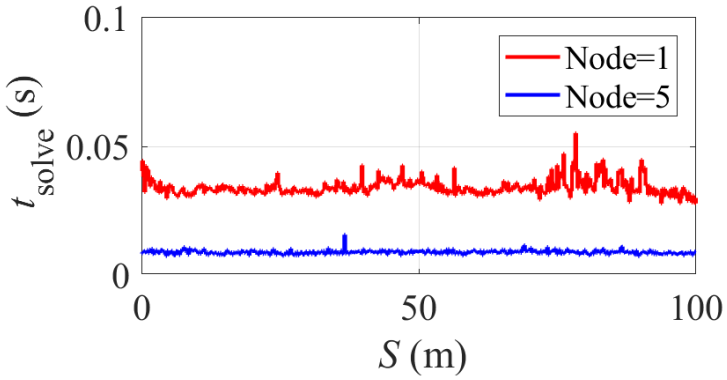


Figure 6.3: Solve time.

Table 6.1: Summary of path-tracking performance of N1 and N5 controllers in the double U-turn scenario.

	N1	N5
RMS ε_V (m/s)	0.139	0.105
Max $\lvert\varepsilon_V\rvert$ (m/s)	0.206	0.169
RMS ε_y (m)	0.040	0.043
Max $\lvert\varepsilon_y\rvert$ (m)	0.120	0.181
Average t_{solve} (s)	0.029	0.009
Max t_{solve} (m)	0.060	0.016

that the computation time is consistently below the sampling time, which determines the frequency that the solver is executed in. Considering about the computation time as well as sampling time of the two controllers, the N1 controller is not able to be running in real time, but the N5 controller is supposed to be real-time implementable, which is a significant improvement with the modified formulation. What's more, the control performance is maintained with the N5 controller. As the result, the further control development is based on the N5 formulation.

Table 6.1 shows a summary of the path-tracking performance of the N1 and N5 controllers in the double U-turn scenario.

6.3 Changing Prediction Horizon

Based on the N5 formulation, here different prediction horizon time is attempted to evaluate the control performance on the vehicle. The sampling time of the controllers is fixed to 0.1s with 5 internal integration steps, while the prediction horizon varies from 0.6s to 1.4s.

The controllers are tested in the double U-turn scenario, and the simulation results of vehicle performance are shown in Figure 6.4. Generally speaking, better path tracking is

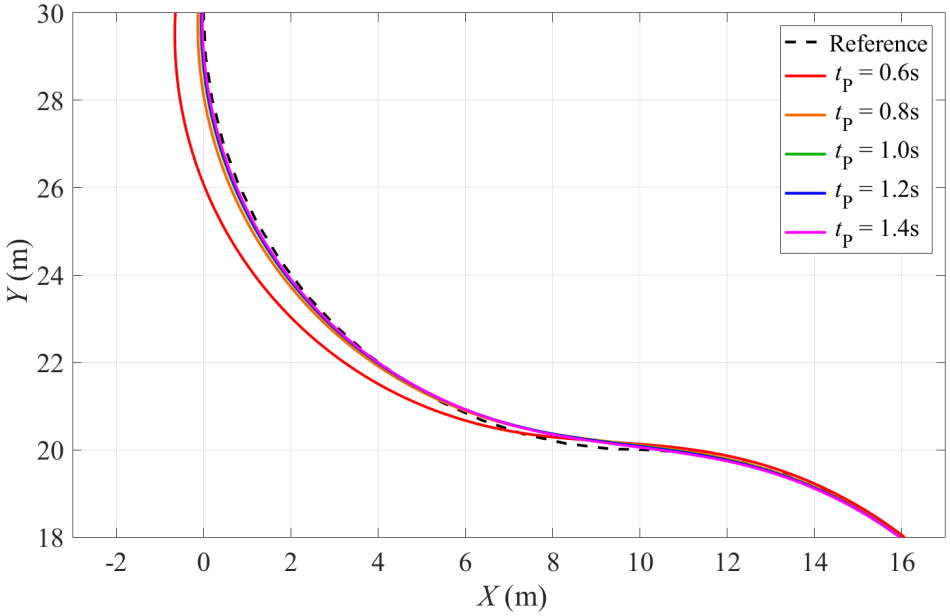
achieved with longer prediction horizon. Largest lateral tracking error takes place with 0.6s prediction horizon, while in comparison, with the longest horizon time of 1.4s, the lateral tracking error is maintained within 0.15m even at the harsh change of direction. This proves that longer prediction horizon does help to improve the control performance.

Figure 6.5 shows the control commands of the controllers. Basicly the controller commands are similar among the controllers, but with the shortest prediction horizon of 0.6s, the controller is not able to plan the manoeuvre well, and thus the control commands are fluctuating and quite deviated from the other controllers.

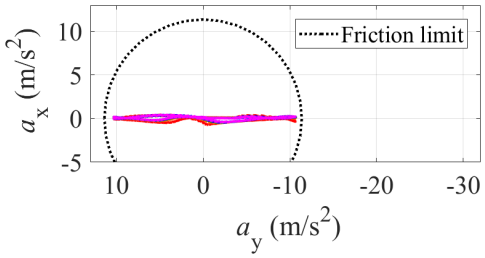
On the other hand, longer prediction horizon increases the computational burden. Figure 6.6b shows the computation time of the controllers with different prediction horizon. It can be seen that a longer prediction horizon leads to longer computation time. The controller with 0.6s horizon has an average computation time of 5.8ms, while the one with 1.4s horizon has an average computation time of 13.0ms. Furthermore, from Figure 6.6a, it can be seen that there is exitflag other than 1 taking place with the 1.4s horizon, indicating that there is computational error caused by the huge computation led by the long prediction horizon.

By comparing the control performance under different prediction horizon, it can be seen that the prediction horizon of 1 s provides the best compromise between performance and computation time. Thus it is further evaluated with different sampling time in the next section.

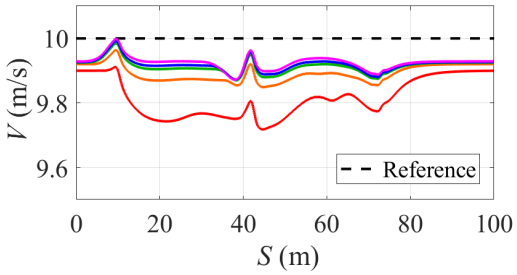
Table 6.2 shows a summary of the path-tracking performance with different prediction horizon in the double lane change scenario.



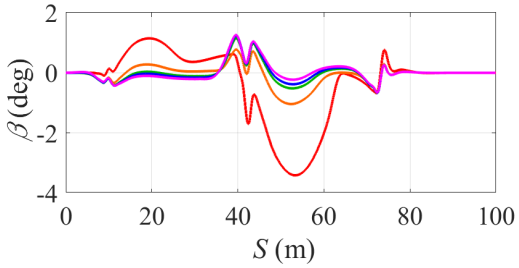
(a) Vehicle trajectory



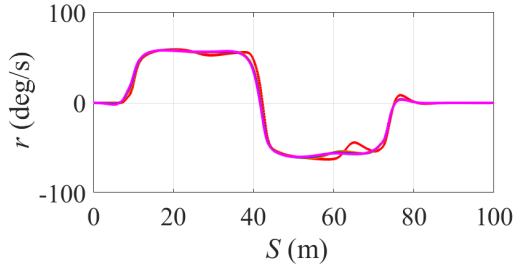
(b) g-g diagram



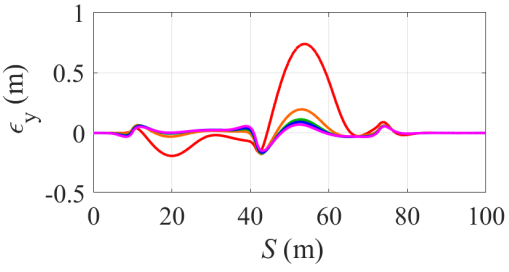
(c) Velocity



(d) Side slip angle

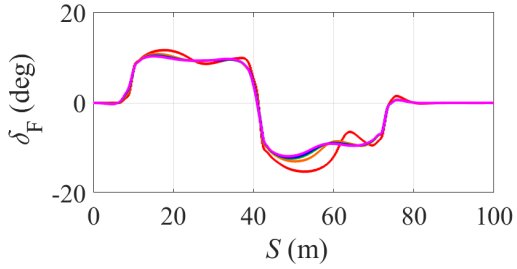


(e) Yaw rate

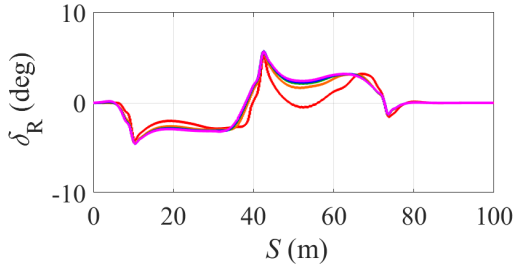


(f) Lateral tracking error

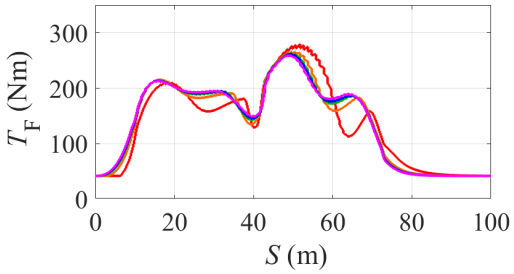
Figure 6.4: Vehicle States in U-turn



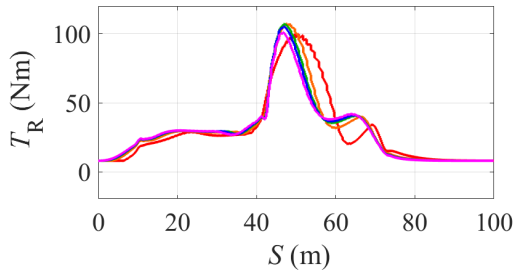
(a) Front steering commands.



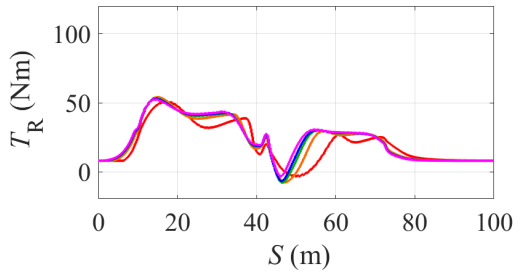
(b) Rear steering commands.



(c) Front wheel torque commands.

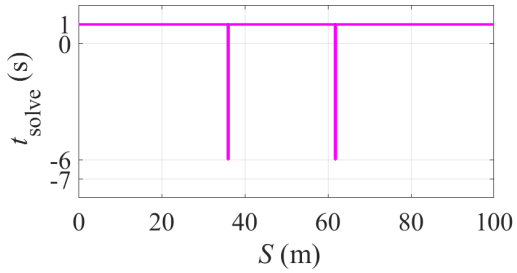


(d) Rear left wheel torque commands.

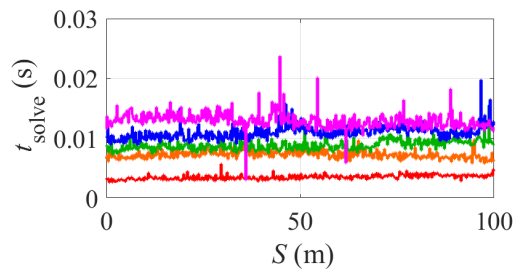


(e) Rear right wheel torque commands.

Figure 6.5: Control commands in double-U-turn.



(a) Exitflag.



(b) Solve time.

Figure 6.6: Forces Pro solver information.

Table 6.2: Summary of path-tracking performance with different prediction horizon in the double U-turn scenario.

t_p (s)	0.6s	0.8s	1.0	1.2	1.4
RMS ε_V (m/s)	0.194	0.111	0.090	0.084	0.077
Max $\lvert\varepsilon_V\rvert$ (m/s)	0.281	0.151	0.130	0.127	0.128
RMS ε_y (m)	0.249	0.063	0.045	0.042	0.038
Max $\lvert\varepsilon_y\rvert$ (m)	0.740	0.196	0.171	0.164	0.151
Average t_{solve} (s)	0.004	0.007	0.009	0.011	0.013
Max t_{solve} (s)	0.006	0.010	0.156	0.020	0.024

6.4 Changing Sampling Time

Based on the prediction horizon of 1s, the control performance of the 4WS-TV controller is then tested with various sampling time from 0.04s to 0.25s. The internal integration is fixed to 5 steps.

Figure 6.7 shows the vehicle states during simulation. The 0.25s sampling time provides the worst tracking performance, especially at the end of the double U-turn, dramatic fluctuation can be noticed in the vehicle states. This is because higher sampling time leads to lower control frequency, which means that the controller is not able to respond to the quick change in vehicle dynamical condition. In addition, with higher sampling time, the accuracy of state estimation during the internal process of MPC is reduced. With the reduction of the sampling time, the steady-state tracking error of velocity reduces, and generally the lateral tracking error gets smaller.

Figure 6.8 shows the control commands of the controllers with different sampling time. There is large oscillation in the commands of the controller with 0.25s sampling time, which is the reason of the fluctuation in vehicle states. It can also be seen that some deviation takes place with the shortest sampling time of 0.04s.

From Figure 6.9a, it can be seen that some bad exitflag happens with the longest and

Table 6.3: Summary of path-tracking performance with different sampling time in the double U-turn scenario.

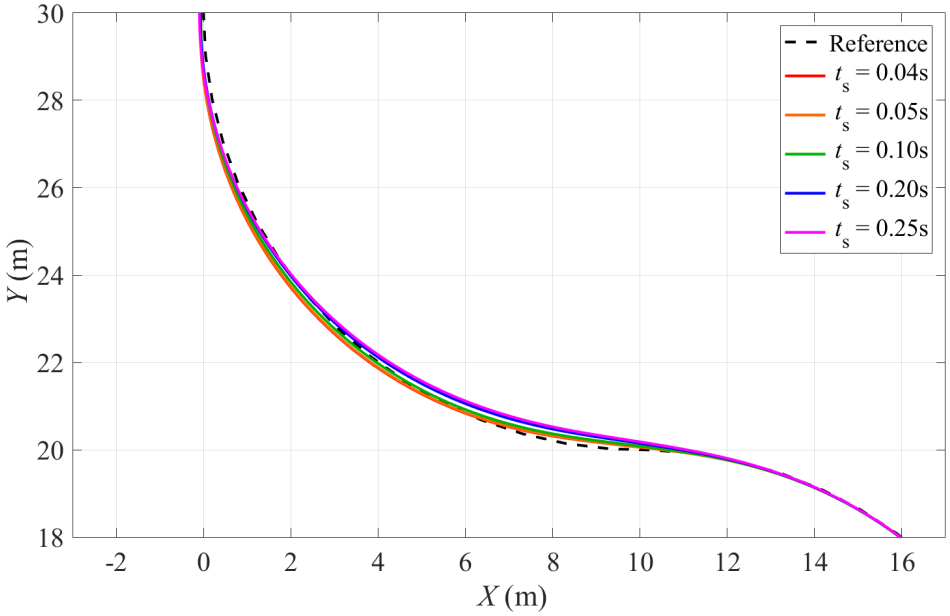
t_s (s)	0.04s	0.05s	0.10	0.20	0.25
RMS ε_V (m/s)	0.083	0.081	0.090	0.112	0.154
Max $\lvert\varepsilon_V\rvert$ (m/s)	0.135	0.122	0.126	0.155	0.455
RMS ε_y (m)	0.068	0.051	0.045	0.067	0.132
Max $\lvert\varepsilon_y\rvert$ (m)	0.180	0.169	0.171	0.268	0.452
Average t_{solve} (s)	0.028	0.020	0.009	0.004	0.003
Max t_{solve} (m)	0.074	0.026	0.156	0.006	0.004

shortest sampling time, which explains the oscillation in control commands with these two controllers. Figure 6.9b shows the computation time of the controllers. It can been a shorter sampling time leads to longer computation time, due to the increased prediction steps for the solver to compute.

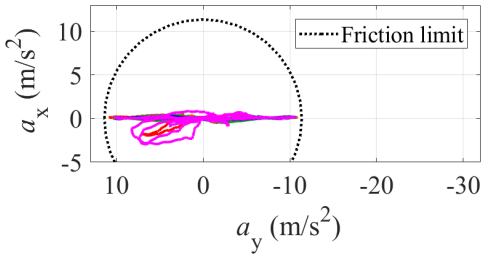
Table 6.3 shows a summary of the path-tracking performance with different sampling time in the double lane change scenario.

6.5 Summary

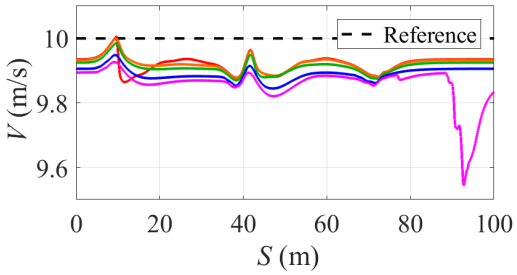
In this chapter, the control performance of the proposed 4WS-TV controller is further evaluated with various sampling time and prediction horizon, to find out the most suitable configuration of the NMPC formulation for real-time implementation. First of all, it is tried to reduce the control frequency by increasing the sampling time of the formulation, while introducing additional integration steps for system state estimation. According to the simulation, the N5 controller with longer control horizon but equivalent sampling time for integration achieve similar control performance with the N1 controller, while the computation time is significantly reduced so that the N5 controller is supposed to be able to run in real time.



