An optimization based approach to autonomous drifting

A scaled implementation feasibility study

B. Verlaan

Master of Science Thesis

Delft University of Technology

Delft Center for Systems and Control

An optimization based approach to autonomous drifting

A scaled implementation feasibility study

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

B. Verlaan

April 15, 2019

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Abstract

Development of the autonomous vehicle has been a trending topic over the last few years. The automotive industry is continuously developing Advanced Driver-Assistance Systems (ADAS) that partially take over the driver's workload. This has resulted in an increase in vehicle safety and a decrease in fatal crashes [1]. Full vehicle autonomy has not yet been reached, as the control systems involved are not yet capable of handling every situation. One of these critical situations is when a vehicle enters the unstable motion of drifting. A vehicle is prone to drifting on low-friction surfaces, and also during these generally unstable maneuvers, the autonomous system should be able to remain in control. The performance of an autonomous drifting controller should be exemplified by the experience of rally drivers in how to handle a vehicle and keep control of a vehicle while drifting. The objective of this thesis is to design a control system which is capable of handling a vehicle during a drifting motion and to follow a certain desired path.

Vehicle dynamics are modeled as a three-state bicycle model to simplify the complex dynamics of the vehicle and the interaction between tyre and road. The definition of longitudinal wheel slip is reformulated to a smooth alternative to accommodate gradient based solving. With the system dynamics defined, the drifting motion is analyzed and equilibrium points are identified, showing differences between low- and high friction surfaces. Initially, a Model Predictive Control (MPC) strategy is applied with the purpose of steering the vehicle to desired drifting equilibria. Hereafter, the control system is extended to provide path following properties and addition of a dynamic velocity controller allows for a larger range of equilibria to be reached. The simulation setup intends to capture the experimental environment in the Network Embedded Robotics DCSC lab (NERDlab) at the Delft Center for Systems and Control (DCSC) department. Simulating a 1:10 scaled model allows to investigate the challenges that arise when implementing the control strategy on a scaled vehicle. These simulations show that autonomous drift control using the designed MPC strategy is possible, even when accounting for possible uncertainties such as delay, noise, and model mismatch.

Table of Contents

List of Figures

List of Tables

Preface and Acknowledgements

This is document is a part of my Master of Science graduation thesis.
Autonomous driving is currently seeing a huge development. This technology is seen as the future of (personal) transportation, which I believe will be a huge benefit to society and will help the overall mobility. This demand for mobility is increasing and as more and more vehicles enter the road result the congestion levels keep rising. Autonomous driving vehicles are ultimately expected to improve highway efficiency as computerized systems should be able to respond much quicker opposed to humans.
Though I see driving as a fun activity, and hope to see that humans will always still be able to drive manually as motor sports have always intrigued me. From looking at these motor sports I find rally-driving to be an interesting sport where high-skilled drivers perform extreme side-slip drifting maneuvers to tame a race track at high speeds. It is from this that I found inspiration in doing this research. Images as found on the cover of this thesis: *Mad Mike Whiddett performs in his BADBUL RX8 at Hampton Downs race track, Waikato, New Zealand on May 09, 2018 (©Graeme Murray/Red Bull Content Pool)*, excited and inspired me to work on this project.
As ADAS systems are being developed at a rapid rate I hope to contribute to the development of autonomous driving, starting with this thesis.

I would like to thank my supervisor dr.ir. Tamás Keviczky for his assistance during the writing of this thesis. My thanks also go out to dr. Barys Shyrokau for his feedback and lessons learned in vehicle dynamics control.

Delft, University of Technology B. Verlaan
April 15, 2019

“There is nothing like a dream to create the future.”
— *Victor Hugo*

Chapter 1

Introduction

1-1 Why autonomous drifting?

Ever since the first automobile was built by Nicolas-Joseph Cugnot in 1769 [4] a lot of progress has been made in the development of automobiles in terms of propulsion, comfort, and safety. With more than 700 million cars roaming the world, the automobility system currently brings fluid movement and freedom as its flexibility enables to travel at any time in any direction along the complex road systems of western societies [5]. This beneficial system, however, has also brought forth some issues.

The first accident to occur resulted in the death of Bridget Driscoll in August 1896. The death was labeled an ”accidental death”, and it was hoped that such a thing would never happen again. People perceived this as a chance phenomenon, a perception which was a characteristic view on road safety till the 1920s. The present paradigm is more developed, refraining from the term ”accident” to describe crash causes and lays focus on the interaction between humans, vehicles, roads, and types of errors, blaming the system as a whole.

Advanced Driver-Assistance Systems (ADAS) are developed to prevent crashes from happening in the first place, with examples as Anti-lock Braking Systems, Adaptive Cruise Control, and Lane Departure Warning Systems. These systems improve safety by intervening during dangerous situations and/or ease the driving workload by decreasing demands placed on the driver. This decrease in driver workload is a trending topic and leading towards the ability of vehicles to drive autonomously. However, before full vehicle autonomy can be reached the control systems facilitating this have to be able to handle the vehicle in all conditions and during all circumstances. Low-friction conditions as snow and ice greatly increase the risk of a vehicle crashing [6], and it is in these conditions that drivers lose control of the vehicle and potentially get into an unwanted drift.

Vehicles are prone to slip and have the possibility to enter a drifting state either due to aggressive evasive maneuvers or loss of control due to low friction road conditions. The average driver is not experienced with this vehicle behavior and is often not able to (retrieve) control during such an event. Particularly for this reason, it is useful for the vehicle to be able to take autonomous control of this drifting vehicle and be able to steer the vehicle into a safe trajectory. Autonomous Drift Control (ADC) could be this addition to the current ADAS, providing an increase of vehicle and traffic safety.

Rally-drivers set the example when it comes to controlled handling of a vehicle on its limits. Experience and skill with drifting allow them to corner faster, sustain drifting and safely return to a normal driving state. This control allows these drivers to push the limits of drifting at both high speeds and high body side-slip angles. The maneuverability which is normally only accessible to skilled rally drivers can be computed and performed autonomously by a Finite State Machine [7]. These state-of-the-art autonomous systems are able to control a vehicle and utilize the limit-handling which is drifting to achieve minimum time cornering [3, 8]. When it comes to safety concerns autonomous drifting has shown the ability to perform evasive maneuvers [9], showing that potential vehicle safety increase lies in controlling the (unwanted) drift.

The current situation shows that autonomous vehicles are in up-rise, and full vehicle autonomy is expected to arrive in 2030 [10]. ADAS systems are continuously being developed, tested and implemented to enable this transition to autonomous driving. It is important to continue research in this field to drive the transition to fully autonomous driving and allow safe application under all driving conditions.

Full vehicle autonomy places control of the vehicle outside of the hands of human drivers. This means that both the control of the steering and control of the velocity of the vehicle will have to be accounted for. The autonomous drifting control scheme must be designed in such manner that deviation of a desired, predefined, trajectory is minimized. To achieve this the controller should be able to change the driving velocity of the vehicle to accommodate a large range of curved trajectories. The control system must be able to compute the desired control action while driving and is thus required to compute solutions at least just as fast as the operating frequency. The system must also be robust enough to ensure an operating envelope which is large enough to allow activation in a wide range of situations and steer the vehicle towards a controlled drifting maneuver. After the design of this control system, the proposed strategy is evaluated for previously mentioned performance indicators.

Further simulation of a scaled version of the system, able to capture the experimental environment at the Network Embedded Robotics DCSC lab (NERDlab), will act as a feasibility study. This study will provide an illustrative example indicating under which circumstances implementation on a scaled system is possible. Result hereof will show which design aspects of both controller- and vehicle are more critical than others, as well as provide design criteria and recommendations for the experimental setup for future continuation of this research.

1-2 Research objective

In this thesis, a control system is designed to control a vehicle with a drifting motion on low-friction surfaces. The control system will be given full control of the vehicle to simulate a fully autonomous driving situation with the challenge being to autonomously initiate a drifting maneuver and thereafter stabilize this drifting motion. Enhancement of the control system will enable path-following properties and generate robustness for a larger range of scenarios. This system is first simulated on an actual sized vehicle whereafter a simulation study, aiming as a feasibility study, is done to analyze whether implementation of this control system on a scaled vehicle is feasible. As there are differences between ideal simulation conditions and actual real-world conditions, an attempt is done to account for these uncertainties. The final control system should be able to initiate and maintain a steady-state drifting motion for a scaled vehicle while regarding a large range of uncertainties.