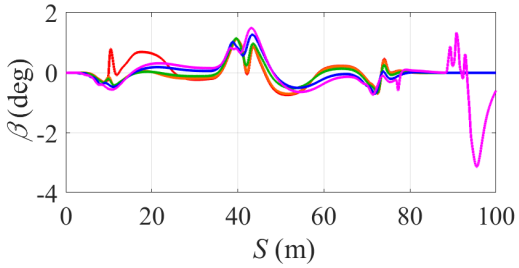
(a) Vehicle trajectory



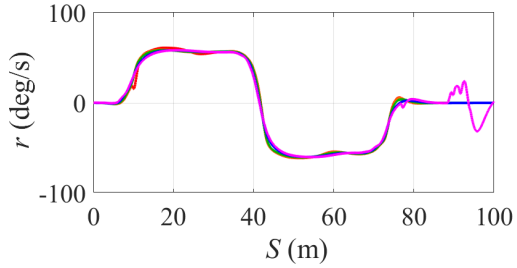
(b) g-g diagram



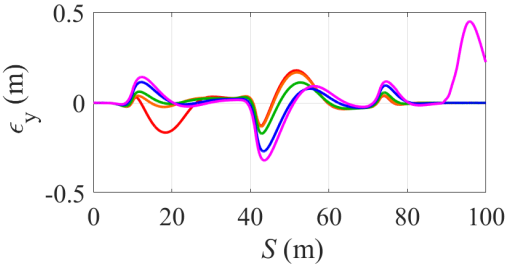
(c) Velocity



(d) Side slip angle

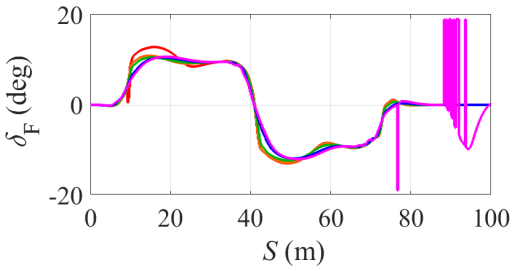


(e) Yaw rate

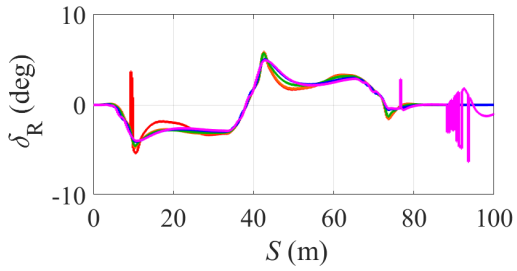


(f) Lateral tracking error

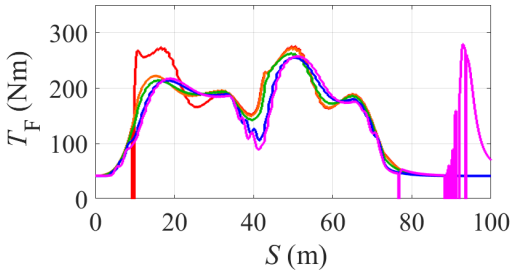
Figure 6.7: Vehicle States in U-turn



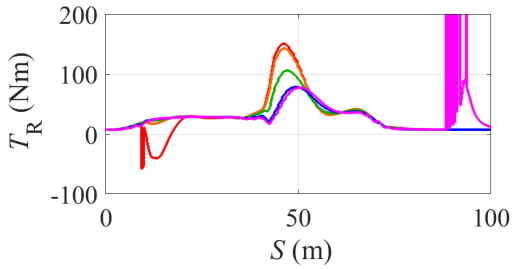
(a) Front steering commands.



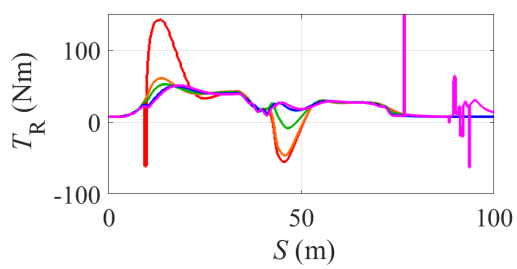
(b) Rear steering commands.



(c) Front wheel torque commands.

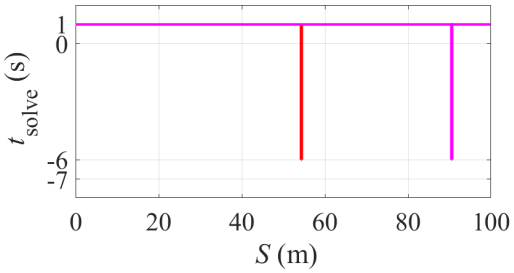


(d) Rear left wheel torque commands.

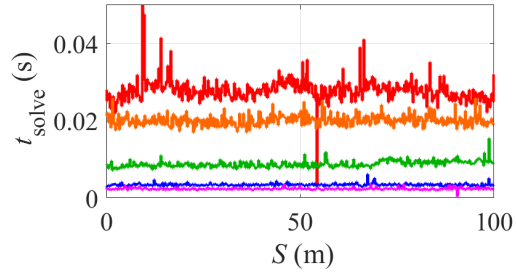


(e) Rear right wheel torque commands.

Figure 6.8: Control commands in double-U-turn.



(a) Exitflag.



(b) Solve time.

Figure 6.9: Forces Pro solver information.

Next, the controller is tested based on the N5 formulation with different prediction horizon. It can be seen that in general, better control performance can be achieved with longer prediction horizon, because a longer prediction horizon helps the controller to better plan the manoeuvre according to the objectives. However, a longer horizon also leads to longer computation time due to the additional computational burden. Besides, a few bad exitflag takes place with the longest prediction horizon of 1.4s, which is probably due to the largest computational burden on the solver that makes it difficult to find the optimal solution.

The controller is then tested with different sampling time. With the same prediction horizon, a longer sampling time means that less integration steps is carried out during the same period of time, leading to less accuracy in the estimation of the system states. This is proven in the simulation, where worse path tracking performance in general is achieved with the longer sampling time. Given the same prediction horizon of 1s, a sampling time of 0.25s means that the frequency of the control action is only 4Hz, and thus the controller is not able to respond quickly to the change of condition. On the other hand, longer sampling time helps to reduce the required computation, which benefits the real-time operation of the controller. The sampling time of 0.04s generates quite much computational burden as what a long prediction horizon does, but the control performance is not obvioustlt improved. What's more, the additional computation also makes it more difficult for the MPC solver to find out the optimal solution, as indicated by the bad exitflag during simulation.

It needs no be mentioned that the testing vehicle has got a limitation on the fastest control frequency it can take, as the steering commands can only be updated every 100ms. According to this as well as all the simulation results in this chapter, the setup with 5 integration steps within 0.1s sampling time and 1s prediction horizon makes the best compromise between control performance and the computation time, and thus will be applied for implementation in real time for practical testing.

Chapter 7

Practical Testing

7.1 Introduction

In this chapter, the developed 4WS-TV controller is implemented on a real vehicle for practical validation, with the sampling time and prediction horizon time setup discussed in the last chapter. A real time target machine installed on the vehicle is utilised to execute the control system, and the vehicle is supposed to carry out several manoeuvres that require the operation at the limits of handling.

It should be mentioned that during the practical testing session, the real-time computation capability was not as easily met as it was in the simulation. There has been solve time or possible convergence issues with the solver indicated by a CPU overload problem with the real time target machine when running the proposed control algorithm. The problem took place occasionally, but this could lead to failure in control performance and even serious safety issue. In order to prevent the CPU overload problem from happening while retaining the sampling time and prediction horizon time setup, simplification has been done in the system model of MPC, by removing the coupling of longitudinal and lateral tyre force. In particular, the saturation of maximum lateral tyre force set by Equation 3.16

is removed, while assuming that the maximum force is determined simply by the vertical tyre load as well as the tyre-road friction coefficient. This simplification helps to reduce the computation time of the control algorithm, and thus to guarantee real-time operation of the controller. It is expected that there is some drawback in the path-tracking performance during the practical testing, while according to the final test results, the control performance is still acceptable. During the practical testing session, it was mainly focused on the lateral dynamics of the vehicle, and significant change in longitudinal velocity is not required, which could be one of the reasons that the drawback in control performance is not obvious. Further research will be carried out in the future to investigate the detailed impact of the specific simplification, and possible approaches to guarantee real-time operation with presence of the saturation in practice.

This chapter first demonstrates the vehicle used in the practical testing and the testing environment, followed by the hardware and software configuration of the control system. Then the results of the practical testing are shown in this chapter. The vehicle has been tested in four scenarios, including double lane change, constant radius turning, figure 8 tracking and general path tracking. In order to evaluate the effects of the controller, simulation is carried out in the same scenarios and conditions, and the results are shown for comparison.

7.2 Configuration

7.2.1 Testing Platform and Environment

The testing platform used for the practical testing session is provided by Delta Motorsport under the AID-CAV project. AID-CAV stands for ”Advanced Integrated Dynamics for Connected and Autonomous Vehicles”. The project is sponsored by UK Research and Innovation (UKRI), and aims to develop key ”vehicle platform” technologies for the next

generation of autonomous vehicles, which are supposed to have enhanced capability and performance [88].

As introduced before, the platform is equipped with 4WS capability, and TV functionality on the rear axle. The front wheels are driven by a single motor through differential. Figure 7.1 shows a photo of the platform in the testing area. This vehicle provides a great opportunity to validate the control strategy developed in this thesis in practical operation.

Figure 7.1: The AID-CAV Phase 2B vehicle platform used as the testing vehicle.

Figure 7.2: Satellite birdview of the MUEAVI testing area.

The practical testing is carried out in part of the the MUEAVI area in Cranfield University. MUEAVI refers to "Multi User Environment for Autonomous Vehicle Innovation", which aims to provide both urban and off-road environments for autonomous vehicle testing [89]. Figure 7.2 shows a birdview of the testing area in Google Map.

7.2.2 Hardware Setup

The testing vehicle is equipped with the following real-time processing devices: a dSpace MicroAutoBox II, a Speedgoat Mobile and an OxTS RT1003. The Speedgoat Mobile is where the developed control algorithm in this thesis is implemented. It takes the all the necessary status information of the testing vehicle and determines the optimal control commands for the vehicle. The MicroAutoBox II acts as the main ECU of the vehicle. It runs the master control system developed by Delta Motorsport, which communicates with all the sensors and actuators of the steer-by-wire system (SBW), brake-by-wire system (BBW) and inverters (INV). MicroAutoBox II is responsible for monitoring the vehicle status, providing information to the Speedgoat Mobile and passing the control commands to the actuators. The OxTS RT1003 is an inertial navigation systems which is able to provide the measurements of position, orientation and velocity of the vehicle. The communication among the devices is based on Controller Area Network (CAN), and figure 7.3 shows the hardware topology of the testing vehicle. The detailed specifications of the processing devices are listed in Appendix D.

With regards to the measurements by OxTS RT1003, it is worth to mention that real-time kinematic positioning (RTK) is enabled to enhance the accuracy of measurements, which is realised with a base station providing real-time corrections to the GNSS data. Table 7.1 shows the average measurement accuracy of the localisation signals with and without RTK during the practical testing..

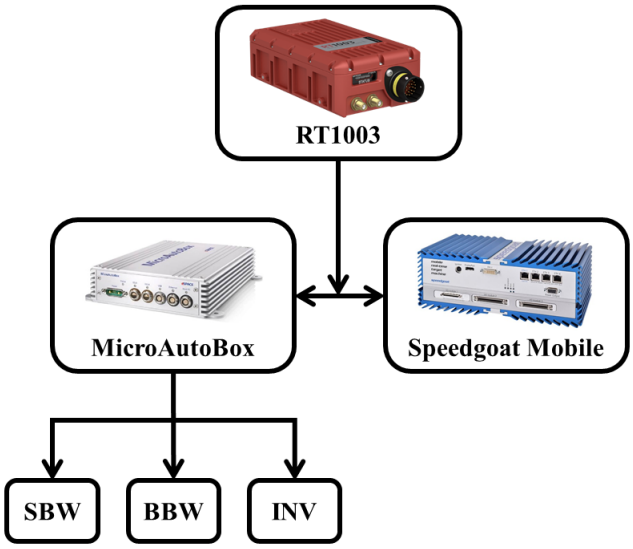


Figure 7.3: Hardware topology of the testing vehicle for data processing.

Table 7.1: Comparison of measurement accuracy with and without RTK

	Without RTK	With RTK
Position accuracy north (m)	0.3719	0.0136
Position accuracy east (m)	0.3708	0.0136
Position accuracy down (m)	0.8999	0.0147
Velocity accuracy north (m/s)	0.0302	0.0110
Velocity accuracy east (m/s)	0.0302	0.0110
Velocity accuracy down (m/s)	0.0367	0.0055
Yaw angle accuracy (deg)	0.0760	0.0628
Pitch angle accuracy (deg)	0.0564	0.0379
Roll angle accuracy (deg)	0.0556	0.0379

7.2.3 Control Structure

Figure 7.4 shows the diagram of the entire control algorithm to be running in the Speedgoat. First of all, the vehicle status is received and processed in the CAN Rx block. Next, the MPC controller collects all the required information of the vehicle status, and finds out the optimal control commands for the intended manoeuvre. Before the commands are sent out, the system condition including communication, vehicle status and ForcesPro solver output are inspected by the safety controller. If anything goes wrong, the safety controller will void the control commands or request emergency stop to avoid incidents. Finally, the control commands together with necessary diagnostic information are transmitted to the MicroAutoBox. The control algorithm runs in multi-rate during real-time operation. The sampling time of the MPC controller is 0.1s as discussed in Chapter 6, while the other parts of the control system run in the sampling time of 0.01s.

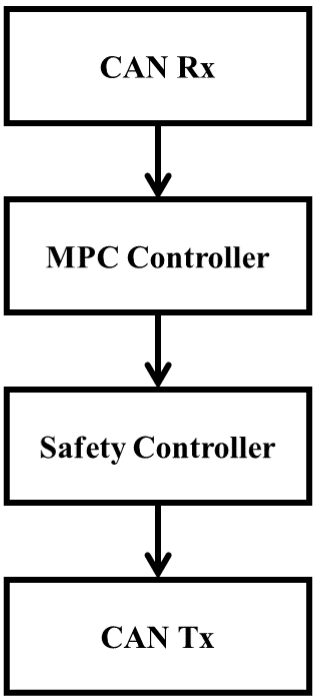


Figure 7.4: System Setup of the Control Algorithm.

Figure 7.5 shows the state flow of the control system included in the safety controller. The key purpose of the safety controller is to make sure that only reliable commands is trans-

mitted for execution. First of all, critical messages received from the CAN are inspected according to two diagnostic information to make sure that correct data is received. One information is the CRC checksum attached to the messages, which is used to identify if the received data is genuine. 8-bit CRC Checksum is used in this project, and SAE CRC8 J1850 is used for CRC parameterisation. It provides reliable CRC check of the CAN messages in this project, while requiring less computation compared to CRC algorithms with more bits. It should be mentioned that the CRC checksum is also applied on the messages to be sent by the Speedgoat. The other diagnostic information is the counter signal coming with the messages. This helps the controller to make sure if the data is received in time. If there is inconsistency in the increment of the counter signal, it is suggested that a delay in transmission has happened.

Besides the data transmission condition, the safety controller also monitors the exitflag of the MPC solver as well as the condition of the OxTS. Failure in finding the optimal solution of the MPC problem, or in the localisation of the vehicle, can also lead to the ”Fault” condition of the control system. As a critical information, the states of the control system is sent to the MicroAutoBox. Automation will only be activated when the control system is in ”Ready”, and a ”Fault” condition will lead to the emergency stop of the vehicle.

7.3 Testing Results

In this section, the results of the practical testing are demonstrated. The performance of the developed controller is tested in four scenarios, including double lane change, constant radius turning, figure 8 tracking and general path tracking. Double lane change is a standard manoeuvre used in the industry to evaluate vehicle stability characteristics. Constant radius turning is another standard benchmark to evaluate the steady-state handling performance of the vehicle at its limits of handling. In addition, figure 8 tracking

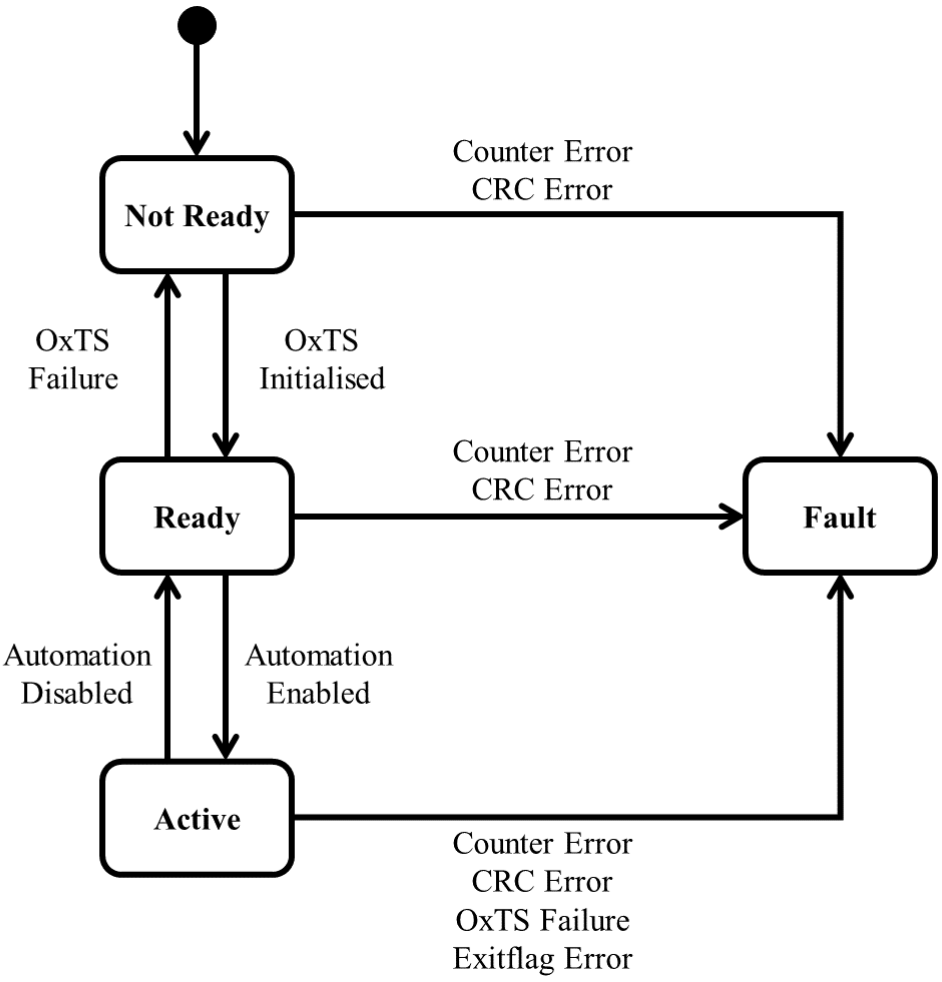


Figure 7.5: State flow of Speedgoat Operation within Safety Controller.

is a good extension to constant radius turning. It introduces a transient process into the scenario, which makes it a much more demanding manoeuvre. Last but not least, a general path-tracking scenario is also introduced in practical testing as it is close to real life driving. Beside normal normal path tracking, more challenging tests are carried out with additional references of vehicle states provided to the controller, which includes heading angle and side slip angle in order to manipulate vehicle attitude during path tracking.

In order to prepare for the practical testing, simulation is carried out in advance within the same scenario and condition. A model of the MUEAVI testing area is built in CarMaker, which is shown in Figure 7.6. This helps to have a better idea of the prospective vehicle performance in real time. The simulation results are also demonstrated here to have a comparison of the simulation and testing performance.