This design of a control system capable of autonomous drifting on low-friction surfaces and performing a feasibility analysis for a Rear Wheel Drive (RWD) scaled vehicle implementation at the NERDlab at the Delft Center for Systems and Control (DCSC) department is divided into the following steps;

- Design of a controller capable of initializing a drift motion and stabilizing a RWD vehicle around a set drifting equilibrium.
- Expand this system so that the vehicle can be stabilized onto any reference equilibrium and can switch between various drift equilibria.
- Extend the control system design with a controller, allowing the system to exert path-following properties for any nonzero driving velocity.
- Adjust the controller in such a way that it is able to successfully control a vehicle model representative of a scaled vehicle at the NERDlab.
- Subject this model to a range of uncertainties which could occur at the NERDlab, re-tune and evaluate the designed control system for drifting behavior of the scaled vehicle.

1-3 Thesis outline

To achieve this, in Chapter 2 the physical system of a vehicle is analyzed to allow modeling of the vehicle dynamics and identify properties and characteristics of tyre-road interaction. Hereafter the drifting maneuver is dissected and analysis of the equilibria during drifting are discussed. This will allow for the design of a system which is able to generate reference signals for the system. This is done to accommodate various road/track properties which are discussed in Chapter 3. When all the boundary conditions are designed, the actual control system based on Model Predictive Control (MPC) is proposed in Chapter 4. This MPC controller is evaluated in Chapter 5 doing various drifting maneuvers and hereafter is when can be determined if the control system could be implemented on a scaled system. This feasibility study on a 1:10 scaled vehicle implementation is the subject of Chapter 6, and the robustness against various uncertainties is evaluated before concluding in Chapter 7.

Chapter 2

The vehicle as physical system

Vehicles exist in numerous variations ranging from large Sports Utility Vehicles to smaller and compact hatchbacks and many more. However, the dynamics describing these vehicles can be generalized to describe it's behavior. This vehicle can be described using quantities as the vehicle mass, height or length to define its dynamics. First, looking from above, a few things can be said regarding a vehicle traveling along a certain path.

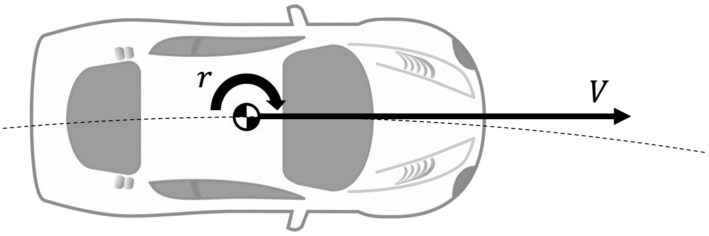

Figure 2-1: Vehicle taking a right-hand corner

Looking at the depicted vehicle driving along the dashed path in Figure 2-1 one can immediately identify two key characteristics. The vehicle is traveling at a certain velocity (V), and it is traveling along a curved trajectory (κ) due to a slight rotation (r) around it's Center of Gravity (CoG). Two equations can describe this characteristic behavior of the vehicle (2-1). These equations describe the curvature with which the vehicle is traveling and the overall vehicle velocity;