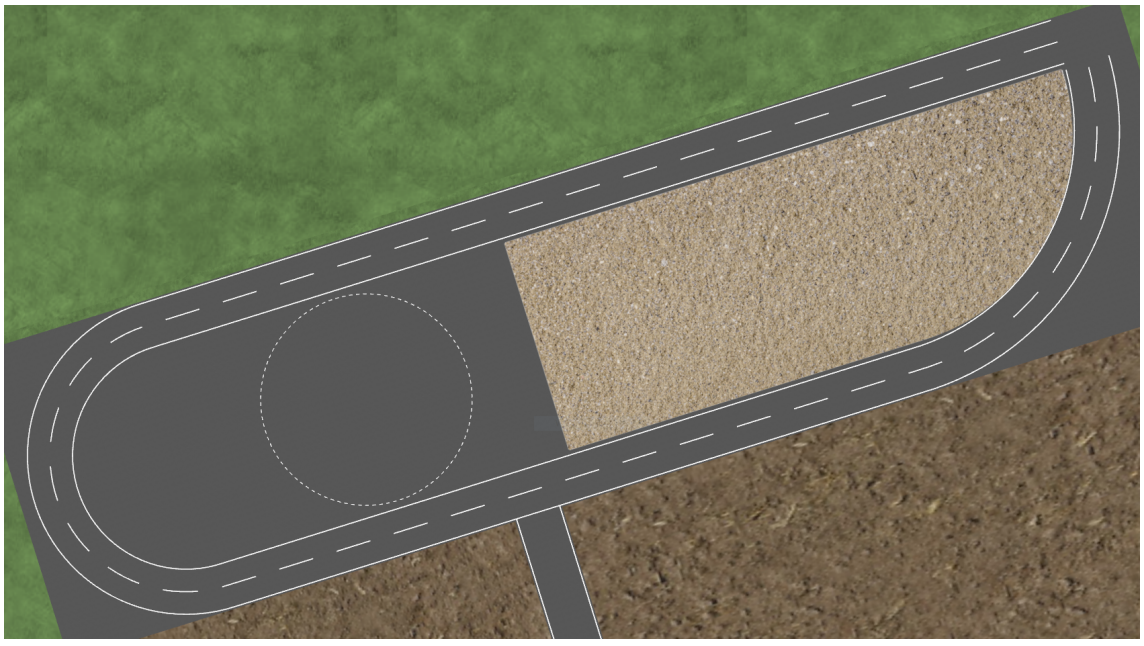


Figure 7.6: Model of MUEAVI track in CarMaker.

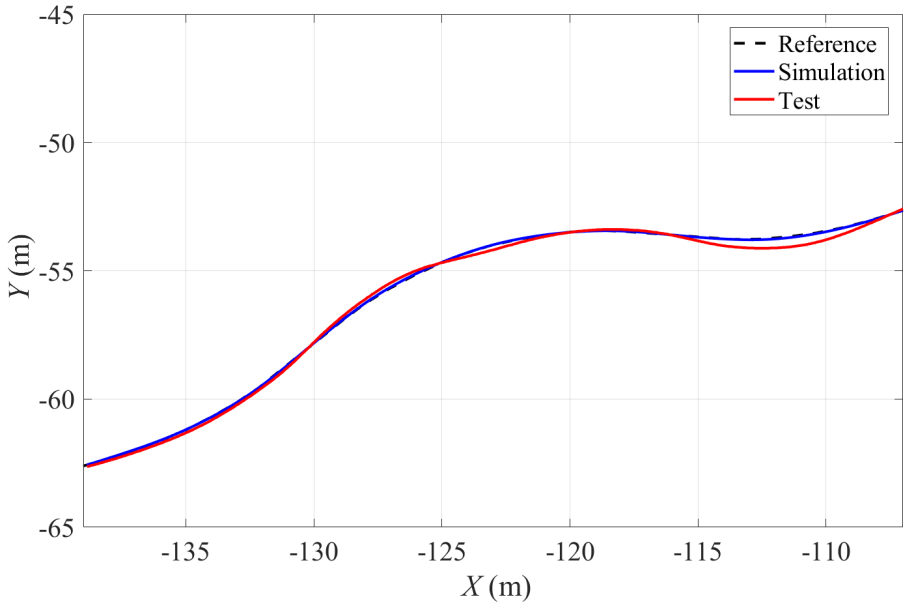
7.3.1 Double Lane Change Scenario

Double Lane Change without Heading Angle Reference

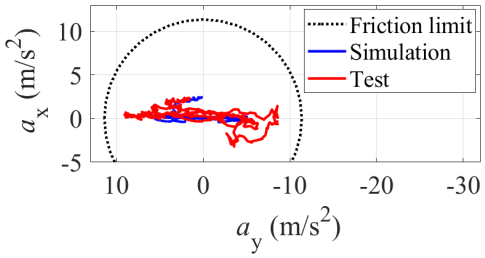
The developed control algorithm is first tested in a double lane change scenario. Due to the limitation of testing area, the standard double lane change track cannot be carried out in practical testing, and thus a substitute double lane change path with reduced length is used as reference in practical testing. The dimension of the path is available in Appendix B.3.

The controller is first supposed to track the reference path at a constant speed of 8m/s. This is not the highest velocity that the vehicle can achieve on the specific path, but is the maximum due to the limitation of testing area and safety reasons. Figure 7.7a shows the trajectory of the vehicle in simulation and experiment, and similar path tracking results have been achieved. Figure 7.7b shows the acceleration of the vehicle, and it can be seen that the vehicle does not reach the friction limit. The acceleration is up to $6m/s^2$ in simulation, and is up to $9m/s^2$ in the experiment due to oscillation. As shown in Figure 7.7c, neither in simulation nor in the testing is the vehicle able to reach the reference velocity before entering the double lane change, which is because of the lack in distance for acceleration. After reaching the reference velocity, the controller is able to maintain the velocity tracking error within 0.1m/s. In comparison, larger variation can be noticed during the path tracking process. Similar situation can be noticed in Figure 7.7d, where the vehicle's side slip angle is small in simulation, but there is larger fluctuation in the experiment. Both simulation and experiment have similar yaw rate response. Figure 7.7f shows the lateral tracking error of the vehicle. It can be seen that the vehicle does a better job in the simulation with maximal position deviation of 0.04m, while in the practical testing the lateral deviation increases to 0.4m. The maximum tracking error takes place at the exit of the double lane change, and is reduced by the controller soon afterwards.

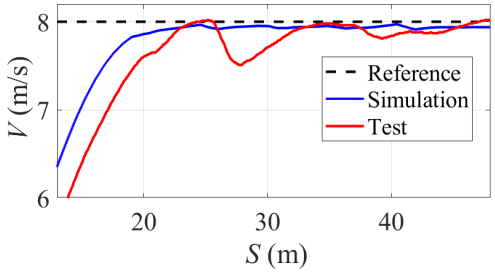
Figure 7.8 shows the control commands during the double lane change scenario. From



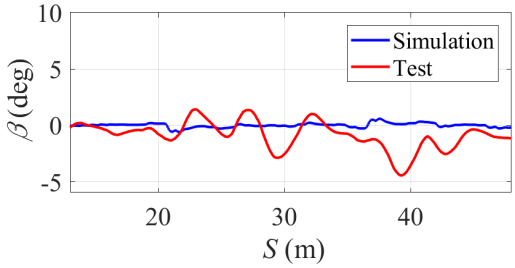
(a) Vehicle trajectory



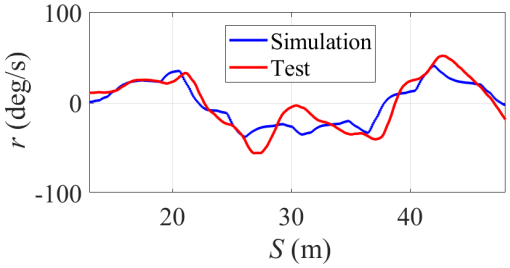
(b) g-g diagram



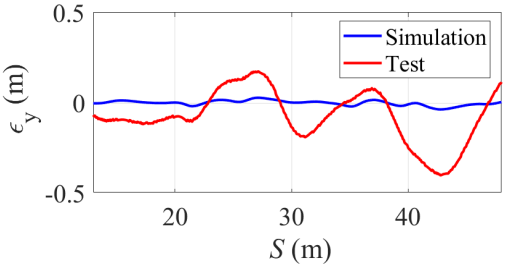
(c) Velocity



(d) Side slip angle

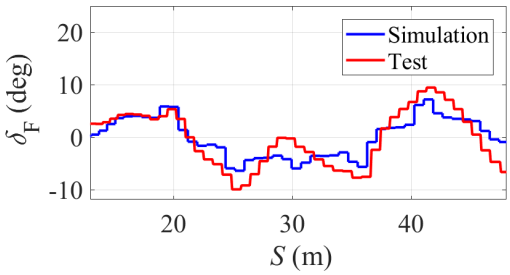


(e) Yaw rate

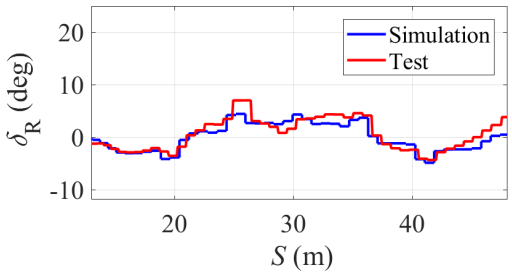


(f) Lateral tracking error

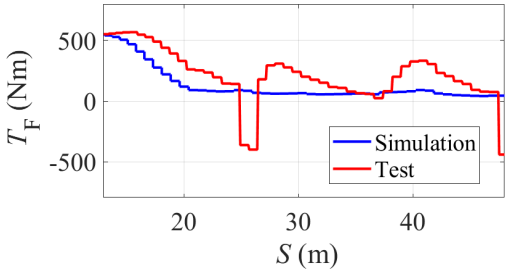
Figure 7.7: Vehicle states during double lane change without heading angle reference.



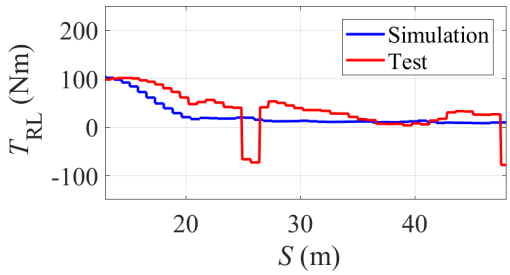
(a) Front steering commands



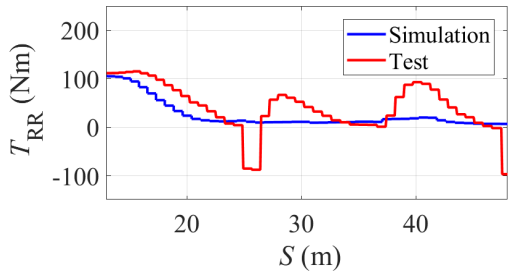
(b) Rear steering commands



(c) Front wheel torque commands.

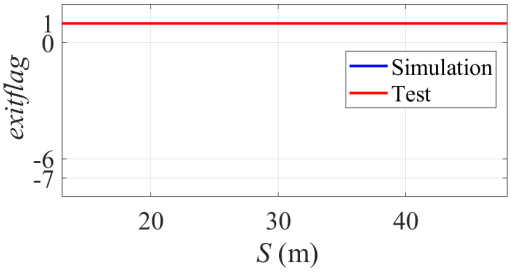


(d) Rear left wheel torque commands.

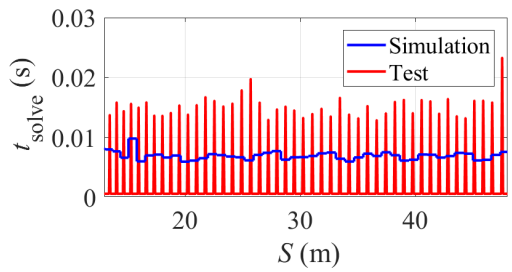


(e) Rear right wheel torque commands.

Figure 7.8: Control commands during double lane change without heading angle reference.



(a) Exitflag



(b) Computation time

Figure 7.9: Forces Pro solver information during double lane change without heading angle reference.

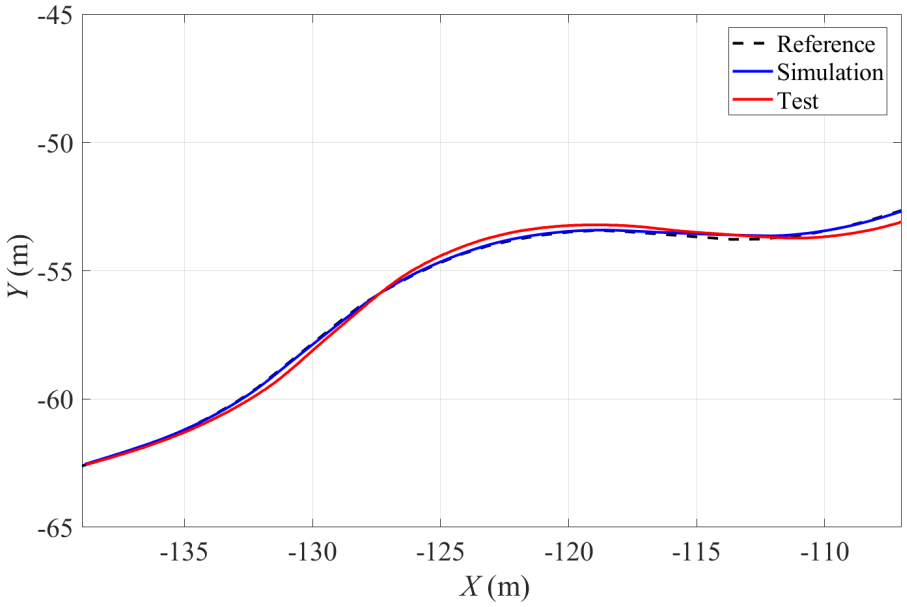
Figure 7.8a and Figure 7.8a, it can be seen that similar steering commands are requested by the controller in simulation and experiments. By comparing the front and rear steering angles, it can be told that opposite front and rear steering is demanded to generate the required yaw moment for turning, which is in accordance with the results in the previous simulation. In terms of the wheel torque commands, a gentle deceleration can be observed in the experiments, which does not take place in the simulation. The wheel torque commands remain small after the vehicle gets close to the reference velocity, but in the practical testing more torque is required during turning.

Figure 7.9a shows the exitflag of the MPC solver in both simulation and experiment, and it is suggested that the controller is able to find the optimal solution throughout both situations. The computation time in both situations is shown in Figure 7.9b.

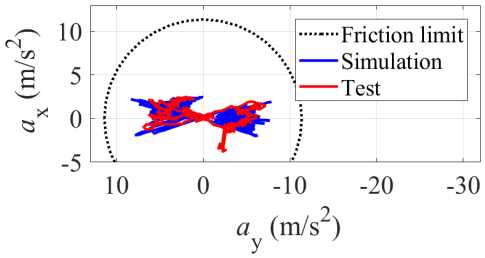
Double Lane Change with Heading Angle Reference

Following the normal double lane change, additional reference is provided on the heading angle of the vehicle. The vehicle is supposed to track the double lane change path at the constant velocity of 8m/s, and with a fixed heading angle parallel to the initial orientation of the reference path.

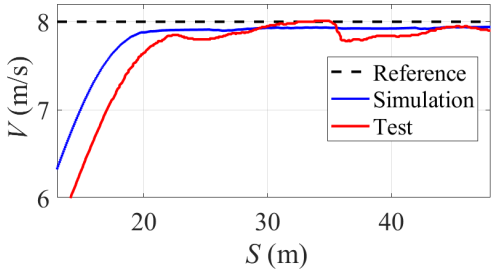
Figure 7.10 shows the vehicle states during the double lane change with heading reference. The acceleration of the vehicle is up to $9m/s^2$. The controller again tracks the reference velocity well in simulation with steady-state error within 0.1m/s. There is a slight drop in velocity in the experiment, but the deviation is up to 0.2m/s. Figure 7.10d shows the side slip angle of the vehicle, and it can be seen that side slip angle increases gradually to around 19° in both direction to negotiate the double lane change scenario with fixed heading angle, which results in the yaw rate of the vehicle close to zero in both simulation and experiment as shown in Figure 7.10e. The controller achieves similar trajectories in simulation and experiment, and the lateral tracking error is shown in Figure 7.10f. The lateral deviation is up to 0.18m in simulation, while exceeds 0.4m in the



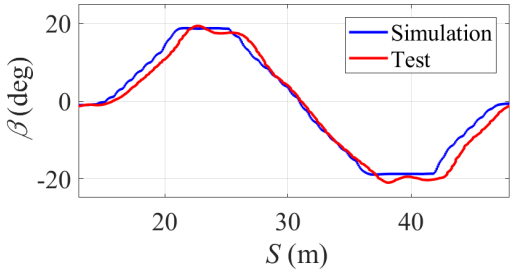
(a) Vehicle trajectory



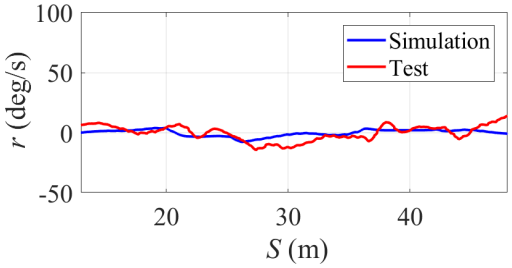
(b) g-g diagram



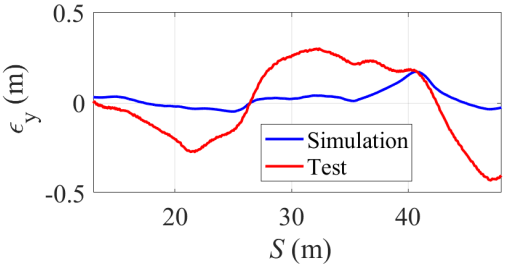
(c) Velocity



(d) Side slip angle

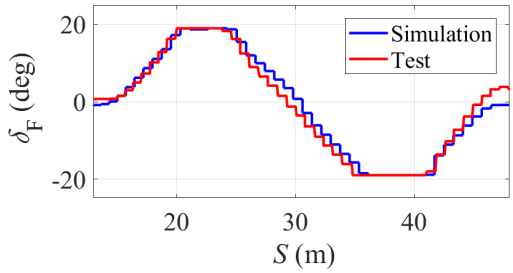


(e) Yaw rate

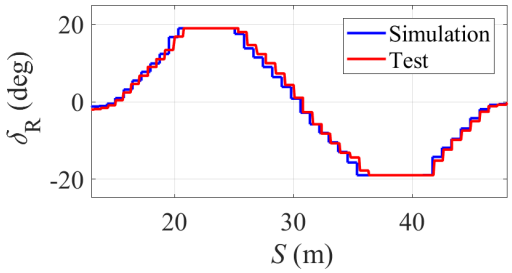


(f) Lateral tracking error

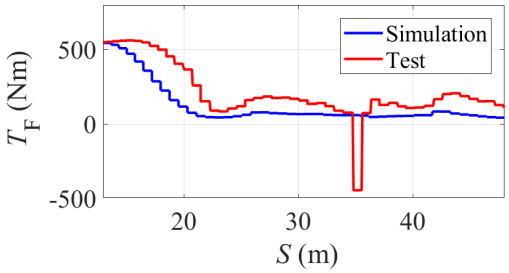
Figure 7.10: Vehicle states during double lane change with heading angle reference.



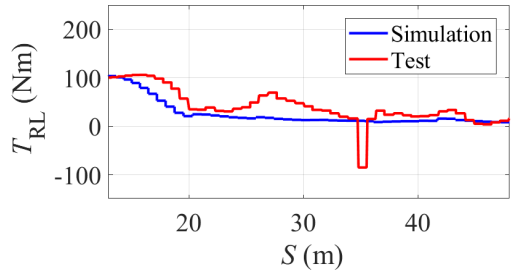
(a) Front steering commands



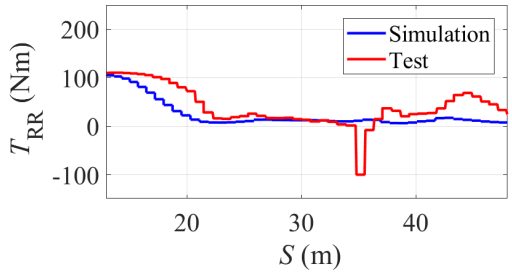
(b) Rear steering commands



(c) Front wheel torque commands.

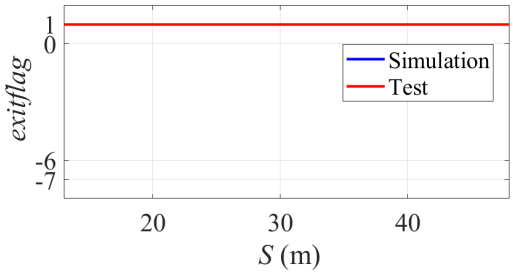


(d) Rear left wheel torque commands.

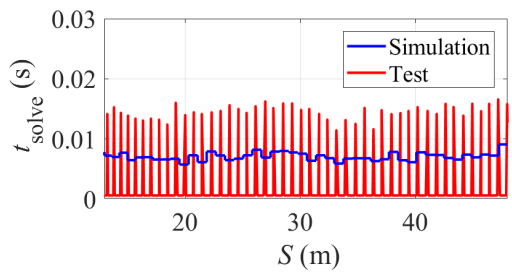


(e) Rear right wheel torque commands.

Figure 7.11: Control commands during double lane change with heading angle reference.



(a) Exitflag



(b) Computation time

Figure 7.12: Forces Pro solver information during double lane change with heading angle reference.

practical testing.

Figure 7.11a and Figure 7.11b show the steering commands during the manoeuvre. Different from the double lane change without heading reference, in this situation the controller demands equivalent front and rear steering angles in the same direction for crabbing to follow the reference path at fixed heading angle, and the steering commands are quite similar in the simulation and experiment. This demonstrates the enhancement in the vehicle's flexibility brought by the 4WS capability.

Figure 7.12a shows the exitflag information of the MPC solver. The MPC problem is solved without issue in both simulation and practical testing. The computation time is shown in Figure 7.12b.

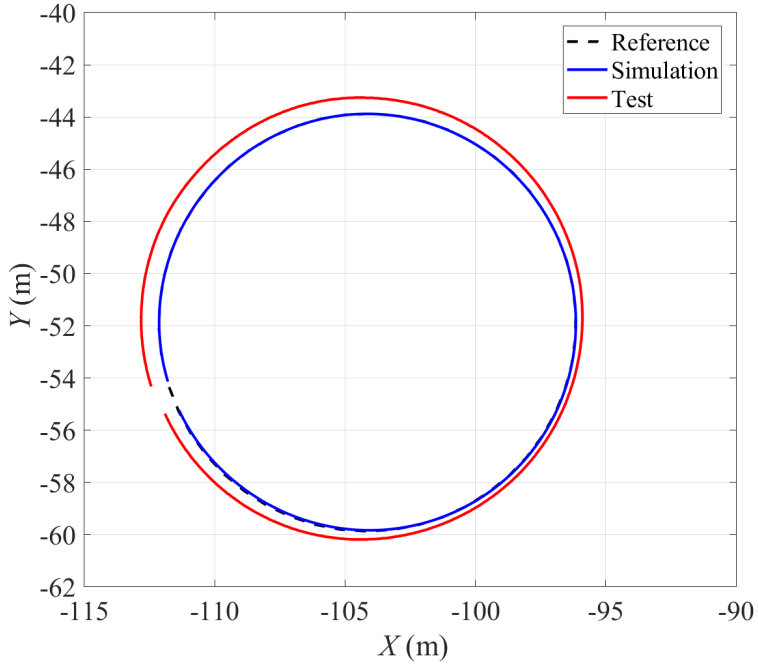
Despite not reaching the handling limits of the vehicle, the matching performance in simulation and experiments has proven the effectiveness of the developed control algorithm in the path tracking control of the autonomous vehicle. Furthermore, being able to track the double lane change path with and without heading reference has demonstrated that the application of 4WS is able to improve the flexibility of the vehicle, and that the proposed controller is able to take advantage of that to manipulate the attitude of the vehicle.

7.3.2 Constant Radius Turning Scenario

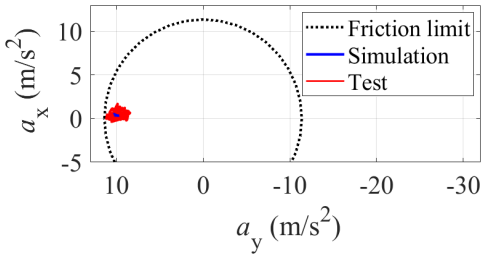
Constant Radius Turning without Side Slip Angle Reference

Following the double lane change scenario, the controller is then tested in the constant radius turning scenario. This is good for inspecting the steady-state control performance. The reference path in this scenario is a circle with the radius of 8m, which is shown in Appendix B.4. The vehicle is supposed to track the circle at a constant velocity of 9m/s.

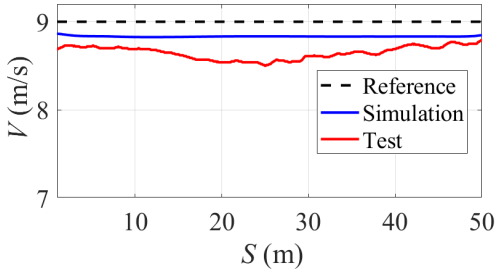
Figure 7.13a shows the trajectories of the vehicle in simulation and experiments. It can be seen that the vehicle is quite close to the reference path in simulation, and there is a



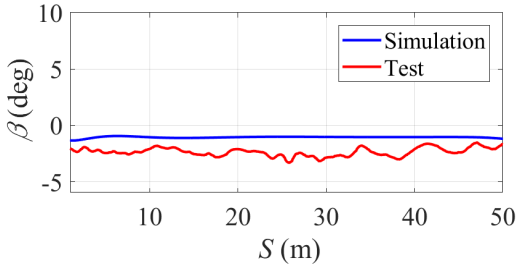
(a) Vehicle trajectory



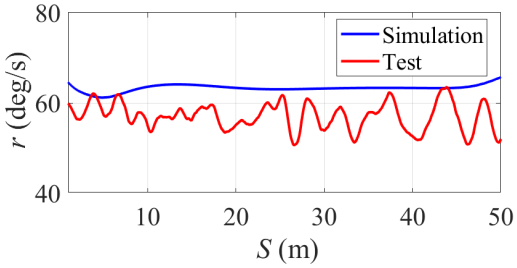
(b) g-g diagram



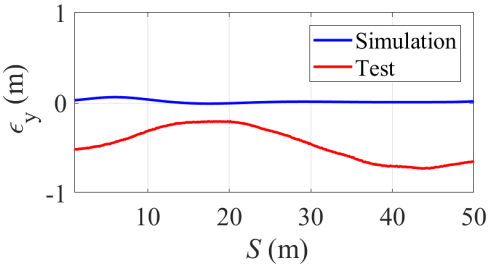
(c) Velocity



(d) Side slip angle

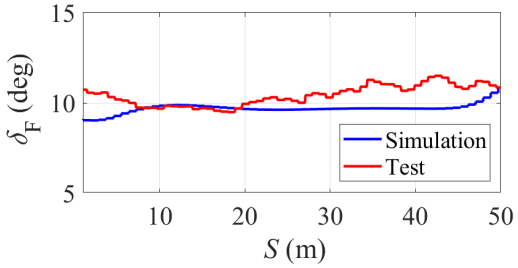


(e) Yaw rate

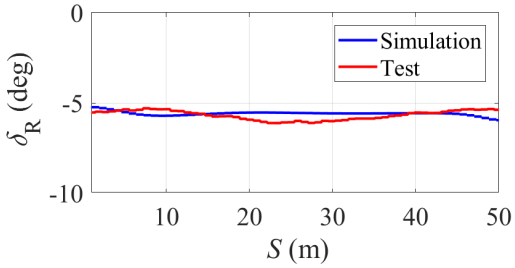


(f) Lateral tracking error

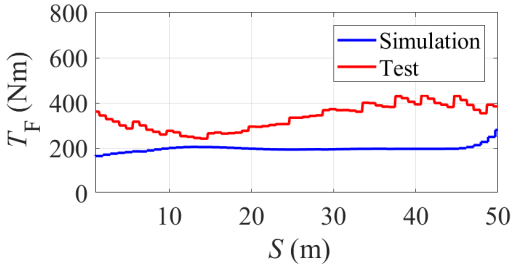
Figure 7.13: Vehicle states during constant radius turning without side slip angle reference.



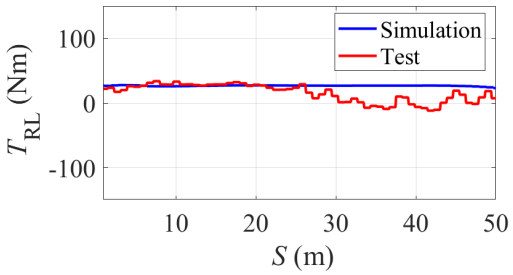
(a) Front steering commands



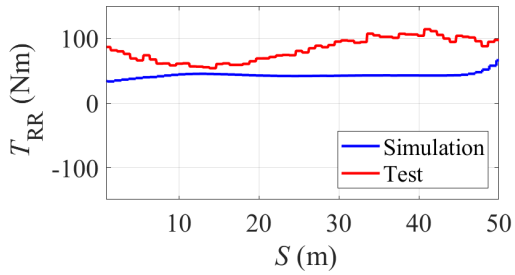
(b) Rear steering commands



(c) Front wheel torque commands.

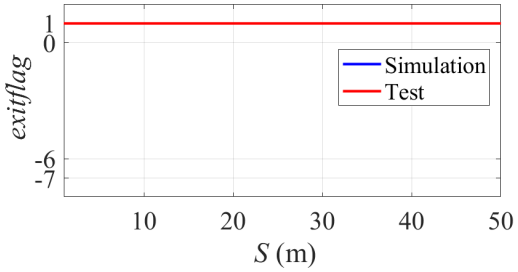


(d) Rear left wheel torque commands.

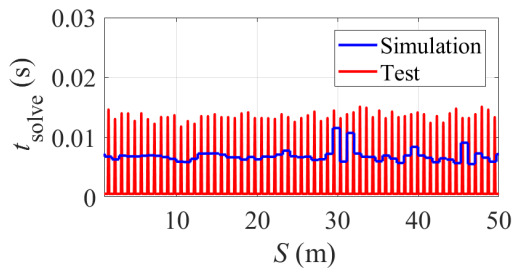


(e) Rear right wheel torque commands.

Figure 7.14: Control commands during constant radius turning without side slip angle reference.



(a) Exitflag



(b) Computation time

Figure 7.15: Forces Pro solver information during constant radius turning without side slip angle reference.

larger deviation in the experiment. The acceleration of the vehicle is displayed in Figure 7.13b, which shows that the vehicle has reached the limits of handling in both situations. As shown in Figure 7.13c, the controller tracks the reference velocity with a steady-state deviation up to 0.2m/s in simulation, and with a maximum of 0.5m/s deviation in the experiment. The side slip angle is close in simulation and experiment, while the yaw rate in experiment is lower than the simulation due to the larger actual turning radius. Figure 7.13f shows the lateral tracking error of the vehicle. The controller does a great job in tracking the reference path with steady-state deviation around 0.01m, which is impressive given the extreme condition under which the vehicle is operating. There is some drawback in the tracking performance in experiment, where the lateral tracking error is up to 0.75m.

Figure 7.14 shows the control commands during the constant radius turning manoeuvre. From Figure 7.14a and Figure 7.14b, it can be seen that the controller requests similar steering angles in simulation and experiment. Opposite front and rear steering angles are demanded to satisfy the required yaw moment. As shown in Figure 7.14c, more driving torque is applied on the front wheels in practical testing to maintain the velocity of the vehicle. Figure 7.14d and 7.14e shows the torque commands on the rear wheels. There is not much torque difference between the rear wheels in simulation, but in the experiment TV is activated on the rear axle to generate additional yaw moment to stabilise the vehicle.

Figure 7.15a shows that there is no issue with the MPC solver, and the computation time is shown in Figure 7.15b.

Constant Radius Turning with Side Slip Angle Reference

The constant radius turning manoeuvre is then extended to include additional reference of the side slip angle. The vehicle is supposed to track the same circle at the reference velocity of 9m/s, and with a side slip angle of 20°.

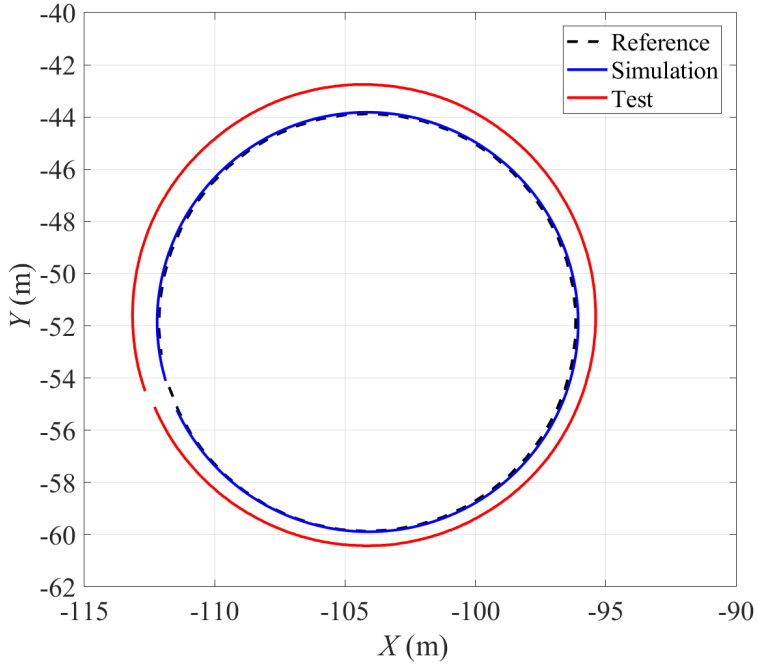
Figure 7.16 shows the vehicle states during constant radius turning with side slip refer-

ence. The acceleration of the vehicle is shown in Figure 7.16b, and it can been that the vehicle again reaches the friction limits. As shown in Figure 7.16c, the tracking of velocity is good in both simulation and experiment, with slight variation in the experiment. Figure 7.16d shows the side slip angle of the vehicle. It can be seen that the vehicle is not able to reach the 20° side slip angle reference, and the steady-state value is around 15°. The yaw rate of the vehicle is shown in Figure 7.16e, and again because of the larger turning radius, the yaw rate in practical testing is to some extent lower than that in the simulation. Figure 7.16f shows the lateral deviation of the vehicle. Similar to the situation without side slip reference, the controller has achieved great tracking performance in simulation. Despite the lateral tracking error up to 1m, the controller has shown its effectiveness in controlling the autonomous vehicle at the handling limits.

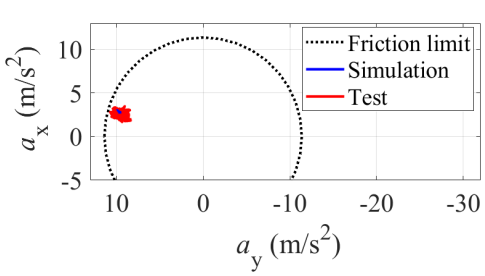
Figure 7.17a and Figure 7.17b shows the steering commands of the controller. It can be seen that the rear steering is saturated at its limitation of 19°, while a small front steering angle in the same direction is demanded by the controller in both simulation and experiment. The front and rear steering in the same direction helps to track the side slip reference, and it is the difference between them that generates the required yaw moment for turning. Figure 7.17d and Figure 7.17e shows the torque commands on the rear wheels. In both simulation and experiment TV is utilised so that additional yaw moment is generated for vehicle stabilisation, and the larger torque is requested in the practical testing.

Figure 7.18a shows the exitflag of the MPC solver, which indicates successful execution of the MPC solver during the scenario. The computation time of the control system is shown in Figure 7.18b.

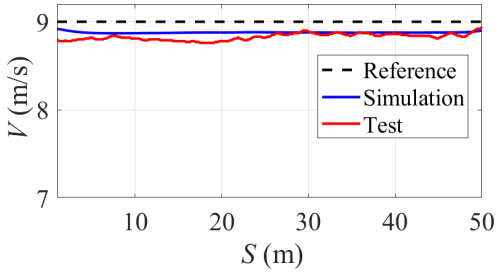
In this section, the developed controller is validated in the constant radius turning scenario. In the situation without side slip reference, the controller is able to track the reference path at the reference velocity while maintaining the vehicle stability. In addition, the controller remains effective with the presence of side slip angle reference, which demonstrates the advantage of the proposed control formulation in autonomous vehicle at the



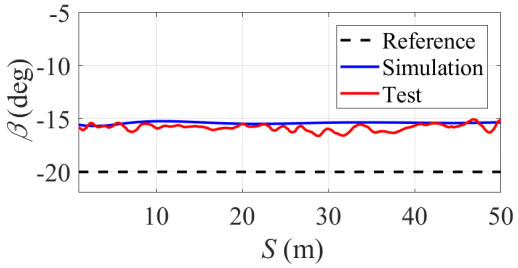
(a) Vehicle trajectory



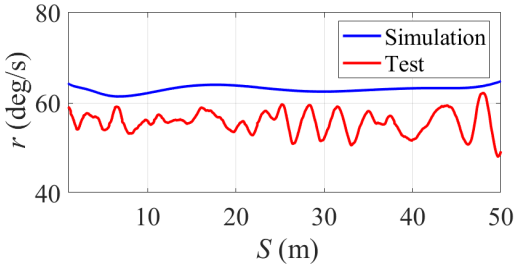
(b) g-g diagram



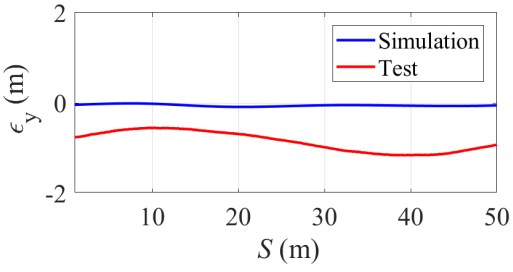
(c) Velocity



(d) Side slip angle

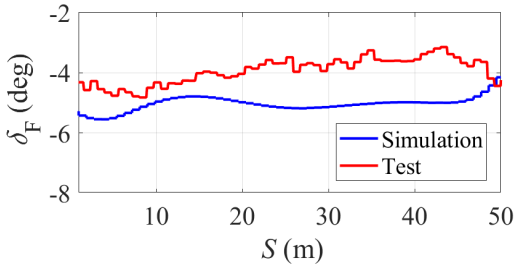


(e) Yaw rate

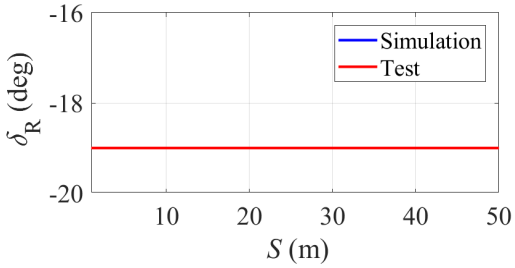


(f) Lateral tracking error

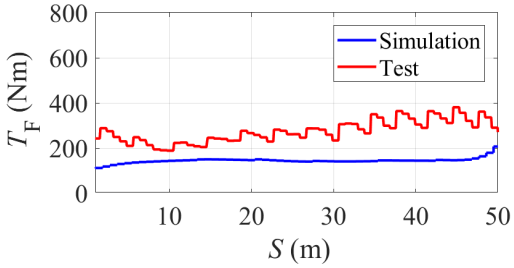
Figure 7.16: Vehicle states during constant radius turning with side slip angle reference.



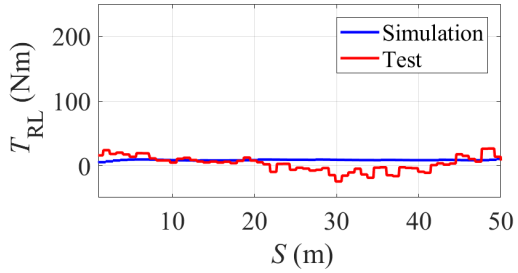
(a) Front steering commands



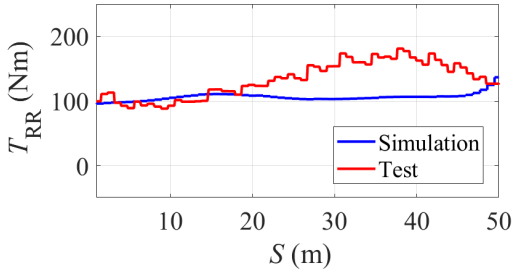
(b) Rear steering commands



(c) Front wheel torque commands.

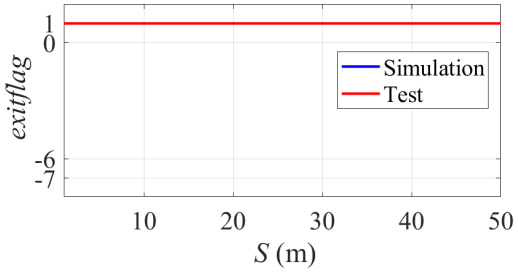


(d) Rear left wheel torque commands.

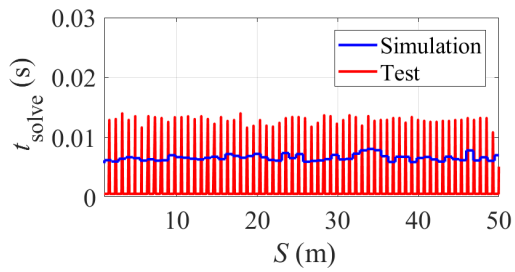


(e) Rear right wheel torque commands.

Figure 7.17: Control commands during constant radius turning with side slip angle reference.



(a) Exitflag



(b) Computation time

Figure 7.18: Forces Pro solver information during constant radius turning with side slip angle reference.

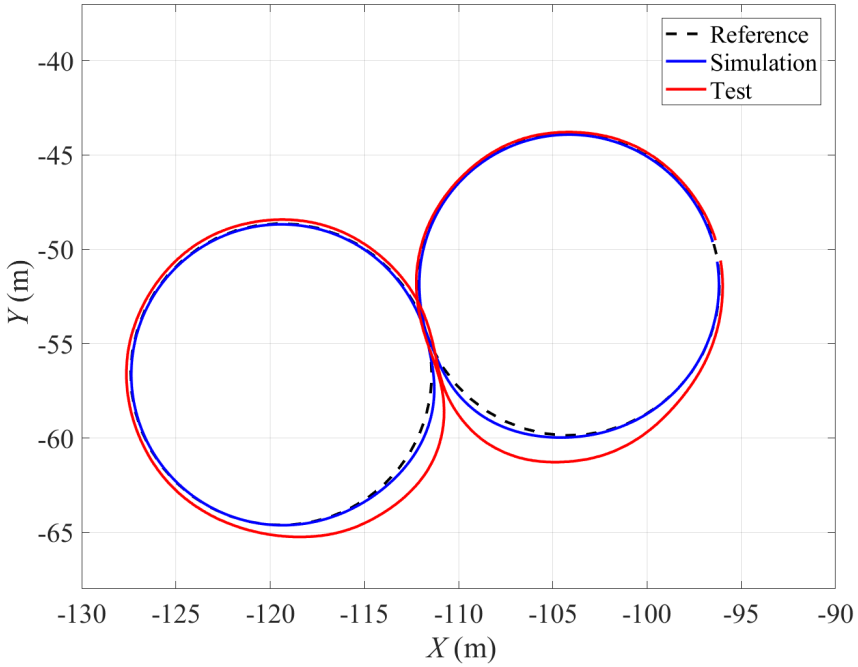
limits of handling.

7.3.3 Figure 8 Scenario

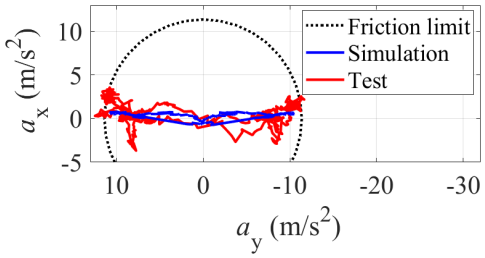
Figure 8 Tracking without Side Slip Angle Reference

Following the constant radius turning scenario, the reference path is expanded to a figure 8 track. It consists of two circles tangent to each other, and the radius of both circles are 8m. The layout of the reference path is shown in Appendix B.4. The vehicle is supposed to track two circles one after the other as a closed loop, and a reference velocity of 8m/s is given for the path tracking purpose.

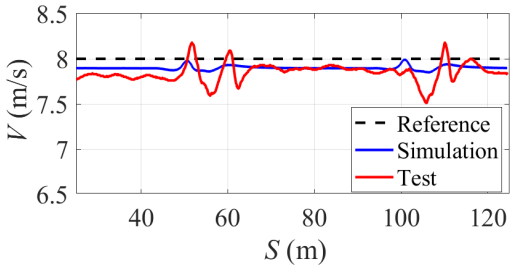
Figure 7.19 shows the vehicle states during the figure 8 tracking. Figure 7.19a shows the trajectory of the vehicle in comparison with the reference path. It can be seen that at the cross point of the two circles, the vehicle deviated from the reference path in simulation due to the quick change of turning direction, and the deviation is larger in the experiment. When it gets back to steady-state, the tracking performance is better. The g-g diagram as shown in 7.19b suggests that the vehicle is operated at its handling limits on both circles in simulation as well as in the experiment. The steady-state tracking error of the velocity is around 0.2m/s, while more variation in the velocity up to 0.5m/s takes place in the experiment at the cross point, where the vehicle switches from one steady-state condition to another within 3 seconds. Figure 7.19d shows the side slip angle of the vehicle. It can be seen that the side slip angle remains small during the steady-state condition, while spikes take place at the cross point in practical testing. The yaw rate of the vehicle is shown in Figure 7.19e. Despite the oscillation in experiment compared with simulation, the controller has done a good job in stabilising the vehicle in such an extreme condition. With the presence of the dramatic transition between two steady-state conditions, the controller has managed to stabilise the yaw rate efficiently. The lateral tracking error is shown in Figure 7.19f. The controller is able to track the figure 8 path with a maximum



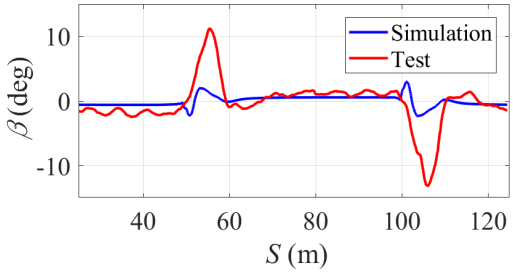
(a) Vehicle trajectory



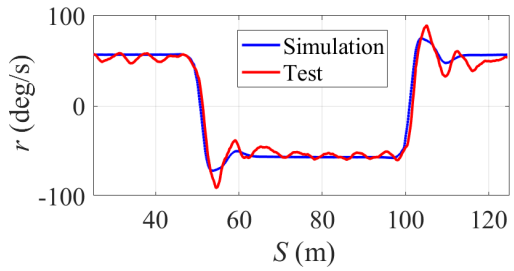
(b) g-g diagram



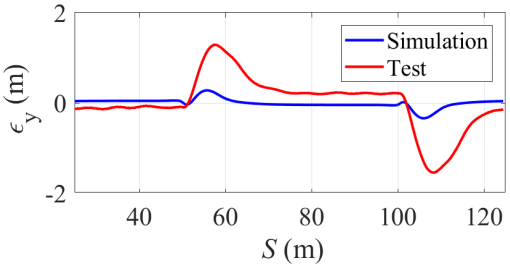
(c) Velocity



(d) Side slip angle

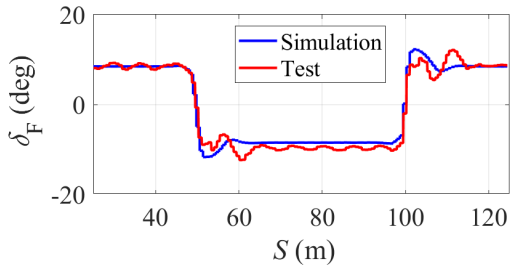


(e) Yaw rate

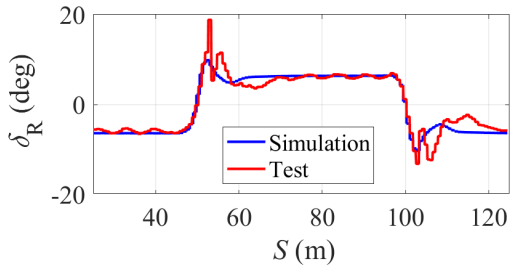


(f) Lateral tracking error

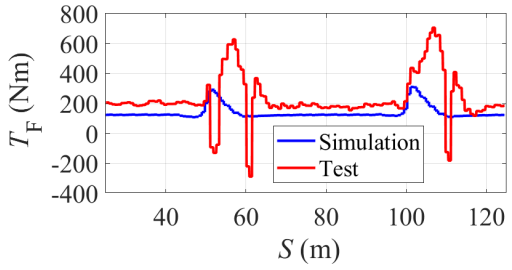
Figure 7.19: Vehicle states during figure 8 tracking without side slip angle reference.



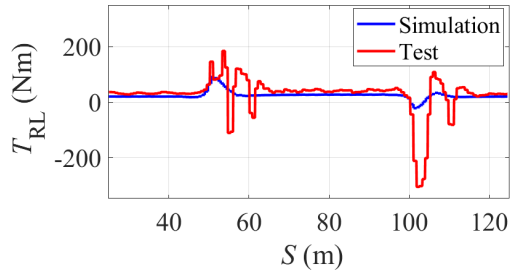
(a) Front steering commands



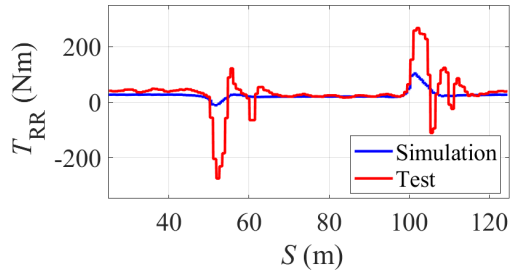
(b) Rear steering commands



(c) Front wheel torque commands.

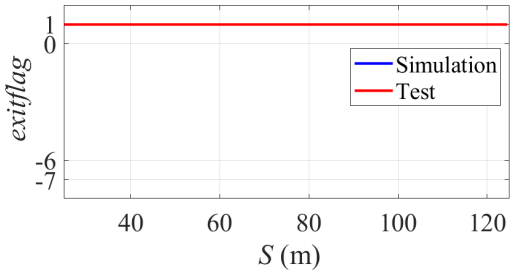


(d) Rear left wheel torque commands.

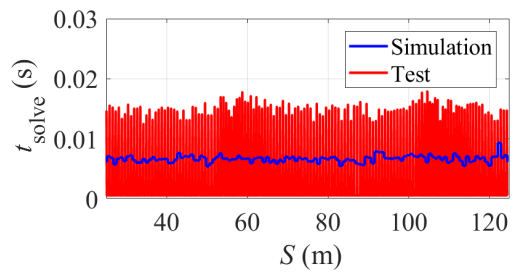


(e) Rear right wheel torque commands.

Figure 7.20: Control commands during figure 8 tracking without side slip angle reference.



(a) Exitflag



(b) Computation time

Figure 7.21: Forces Pro solver information during figure 8 tracking without side slip angle reference.

deviation of 0.35m in simulation, while the tracking error is up to 1.6m in the experiment. The steady-state tracking error is around 0.2m in practical testing, and is close to zero in simulation.

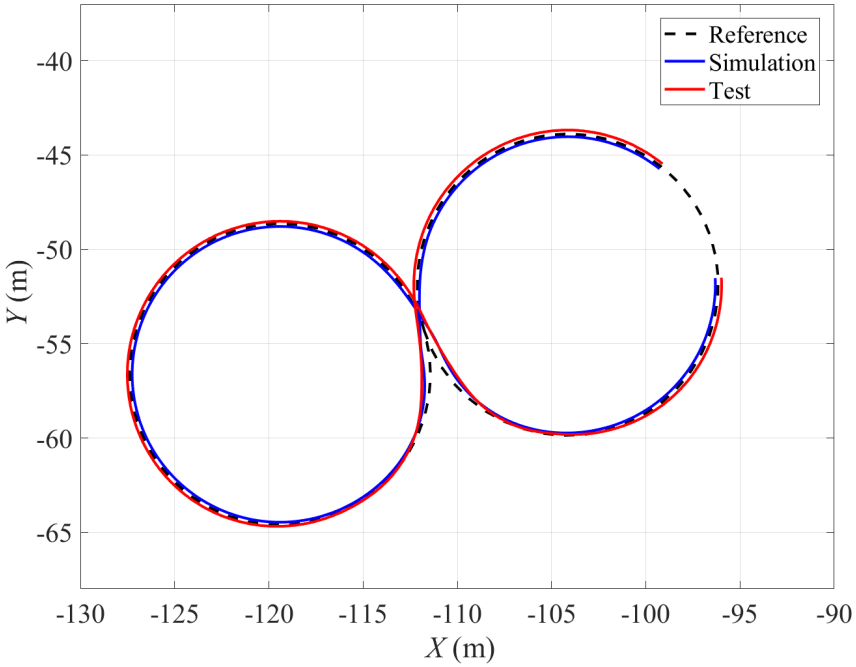
Figure 7.20 shows the control commands during the figure 8 tracking. From Figure 7.20a and Figure 7.20b it can be seen that opposite front and rear steering is applied by the controller during the steady-state turning. The front wheel torque command is shown in Figure 7.20c. During transition, larger torque on the front wheels is required in order to compensate the loss in longitudinal velocity due to the harsh change of lateral motion. Figure 7.20d and Figure 7.20e show the rear wheel torque commands. Differential torque on the rear axle can be observed in simulation which generates the yaw moment to stabilise the vehicle, and the difference is greater in the practical testing.

Figure 7.21a shows the exitflag of the MPC solver. It can be seen that optimal solution is consistently obtained by the controller regardless of the vehicle condition. The computation time of the control system is shown in Figure 7.21b.

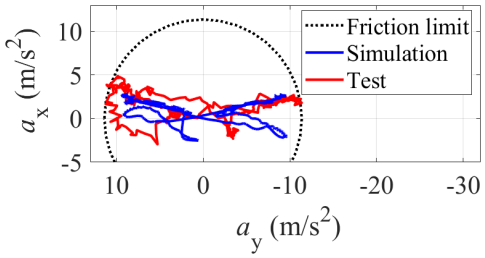
Figure 8 Tracking with Side Slip Angle Reference

In addition to the figure 8 tracking, additional reference of side slip angle is included in the control formulation. In detail, a reference side slip angle of 15° is given to the controller, and the vehicle is still supposed to track the figure 8 path at the constant velocity of 8m/s.

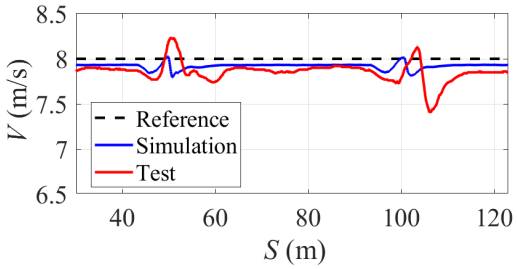
Figure 7.22 shows the vehicle states in simulation and experiment. It can be seen in Figure 7.22a that the vehicle tends to cut the corner at the transition point to avoid going wide from the reference path, and thus good tracking performance has been achieved by the controller in both simulation and experiments. Figure 7.22b shows the acceleration of the vehicle, which confirms the operation of the vehicle at the limits of handling. As shown in Figure 7.22c, variation in velocity takes place at the transition point, with the tracking error up to 0.2m/s in simulation and up to 0.6m/s in the experiment. Figure 7.22d shows



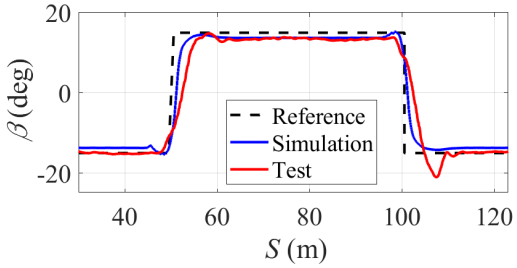
(a) Vehicle trajectory



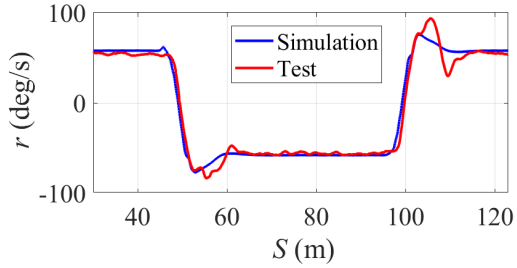
(b) g-g diagram



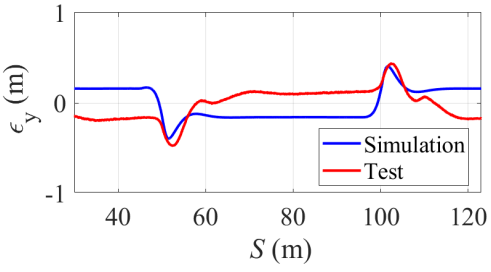
(c) Velocity



(d) Side slip angle

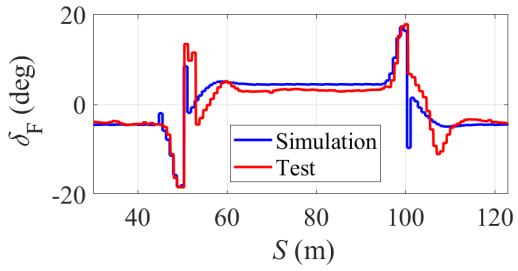


(e) Yaw rate

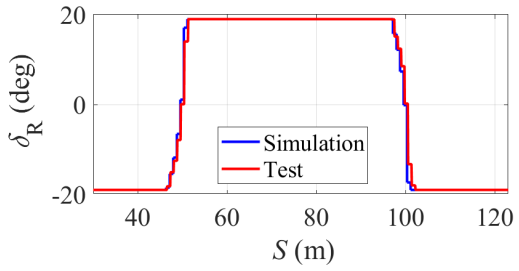


(f) Lateral tracking error

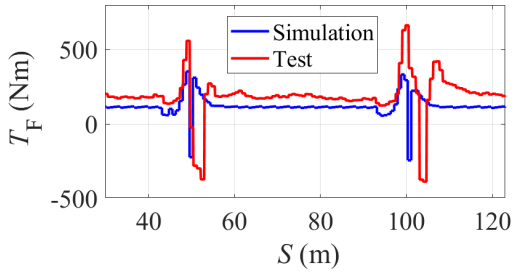
Figure 7.22: Vehicle states during figure 8 tracking with side slip angle reference.



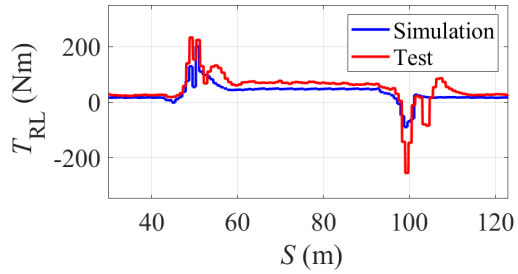
(a) Front steering commands



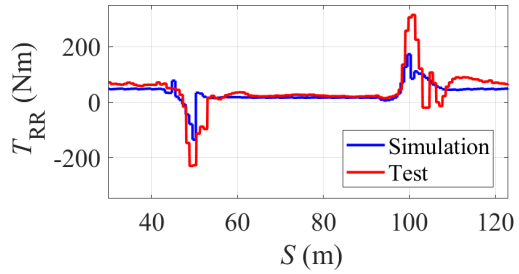
(b) Rear steering commands



(c) Front wheel torque commands.

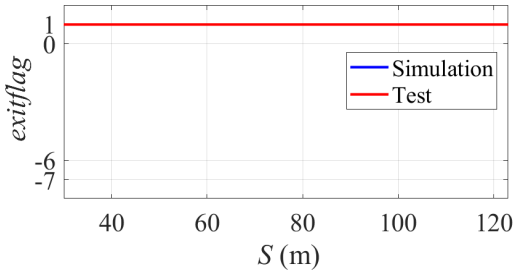


(d) Rear left wheel torque commands.

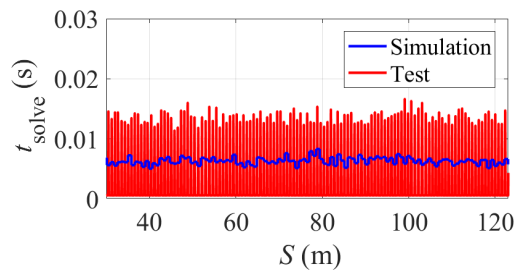


(e) Rear right wheel torque commands.

Figure 7.23: Control commands during figure 8 tracking with side slip angle reference.



(a) Exitflag



(b) Computation time

Figure 7.24: Forces Pro solver information during figure 8 tracking without side slip angle reference.

the side slip angle of the vehicle compared to the reference. The controller is able to track the side slip reference with the steady-state error less than 2 degrees. Figure 7.22e shows the yaw rate of the vehicle, and it can be seen that the controller is effective in stabilising the vehicle in both simulation and experiment during the transition. In the practical testing, there is larger overshoot in the yaw rate at the transition point, but the controller still manages to stabilise it in about 2 seconds. Figure 7.22f shows the lateral tracking error of the vehicle. The lateral deviation is maintained within 0.5m in both simulation and experiment, while the steady-state error is slightly larger than the figure 8 tracking without side slip angle reference.

Figure 7.23a and Figure 7.23b shows the steering commands of the controller during operation. Similar with the constant radius turning scenario with side slip reference, the rear steering is saturated at the constraint, and the front steering tends to be in the same direction with a smaller value. Figure 7.23d and Figure 7.23e shows the rear wheel torque commands, and it can be seen that again large torque differential on the rear wheels takes place at the transition point to compensate the required yaw moment to change the direction of the vehicle promptly while maintaining vehicle stabilisation.

Figure 7.24a shows the exitflag of the MPC solver, and the computation time of the control system is shown in Figure 7.24b.

As one step further from the constant radius turning scenario, the figure 8 tracking introduces significant transition between steady-state conditions, which raises high requirement on the effectiveness and robustness of the controller. According to the results in both simulation and practical testing, the proposed controller has shown its capability in autonomous vehicle control at the limits of handling. Moreover, the figure 8 tracking with side slip reference demonstrates the capability of the proposed controller in manipulating the vehicle's attitude relative to the reference path.

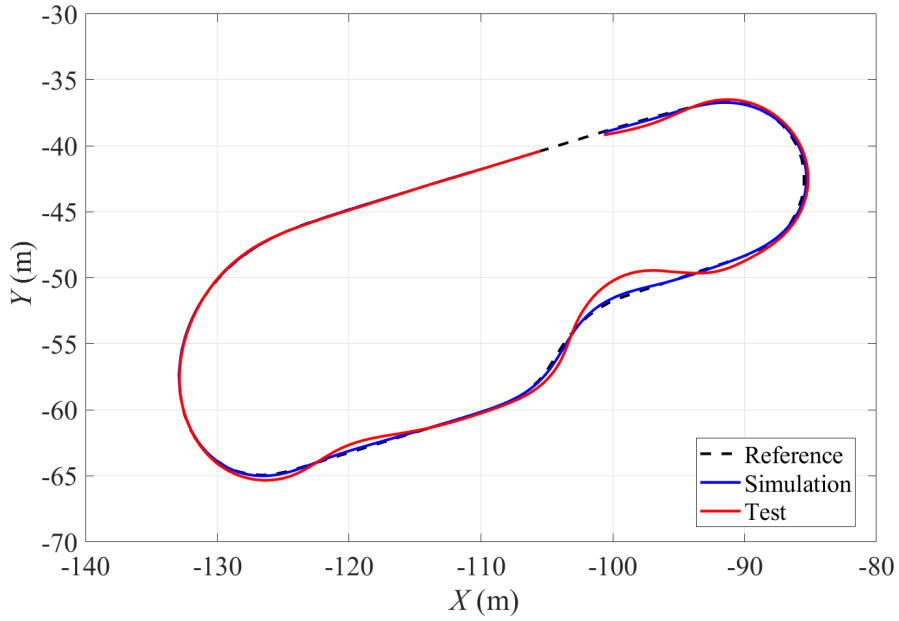
7.3.4 General Path Tracking Scenario

Finally, the developed controller is tested in a general path tracking scenario. The reference path consists of straights, turning and lane change sections, and its layout is shown in Appendix B.5. The vehicle is supposed to track this path at a reference velocity of 7.5m/s.

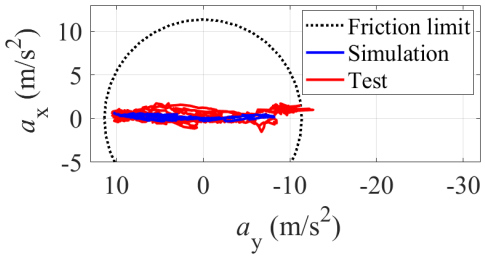
Figure 7.25a shows the trajectory of the vehicle in both simulation and experiment. The controller is able to track the reference path, with a relatively large deviation taking place at the exit of the lane change. Figure 7.25b shows the acceleration of the vehicle during the lap, and it can be seen that the friction limit is reached. Figure 7.25c shows the vehicle's velocity against reference. The velocity tracking is better in the simulation, and there is a maximum deviation of 0.5m/s in the practical testing. Figure 7.25f shows the lateral tracking error of the vehicle. The lateral deviation is less than 0.2m in simulation, and is up to 1.5m at the exit of the lane change in experiment.

Figure 7.26 shows the control commands of the controller. It can be seen that similar control commands are demanded by the controller in simulation and experiment, while there is some fluctuation in the commands in the experiment, which probably leads to the lateral deviation at the exit of land change.

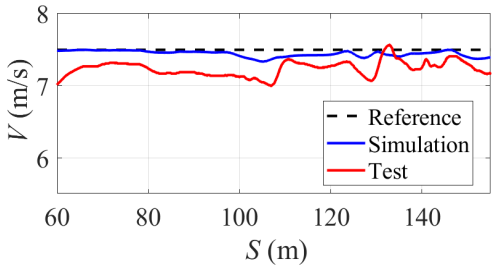
Figure 7.27a shows the exitflag of the MPC solver. Several abnormal exitflag of 0 can be noticed during the experiment, indicating that the solver fails to find the optimal solution within the iteration limits. This happens at the exit of the lane change, and is probably the reason to the large lateral deviation during the experiment. The computation time of the control system is shown in Figure 7.27b.



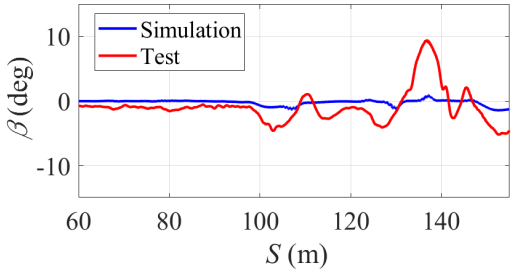
(a) Vehicle trajectory



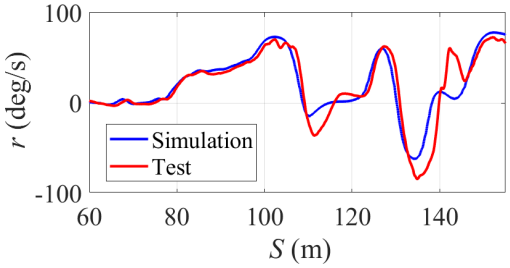
(b) g-g diagram



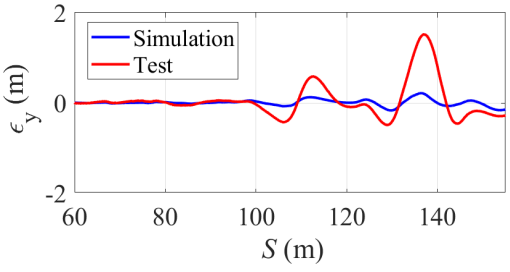
(c) Velocity



(d) Side slip angle

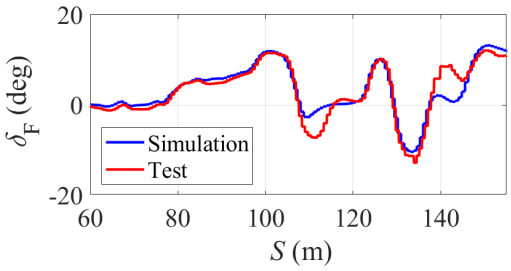


(e) Yaw rate

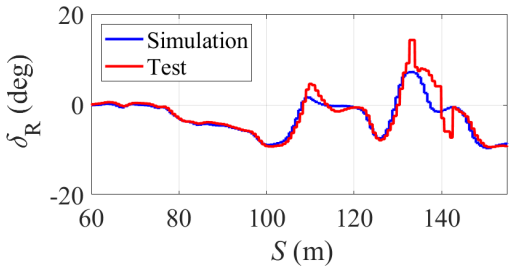


(f) Lateral tracking error

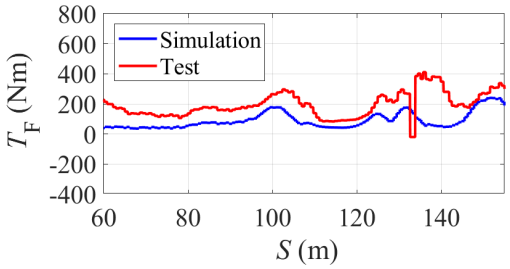
Figure 7.25: Vehicle states during general path tracking.



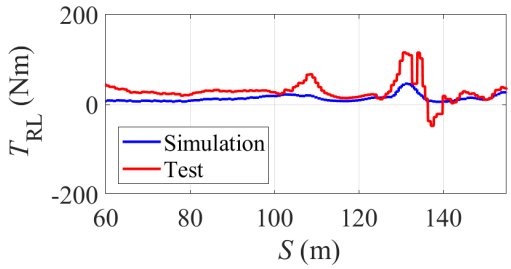
(a) Front steering commands



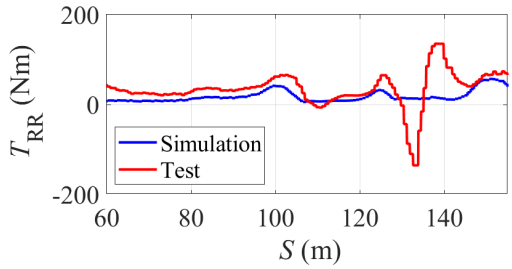
(b) Rear steering commands



(c) Front wheel torque commands.

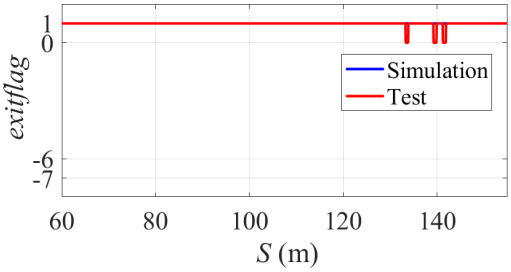


(d) Rear left wheel torque commands.

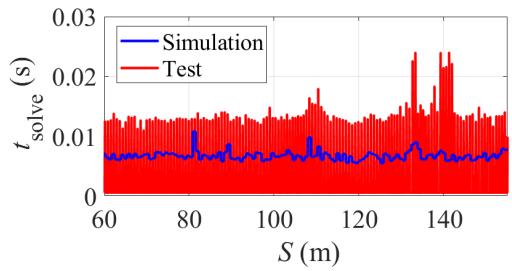


(e) Rear right wheel torque commands.

Figure 7.26: Control commands during general path tracking.



(a) Exitflag



(b) Computation time

Figure 7.27: Forces Pro solver information during general path tracking.

7.4 Summary

In this chapter, the developed controller is implemented on a real vehicle for practical testing. First of all, a complete control algorithm is built up based on the core MPC algorithm, including signal transmission, data validation and safety controller. The entire system is then implemented on the real time target machine which is installed on the testing vehicle.

The controller is first tested in a double lane change scenario. Due to the limitation in testing area, the vehicle couldn't reach the high velocity for limits of handling operation. Despite that, the double lane change testing validates the effectiveness of the proposed controller in autonomous vehicle control for path tracking. In addition, the double lane change with fixed heading reference demonstrates the advantage of the 4WS in enhancing the flexibility of the vehicle.

The controller is then tested in the constant radius turning scenario. The controller is able to track the reference path at the reference velocity, and the g-g diagram confirms that the friction limit is reached, indicating operation at the limits of handling. Moreover, it is also tested to included side slip angle reference in the control formulation. In the constant radius turning scenario, the controller is validated for path tracking and stabilisation of the autonomous vehicles at the steady-state condition at the limits of handling.

Following the constant radius turning, the controller is tested in the figure 8 tracking scenario. As an extension to the circle path, the vehicle is supposed to track two tangent circles at a reference speed of 8m/s. This introduces a significant transition in the path tracking process, which has a high requirement on the robustness of the controller in terms of pathh tracking performance and vehicle stabilisation. Furthermore, the controller is also tested with the presence of side slip reference. The controller manages to track both velocity and side slip reference with relatively small deviation. In terms of the lateral tracking error, the maximal deviation is as small as 0.5m. The figure 8 tracking testing

further validates the control performance of the proposed controller. It is shown that the controller is able to handle challenging transition during extreme conditions, which confirms the effectiveness and robustness of the proposed controller, and also demonstrates the advantage of the proposed over actuation formulation for autonomous vehicle control.

The controller is finally tested in a general path tracking scenario, which combines straights, turning and lane change manoeuvres. The controller is able to track the reference path at the reference velocity, with a relatively large lateral deviation at the exit of the lane change probably due to the failure of the MPC solver in finding the optimal solutions. This may be improved by some fine tuning pf the MPC parameters.

In summary, the developed controller has been validated in the practical testing, in terms of path tracking control at the limits of handling. It demonstrates the advantage of the proposed over actuation formulation in path tracking control ad stabilisation of autonomous vehicles. The application of TV on the rear axle enhances the stability of the vehicle during extreme operation conditions, and 4WS helps to improve the flexibility of the vehicle, which can be used to manipulate the vehicle attitude during operation at the limits of handling.

Chapter 8

Conclusions

8.1 Summary of Contributions

This thesis has developed a control algorithm for an over actuated electric vehicle. The controller is capable of path tracking at the limits of handling, by exploiting the full vehicle dynamical capability with 4WS and TV functionalities. The control design is based on MPC, and the system dynamics for MPC formulation is based on a 3dof vehicle model taking into account the non-linearities in vehicle dynamics and tyre force, which is critical for vehicle control at the limits of handling. The MPC controller is formulated based on this vehicle model, together with the vehicle state reference generated according to a simplified bicycle model. The basic idea of the controller is to maintain the vehicle states as close to the reference as possible, while respecting the constraints in the actuation.

Two path tracking model are first proposed and compared in simulation. In the double lane change scenario, the CTSNF controller based on Cartesian coordinate system is able to reach a higher speed compared to the CVLRF controller based on curvilinear coordinate system. In the double U-turn scenario, the CTSNF controller does a better job in stabilising the vehicle and maintaining smaller tracking error upon the sharp change of

turning direction. Furthermore, the CTSNF controller takes less computation time than the CVLRF controller. As a result, the CTSNF formulation is preferred and thus applied in the further control development. The effect of the application of steady-state reference on the control performance is also studied. By introducing the steady-state reference, it makes the solver easier to find out the optimal solution, which reduces the oscillation in control commands and improves the vehicle stability.

The effects of multi-actuation on the control performance is then investigated. In addition to the CTSNF controller applying both 4WS and TV (4WS-TV), three additional control formulations are proposed with different level of actuation, including FWS only (FWS), 4WS and FWS with TV (FWS-TV). The control performance of these four controllers are compared in the simulation. In the double U-turn scenario, the FWS controller achieves the worst tracking performance in terms of velocity tracking and lateral tracking error, which is due to the lack of multi actuation. In comparison, the introduction of either rear steering or TV can significantly improve the tracking performance as well as vehicle stability. In particular, TV seems to be more effective than 4WS in enhancing vehicle response, because it is able to manipulate the individual driving torque on the rear wheels. Among the four controllers, the 4WS-TV controller achieves the best performance with smallest lateral tracking error, while at the cost of the highest computation time. In the double lane change scenario, the 4WS-TV is the most outstanding one as well that reaches the highest velocity within the constrained lateral deviation on exit. The 4WS controller reaches a higher velocity than the FWS-TV controller, and the FWS controller again provides the worst performance among the four formulation. It can be concluded that the application of 4WS and TV can enhance the vehicle's response at the limits of handling, and by combining these two, the produced over-actuation formulation helps to achieve the best control performance. However, this increases the computation time of the system, which is an disadvantage to the implementation of the developed controller in real time.

Several modification has been done in order to reduce the computation time. First, the

sampling time is increased from 0.02s to 0.1s, while 5 additional integration steps is carried out within the sampling time. By doing so, the frequency of control actions is reduced, while the accuracy of system state estimation is retained to some extent. By comparing the control performance in the double U-turn scenario, the updated sampling time setup has achieved a similar control performance as the previous controller, but the computation time is significantly reduced. Then, the control performance is investigated with different prediction horizon time. Prediction horizon is critical to MPC as it determines the time range within which the system states are taken into account in the optimisation problem. According to the simulation, it is shown that a generally a longer prediction horizon leads to a better path tracking performance. A prediction horizon of 0.6s is not able to provide a prompt response to the required change of turning direction, while with the 1.4s horizon the vehicle is able to maintain the lateral tracking error within 0.2m, and keep the vehicle velocity closest to the reference. However, a longer prediction horizon takes more time for computation, which has a negative influence on the real time execution. In the end, the control performance is compared with different sampling time, and it is shown that the best control performance is achieved with the shortest sampling time, but at the cost of the longest computation time. In overall, the combination of sampling time of 0.1s and prediction horizon of 1s provides the best compromise between good path tracking performance and short computation time, hence it is most suitable for the proposed controller for real time operation.

Finally, the proposed controller is implemented in real time for practical testing. It is first tested in the double lane change scenario. The controller is proven to be effective in the path tracking control of autonomous vehicles. In addition, it is also tested the carry out the double lane change with a fixed heading angle reference. This demonstrates the flexibility in vehicle attitude control provided by the application of 4WS. The controller is then tested in constant radius turning scenario with and without side slip reference. And the results shows that the proposed controller is able to carry out path tracking and stability control at the limits of handling. The controller is then tested in the figure 8 tracking

scenario, which includes significant transition between steady conditions. The turning direction changes sharply at the transition point, which is a challenge for the controller to stabilise the vehicle. The testing results show that the controller is capable of tracking the reference trajectory while maintaining the vehicle stabilisation. This demonstrates the good robustness of the proposed controller. In the end the controller is tested in a general path tracking scenario. The controller is able to negotiate the corner at the vehicle's handling limits, while keeping close to the reference velocity. The lateral tracking error is up to 1.5m at the exit of lane change, which could be caused by the computation issue of the MPC solver, and may be improved by fine tuning of the MPC parameters.

In summary, the control algorithm developed in this thesis has been proven effective in enhancing the vehicle stability as well as performance at the limits of handling. It makes the use of 4WS and TV to maximise the dynamical capability and flexibility of the vehicle. What's more, the controller is real time implementable, and has been validated in practical testing. The controller has shown itself widely suitable for different scenarios and control objectives, with consistent control performance and good robustness.

8.2 Future Work

The work presented in this thesis can be extended in the following areas:

- Fine-tuning of the weighting in the MPC formulation for different scenarios. It cannot be guaranteed that the optimal tracking performance has been achieved in all the scenarios presented in this thesis. It is possible that by further tuning of the MPC weighting, the tracking performance can be improved.

- Investigate the impact of decoupling longitudinal and lateral tyre force on the path tracking performance. Try to figure out a more appropriate approach to reduce the computation time in prantice without decoupling the tyre force.

- The constant velocity reference can be replaced with a speed profile so that the vehicle will be able to keep operating at the handling limits throughout the entire general path. This will help to examine the controller with the requirement of combined longitudinal and lateral control.

- Including the function of path planning into the control structure. Given the low computation time for the proposed controller, it is possible to include real time path planner so that the reference trajectory of the vehicle is generated online according to conditions and vehicle dynamics.

Appendix A

Vehicle and Tyre Parameters

A list of the parameters of the vehicle and tyre in this thesis is shown in Table.

Table A.1: System Configuration of the Controllers

Parameter	Value	Parameter	Value
m (kg)	874.5	B	9.50
I_z (kgm^2)	1597.7	C	1.63
l (m)	1.995	D	1.16
l_f (m)	0.815	C_F	91393.39
l_r (m)	1.180	C_R	63123.40
w (m)	1.530	$\delta_{F,lim}$ (deg)	19
w_l (m)	0.765	$\delta_{R,lim}$ (deg)	19
w_r (m)	0.765	$T_{F,lim}$ (Nm)	800
h (m)	0.297	$T_{R,lim}$ (Nm)	350

Appendix B

Reference Path Layout

B.1 Double U-Turn Scenario

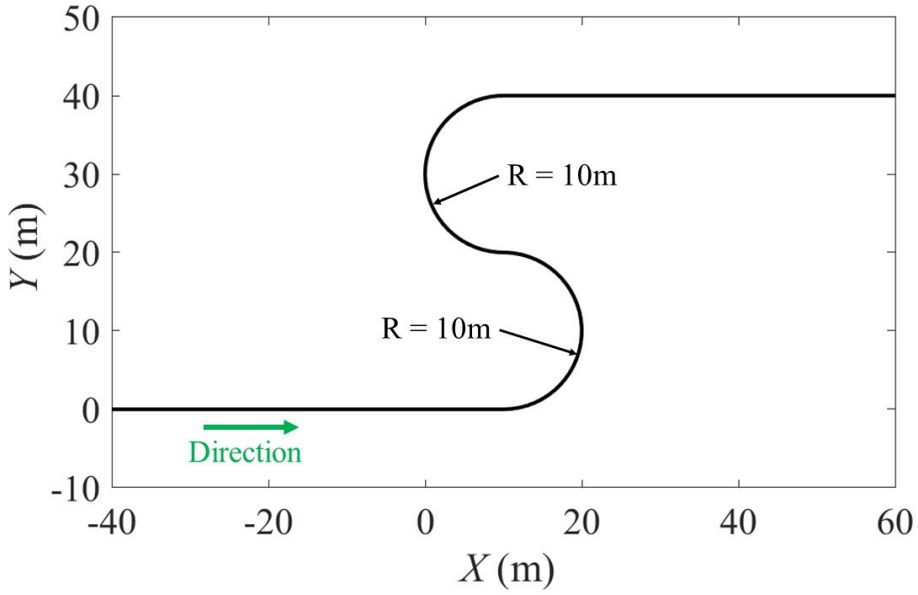


Figure B.1: Specification of the double U-turn scenario. Two half circles with the radius of 10m are connected directly, with straights before and after the path.

B.2 Double Lane Change Scenario in Simulation

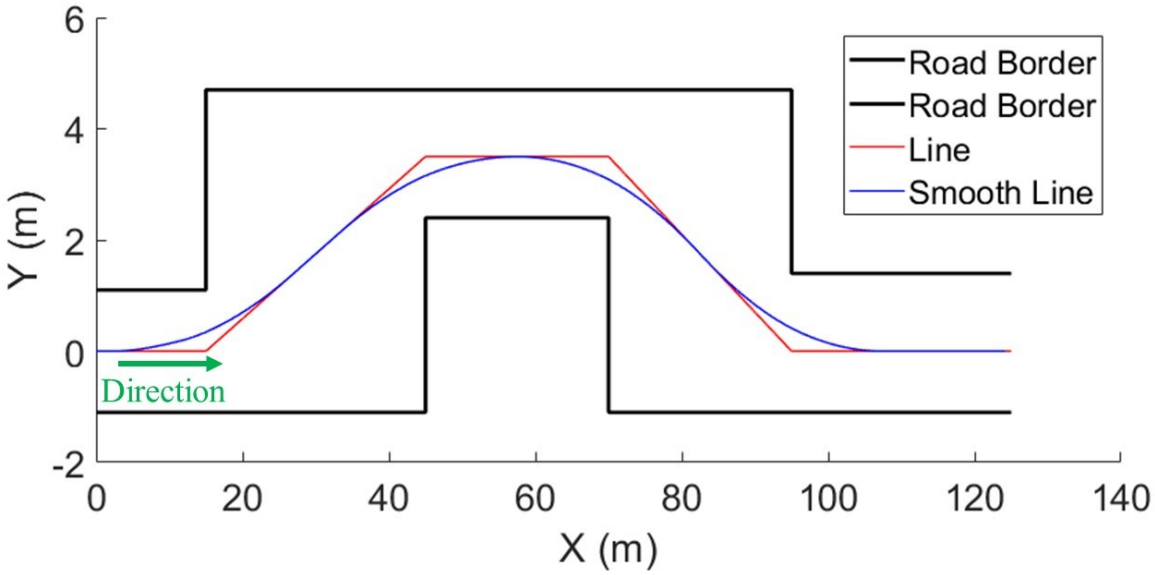


Figure B.2: Specification of the double lane change track defined by ISO 3888-1 2018. The red line is a general path based on the track with unsmooth curvature. In order to generate the reference path, a moving average filter is applied on the Y position along X axis.

B.3 Double Lane Change Scenario in Testing

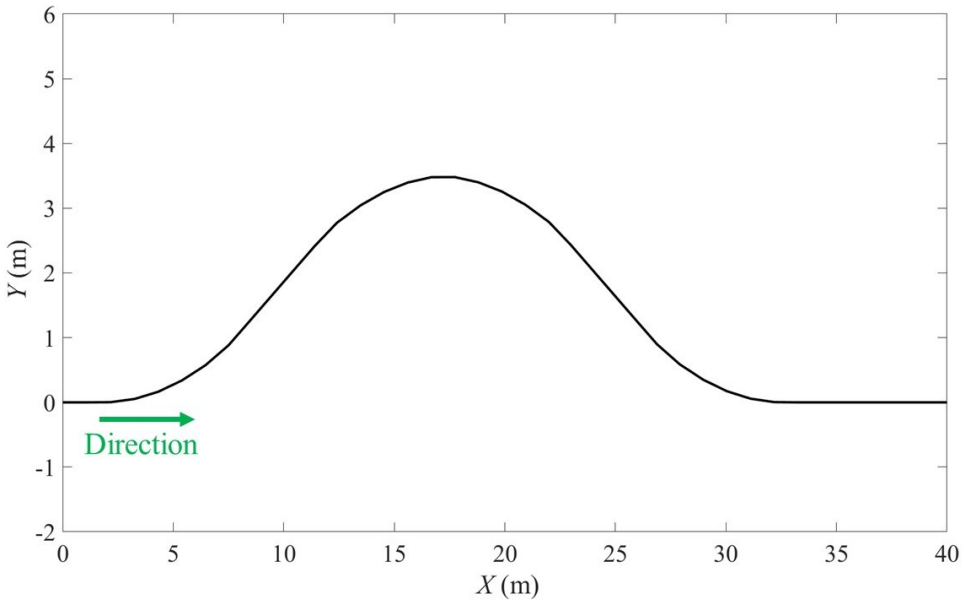


Figure B.3: Specification of the double lane change track used in the practical testing. Due to the limitation of the testing area, the length of the path is shortened.

B.4 Constant Radius Turning and Figure 8 Scenario

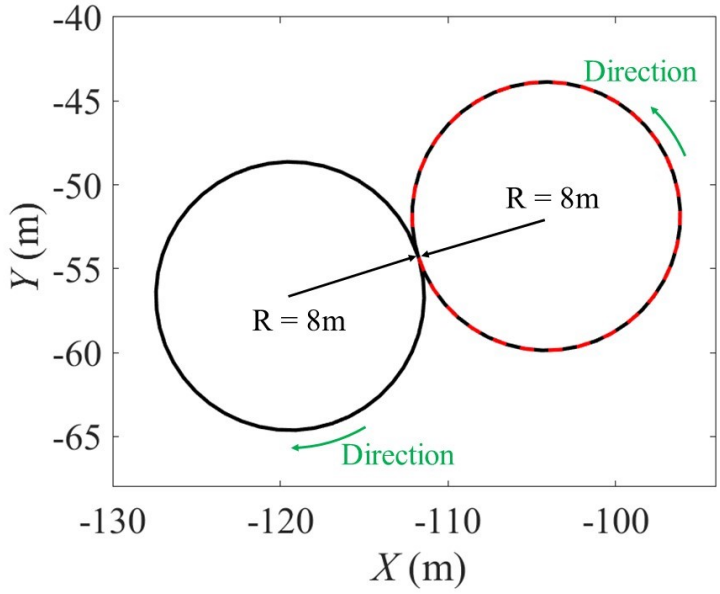


Figure B.4: Specification of the constant radius turning and figure 8 tracking scenario. The red dash line is a circle with the radius of 8m for constant radius turning, and is also part of the figure 8 path. Another circle of 8m radius is attached to it to form the complete .

B.5 General Path Tracking Scenario

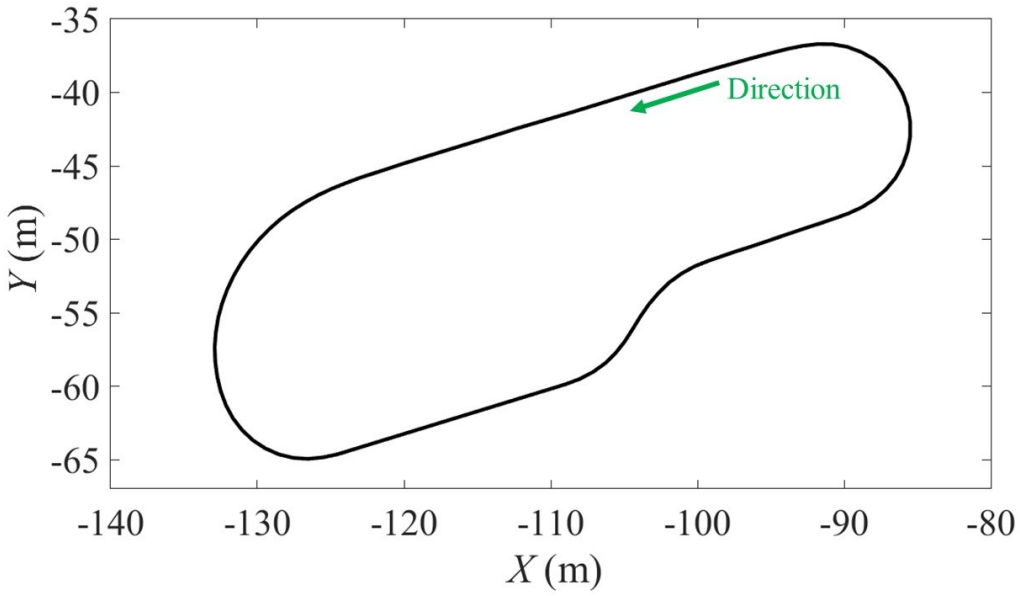


Figure B.5: Specification of the double lane change track defined by ISO 3888-1 2018. The red line is a general path based on the track with unsmooth curvature. In order to generate the reference path, a moving average filter is applied on the Y position against X.

Appendix C

MPC Parameters

Here shows the weighting factors used in the MPC controllers of this thesis.

In Chapter 4

CTSNF:

$$\begin{aligned} Q &= diag(820.70, 820.70, 0.000117, 0.000612, 0.000612) \\ R &= diag(50.00, 50.00, 131.31, 300.00, 300.00, 328.28) \end{aligned} \tag{C.1}$$

CVLRF:

$$\begin{aligned} Q &= diag(820.70, 820.70, 0.000117, 0.000612, 0.000612) \\ R &= diag(50.00, 50.00, 131.31, 300.00, 300.00, 328.28) \end{aligned} \tag{C.2}$$

In Chapter 5

FWS:

$$\begin{aligned} Q &= diag(820.70, 0.000612) \\ R &= diag(50.00, 50.00, 131.31, 300.00, 300.00, 328.28) \end{aligned} \tag{C.3}$$

4WS:

$$\begin{aligned} Q &= diag(820.70, 820.70, 0.000612) \\ R &= diag(50.00, 50.00, 131.31, 300.00, 300.00, 328.28) \end{aligned} \tag{C.4}$$

FWS-TV:

$$\begin{aligned} Q &= diag(820.70, 0.000117, 0.000612, 0.000612) \\ R &= diag(50.00, 50.00, 131.31, 300.00, 300.00, 328.28) \end{aligned} \tag{C.5}$$

4WS-TV:

$$\begin{aligned} Q &= diag(820.70, 820.70, 0.000117, 0.000612, 0.000612) \\ R &= diag(50.00, 50.00, 131.31, 300.00, 300.00, 328.28) \end{aligned} \tag{C.6}$$

In Chapter 6

$$\begin{aligned} Q &= diag(820.70, 820.70, 0.000117, 0.000612, 0.000612) \\ R &= diag(50.00, 50.00, 131.31, 300.00, 300.00, 328.28) \end{aligned} \tag{C.7}$$

In Chapter 7

Constant Radius Turning and Figure 8 Tracking:

$$\begin{aligned} Q &= diag(2462.10, 2462.10, 0.000117, 0.000612, 0.000612) \\ R &= diag(50.00, 50.00, 131.31, 100.00, 100.00, 328.28) \end{aligned} \tag{C.8}$$

Double Lane Change and General Path Tracking:

$$\begin{aligned} Q &= diag(2462.10, 2462.10, 0.000117, 0.000612, 0.000612) \\ R &= diag(50.00, 50.00, 131.31, 500.00, 500.00, 328.28) \end{aligned} \tag{C.9}$$

Appendix D

Hardware Specification

Figure D.1 shows the hardware specifications of the two hardware devices used for control algorithm execution.

Table D.1: Hardware Specification

	Workstation Laptop	Speedgoat Mobile
CPU	Intel Core i7-8750H	Intel Core i7-3555LE
Number of cores	6	2
Number of threads	12	4
Clock frequency	2.2GHz	2.5GHz
Cache	9MB	4MB
RAM	32GB	N/A
Operating System	Windows 10 Enterprise	FreeDOS 1.0

References

[1] On-Road Automated Driving (ORAD) Committee. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, 06 2018.

[2] C. Katrakazas, M. Quddus, W. Chen, and L. Deka. Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. *Transp. Res. Part C Emerg. Technol.*, 60:416–442, 2015.

[3] U. Schwesinger, M. Rufli, P. Furgale, and R. Siegwart. A sampling-based partial motion planning framework for system-compliant navigation along a reference path. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 391–396, 2013.

[4] S. A. Bonab and A. Emadi. Optimization-based path planning for an autonomous vehicle in a racing track. In *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*, volume 1, pages 3823–3828, 2019.

[5] T. Faulwasser, B. Kern, and R. Findeisen. Model predictive path-following for constrained nonlinear systems. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 8642–8647, Dec 2009.

[6] V. A. Laurense, J. Y. Goh, and J. C. Gerdes. Path-tracking for autonomous vehicles at the limit of friction. In *2017 American Control Conference (ACC)*, pages 5586–5591, May 2017.

[7] D. Madås, M. Nosratinia, M. Keshavarz, P. Sundström, R. Philippsen, A. Eidehall, and K. Dahlén. On path planning methods for automotive collision avoidance. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 931–937, June 2013.

[8] J. Funke, M. Brown, S. M. Erlien, and J. C. Gerdes. Collision avoidance and stabilization for autonomous vehicles in emergency scenarios. *IEEE Transactions on Control Systems Technology*, 25(4):1204–1216, July 2017.

[9] K. Lee, S. Jeon, H. Kim, and D. Kum. Optimal path tracking control of autonomous vehicle: Adaptive full-state linear quadratic gaussian (lqg) control. *IEEE Access*, 7:109120–109133, 2019.

[10] D. Watzenig and M. Horn. *Automated Driving*. Springer International Publishing, 2017.

[11] R. C. Coulter. Implementation of the pure pursuit path tracking algorithm. Technical Report CMU-RI-TR-92-01, Carnegie Mellon University, Pittsburgh, PA, December 1991.

[12] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney. Stanley: The robot that won the darpa grand challenge. *Journal of Field Robotics*, 23(9):661–692, 2006.

[13] A. De Luca, G. Oriolo, and C. Samson. *Feedback control of a nonholonomic car-like robot*, pages 171–253. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

[14] J. M. Snider. Automatic steering methods for autonomous automobile path tracking. Technical Report CMU-RI-TR-09-08, Carnegie Mellon University, Pittsburgh, PA, February 2009.

[15] N. H. Amer, H. Zamzuri, K. Hudha, and Z. A. Kadir. Modelling and control strategies in path tracking control, for autonomous ground vehicles: A review of state of the art and challenges. *Journal of Intelligent & Robotic Systems*, 86:225–254, May 2017.

[16] L. Alonso, J. Pérez-Oria, B. M. Al-Hadithi, and A. Jiménez. Self-tuning pid controller for autonomous car tracking in urban traffic. In *2013 17th International Conference on System Theory, Control and Computing (ICSTCC)*, pages 15–20, 2013.

[17] A. O. M. Adeoye, B. I. Oladapo, A. A. Adekunle, A. J. Olademeji, and J. F. Kayode. Design, simulation and implementation of a pid vector control for ehvpmsm for an automobile with hybrid technology. *Journal of Materials Research and Technology*, 8(1):54–62, 2019.

[18] W. Farag. Complex trajectory tracking using pid control for autonomous driving. *International Journal of Intelligent Transportation Systems Research*, 18(2):356–366, may 2020.

[19] R. Marino, S. Scalzi, G. Orlando, and M. Netto. A nested pid steering control for lane keeping in vision based autonomous vehicles. In *2009 American Control Conference*, pages 2885–2890, 2009.

[20] P. Zhao, J. Chen, Y. Song, X. Tao, T. Xu, and T. Mei. Design of a control system for an autonomous vehicle based on adaptive-pid. *International Journal of Advanced Robotic Systems*, 9:1–11, 2012.

[21] F. Roselli, M. Corno, S. M. Savaresi, M. Giorelli, D. Azzolini, A. Irilli, and G. Panzani. H∞ control with look-ahead for lane keeping in autonomous vehicles. In *2017 IEEE Conference on Control Technology and Applications (CCTA)*, pages 2220–2225, Aug 2017.

[22] G. Tagne, R. Talj, and A. Charara. Higher-order sliding mode control for lateral

dynamics of autonomous vehicles, with experimental validation. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 678–683, June 2013.

[23] N. R. Kapania and J. C. Gerdes. Design of a feedback-feedforward steering controller for accurate path tracking and stability at the limits of handling. *Vehicle System Dynamics*, 53(12):1687–1704, 2015.

[24] E. J. Rossetter. *A Potential Field Framework for Active Vehicle Lanekeeping Assistance*. PhD thesis, Stanford University, 2003.

[25] J. Wang, J. Steiber, and B. Surampudi. Autonomous ground vehicle control system for high-speed and safe operation. *Int. J. of Vehicle Autonomous Systems*, 7:218–223, 07 2008.

[26] C. Hu, H. Jing, R. Wang, F. Yan, and M. Chadli. Robust h∞ output-feedback control for path following of autonomous ground vehicles. *Mechanical Systems and Signal Processing*, 70-71:414–427, 2016.

[27] C. Chen, Y. Jia, M. Shu, and Y. Wang. Hierarchical adaptive path-tracking control for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 16(5):2900–2912, 2015.

[28] Q. Yao, Y. Tian, Q. Wang, and S. Wang. Control strategies on path tracking for autonomous vehicle: State of the art and future challenges. *IEEE Access*, 8:161211–161222, 2020.

[29] H. Mouri and H. Furusho. Automatic path tracking using linear quadratic control theory. In *Proceedings of Conference on Intelligent Transportation Systems*, pages 948–953, 1997.

[30] R.S. Sharp, D. Casanova, and P. Symonds. A mathematical model for driver steering control, with design, tuning and performance results. *Vehicle System Dynamics*, 33(5):289–326, 2000.

[31] R.S. Sharp and V. Valtetsiotis. Optimal preview car steering control. *Vehicle System Dynamics Supplement*, 35:101–117, 2001.

[32] S. Jeon, K. Lee, H. Kim, and D. Kum. Path tracking control of autonomous vehicles using augmented lqg with curvature disturbance model. In *2019 19th International Conference on Control, Automation and Systems (ICCAS)*, pages 1543–1548, 2019.

[33] J. M. Maciejowski. *Predictive control: with constraints*. Prentice Hall, Harlow, England; New York, 2002.

[34] E. F. Camacho and C. B. Alba. *Model Predictive Control*. Springer-Verlag London, 2007.

[35] V. K, M. Ambalal Sheta, and V. Gumtapure. A comparative study of stanley, lqr and mpc controllers for path tracking application (adas/ad). In *2019 IEEE International Conference on Intelligent Systems and Green Technology (ICISGT)*, pages 67–71, June 2019.

[36] F. Yakub and Y. Mori. Comparative study of autonomous path-following vehicle control via model predictive control and linear quadratic control. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 229(12):1695–1714, 2015.

[37] G. V. Raffo, G. K. Gomes, J. E. Normey-Rico, C. R. Kelber, and L. B. Becker. A predictive controller for autonomous vehicle path tracking. *IEEE Transactions on Intelligent Transportation Systems*, 10(1):92–102, March 2009.

[38] A. Alessandretti, A. P. Aguiar, and C. N. Jones. Trajectory-tracking and path-following controllers for constrained underactuated vehicles using model predictive control. In *2013 European Control Conference (ECC)*, pages 1371–1376, July 2013.

[39] F. Yakub, A. Abu, S. Sarip, and Y. Mori. Study of model predictive control for

path-following autonomous ground vehicle control under crosswind effect. *Journal of Control Science and Engineering*, 2016, apr 2016.

[40] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat. Predictive active steering control for autonomous vehicle systems. *IEEE Transactions on Control Systems Technology*, 15(3):566–580, 2007.

[41] K. Berntorp, R. Quirynen, T. Uno, and S. Di Cairano. Trajectory tracking for autonomous vehicles on varying road surfaces by friction-adaptive nonlinear model predictive control. *Vehicle System Dynamics*, 58(5):705–725, 2020.

[42] R. Ritschel, F. Schrödel, J. Hädrich, and J. Jäkel. Nonlinear model predictive path-following control for highly automated driving. *IFAC-PapersOnLine*, 52(8):350–355, 2019.

[43] T. Novi, A. Liniger, R. Capitani, and C. Annicchiarico. Real-time control for at-limit handling driving on a predefined path. *Vehicle System Dynamics*, 58(7):1007–1036, 2020.

[44] Y. Chen and J. Wang. Trajectory tracking control for autonomous vehicles in different cut-in scenarios. In *2019 American Control Conference (ACC)*, pages 4878–4883, 2019.

[45] P. Falcone, H. E. Tseng, F. Borrelli, J. Asgari, and D. Hrovat. Mpc-based yaw and lateral stabilisation via active front steering and braking. *Vehicle System Dynamics*, 46(sup1):611–628, 2008.

[46] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou. Information theoretic mpc for model-based reinforcement learning. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1714–1721, 2017.

[47] X. Di and R. Shi. A survey on autonomous vehicle control in the era of mixed-autonomy: From physics-based to ai-guided driving policy learning. *Transportation Research Part C: Emerging Technologies*, 125:103008, 2021.

[48] R. Tchamna and I. Youn. Yaw rate and side-slip control considering vehicle longitudinal dynamics. *International Journal of Automotive Technology*, 14(1):53–60, 2013.

[49] L. Kakalis, A. Zorzutti, F. Cheli, and G. C. Travaglio. Brake based torque vectoring for sport vehicle performance improvement. *SAE International Journal of Passenger Cars - Mechanical Systems*, 1(1):514–525, Apr 2008.

[50] S. C. Baslamisli, I. Polat, and I. E. Köse. Gain scheduled active steering control based on a parametric bicycle model. In *2007 IEEE Intelligent Vehicles Symposium*, pages 1168–1173, 2007.

[51] C. E. Beal and J. C. Gerdes. Model predictive control for vehicle stabilization at the limits of handling. *IEEE Transactions on Control Systems Technology*, 21(4):1258–1269, July 2013.

[52] Y. Gao, T. Lin, F. Borrelli, E. Tseng, and D. Hrovat. Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads. In *2010 Dynamic Systems and Control Conference (DSCC2010)*, 2010.

[53] T. Lee, J. Lee, K. Ahn, S. Lee, and J. Kang. Nonlinear model predictive control for aggressive cornering maneuver considering effect of large steering angle. In *Asia-Pacific Automotive Engineering Conference*, Mar 2019.

[54] M. Thommyppillai, S. Evangelou, and R. S. Sharp. Car driving at the limit by adaptive linear optimal preview control. *Vehicle System Dynamics*, 47(12):1535–1550, 2009.

[55] A. Dakibay and S. L. Waslander. Aggressive vehicle control using polynomial spiral curves. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–7, 2017.

[56] K. M. Kritayakirana. *Autonomous Vehicle Control at the Limits of Handling*. PhD thesis, Stanford University, 2012.

[57] M. Malmir, M. Baur, and L. Bascetta. A model predictive controller for minimum time cornering. In *2018 International Conference of Electrical and Electronic Technologies for Automotive*, pages 1–6, 2018.

[58] E. Alcalá, V. Puig, J. Quevedo, and U. Rosolia. Autonomous racing using linear parameter varying-model predictive control (lpv-mpc). *Control Engineering Practice*, 95:104270, 2020.

[59] A. Liniger, A. Domahidi, and M. Morari. Optimization-based autonomous racing of 1:43 scale rc cars. *Optimal Control Applications and Methods*, 36(5):628–647, 2015.

[60] E. Velenis, D. Katzourakis, E. Frazzoli, P. Tsiotras, and R. Happee. Steady-state drifting stabilization of rwd vehicles. *Control Engineering Practice*, 19(11):1363–1376, 2011.

[61] J. Gonzales, F. Zhang, K. Li, and F. Borrelli. Autonomous drifting with onboard sensors. In *Advanced Vehicle Control – Edelmann, Plöchl & Pfeffer (Eds)*, 2016.

[62] R. Y. Hindiyeh and J. C. Gerdes. A controller framework for autonomous drifting: Design, stability, and experimental validation. *Journal of Dynamic Systems, Measurement, and Control*, 136(5), 07 2014.

[63] J. Y. Goh and J. C. Gerdes. Simultaneous stabilization and tracking of basic automobile drifting trajectories. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 597–602, 2016.

[64] J. Y. Goh, T. Goel, and J. C. Gerdes. A controller for automated drifting along complex trajectories. In *AVEC'18*, 2018.

[65] C. Chatzikomis, A. Sorniotti, P. Gruber, M. Zanchetta, D. Willans, and B. Balcombe. Comparison of path tracking and torque-vectoring controllers for autonomous electric vehicles. *IEEE Transactions on Intelligent Vehicles*, 3(4):559–570, Dec 2018.

[66] E. Siampis, E. Velenis, and S. Longo. Model predictive torque vectoring control for electric vehicles near the limits of handling. In *2015 European Control Conference (ECC)*, pages 2553–2558, July 2015.

[67] G. Vasiljevic and S. Bogdan. Model predictive control based torque vectoring algorithm for electric car with independent drives. In *2016 24th Mediterranean Conference on Control and Automation (MED)*, pages 316–321, June 2016.

[68] E. Mikuláš, M. Gulan, and G. Takács. Model predictive torque vectoring control for a formula student electric racing car. In *2018 European Control Conference (ECC)*, pages 581–588, June 2018.

[69] T. Shim, G. Adireddy, and H. Yuan. Autonomous vehicle collision avoidance system using path planning and model-predictive-control-based active front steering and wheel torque control. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 226(6):767–778, 2012.

[70] J. Guo, Y. Luo, K. Li, and Y. Dai. Coordinated path-following and direct yaw-moment control of autonomous electric vehicles with sideslip angle estimation. *Mechanical Systems and Signal Processing*, 105:183 – 199, 2018.

[71] M. Acosta, S. Kanarachos, and M. E. Fitzpatrick. On full magv lateral dynamics exploitation: Autonomous drift control. In *2018 IEEE 15th International Workshop on Advanced Motion Control (AMC)*, pages 529–534, March 2018.

[72] J. Ni, J. Hu, and C. Xiang. Envelope control for four-wheel independently actuated autonomous ground vehicle through afs/dyc integrated control. *IEEE Transactions on Vehicular Technology*, 66(11):9712–9726, 2017.

[73] J. Loyola and D. Margolis. Variable wheelbase reference for vehicle with active front and rear-wheel steering. *Vehicle System Dynamics*, pages 1–17, 2021.

[74] S. Saxena, V. Kumar, S. S. Luthra, and A. Kumar. 4 wheel steering systems (4was). In *National Conference on Recent Advances in Mechanical Engineering (RAME) – 2014*, volume 1, 2014.

[75] M. Völker and W. Stadie. Electrohydraulic rear axle steering systems for agricultural vehicles. *ATZoffhighway worldwide*, 10(4):22–27, 11 2017.

[76] C. Alexandru. A mechanical integral steering system for increasing the stability and handling of motor vehicles. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 231(8):1465–1480, 2017.

[77] B. Li, S. Rakheja, and Y. Feng. Enhancement of vehicle stability through integration of direct yaw moment and active rear steering. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 230(6):830–840, 2016.

[78] Y. Zhang, A. Khajepour, and X. Xie. Rollover prevention for sport utility vehicles using a pulsed active rear-steering strategy. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 230(9):1239–1253, 2016.

[79] B. Zhang, A. Khajepour, and A. Goodarzi. Vehicle yaw stability control using active rear steering: Development and experimental validation. *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*, 231(2):333–345, 2017.

[80] X. He, Y. Liu, K. Yang, J. Wu, and X. Ji. Robust coordination control of afs and ars for autonomous vehicle path tracking and stability. In *2018 IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 924–929, 2018.

[81] H. Zhang, B. Heng, and W. Zhao. Path tracking control for active rear steering vehicles considering driver steering characteristics. *IEEE Access*, 8:98009–98017, 2020.

[82] Vehicle dynamics and chassis components committee. *Road vehicles — Vehicle dynamics and road-holding ability — Vocabulary*, 12 2011.

[83] R. Rajamani. *Vehicle Dynamics and Control*. Springer, 2012.

[84] H. B. Pacejka and E. Bakker. The magic formula tyre model. *Vehicle System Dynamics*, 21(sup001):1–18, 1992.

[85] T. D. Gillespie. *Fundamentals of vehicle dynamics*. Society of Automotive Engineers, Warrendale, PA, 1992.

[86] A. Domahidi and J. Jerez. Forces professional. Embotech AG, url=https://embotech.com/FORCES-Pro, 2014–2019.

[87] A. Zanelli, A. Domahidi, J. Jerez, and M. Morari. Forces nlp: an efficient implementation of interior-point methods for multistage nonlinear nonconvex programs. *International Journal of Control*, pages 1–17, 2017.

[88] UKRI. Aid-cav. `https://gtr.ukri.org/projects?ref=104277`, 2018.

[89] Cranfield University. Multi user environment for autonomous vehicle innovation (mueavi). `https://www.cranfield.ac.uk/facilities/mueavi`, 2018.