$$\kappa = \frac{r}{V} \tag{2-1a}$$

$$V = \sqrt{v_x^2 + v_y^2} \tag{2-1b}$$

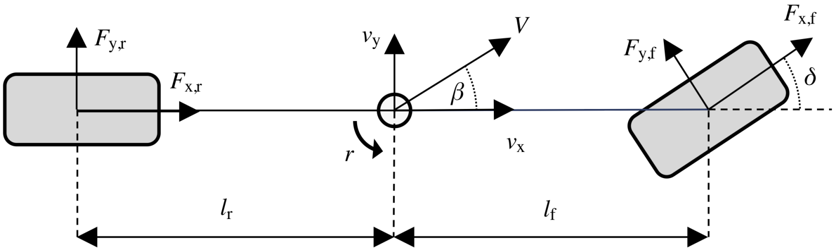

Figure 2-2: Bicycle Model

These two equations are described by three quantities, the longitudinal velocity v_x, the lateral velocity v_y and the yaw rate r. These quantities form the first physical states when describing vehicle dynamics by using differential equations. Dynamical modeling of vehicle behavior can be done in a various amount of ways, accounting or omitting for certain properties and simplifying the system where possible. The most extensive models will capture almost all of its dynamics, e.g. a full-car model will include roll-, pitch- and yaw-dynamics, include the behavior of the suspension system by modeling for sprung- and unsprung-masses and can include external forces as air resistance. Simplification of this model can be done by omitting roll- and pitch- motions and omitting the suspension system. This resulting two-track model now consists of four wheels and a rigid body, and even further simplification is possible.

2-1 Bicycle Model dynamics

Chosen is the three-state bicycle model to simulate the vehicle dynamics, a diagram describing this model can be seen in Figure 2-2. This simplification of vehicle dynamics is obtained when the two front- and two rear wheels are 'merged' together into a single front and rear wheel. The heading direction of the vehicle can be changed by steering with the front wheel (right in Figure 2-2), where the steering angle δ describes the angle of this front wheel with respect to the vehicle heading.

The mass is modeled as a single point at the CoG, separating the front and rear half of the vehicle. In this simplification, the pitch and roll dynamics are neglected and it is assumed to be a rigid body. Newton-Euler balances tell that forces acting on the tyres determine the accelerations and rotation of the vehicle. The differential equations governing this behavior are given as follows;

$$\dot{v}_x = \frac{1}{m}(F_{x,r} + F_{x,f}\cos(\delta) + F_{y,f}\sin(\delta)) - v_y r \tag{2-2a}$$

$$\dot{v}_y = \frac{1}{m}(F_{y,r} + F_{x,f}\sin(\delta) + F_{y,f}\cos(\delta)) + v_x r \tag{2-2b}$$

$$\dot{r} = \frac{1}{I_z}(F_{y,r}l_r + F_{x,f}\sin(\delta)l_f + F_{y,f}\cos(\delta)l_f) \tag{2-2c}$$

There are variations on the three-state bicycle model where often is chosen to use the body side-slip angle (2-3) as a differential state, for example as in [11]. Also, variations exist where the dynamics are modeled in a world-fixed frame coordinate system, though this method is mostly used for path planning purposes as in [12]. Chosen is to use the body-fixed frame method describing the decomposed velocities and the yaw rate as this gives the opportunity to place emphasis on both velocity states separately using the Model Predictive Control (MPC) techniques. The side-slip angle β is seen more as a *method* to take corners rather than the *goal* of the drifting maneuver.

$$\beta = \tan^{-1}\Big(\frac{v_y}{v_x}\Big) \tag{2-3}$$

Most of the parameters describing the vehicle dynamics are governed by the parameters describing the vehicle metrics, however, the forces that the tyres generate are possible subjects of control. The angle at which the front wheel is turned can be controlled either manually or autonomous through a steering wheel, and it is possible to apply gas or brakes in a car and with it change the forces on the tyres. In this analysis a Rear Wheel Drive (RWD) vehicle is taken, allowing only torque input on the rear wheel. From this we can conclude that a crucial element of vehicle dynamics control lies in controlling the forces generated by the tyres, and with this comes the opportunity to be in control of a vehicle's dynamics. To do this it is necessary to get an understanding of how these forces are generated from the interaction between tyres and the road surface.

2-2 Tyre-Road Interaction

The forces acting on the chassis to change the vehicle's velocity are generated at the tyres. Here an interaction takes place between the tyres and the road, friction generated by the relative velocities of the two translates into forces acting on the tyres. The magnitude of the force is determined by various properties of both the road an the tyres, in combination with the normal forces from the vehicle acting on the tyres. As there are many types of road surfaces, each bringing forward a different interaction between tyre and road. The road-friction coefficient μ_r tells us how much of the vehicles normal force acting on the tyres can maximally be translated into friction, generating forces on the tyre. For example, a dry asphalt road will have a typical μ_r of around 1, where an ice surface will bring forward a value of around 0.1.

Various methods to model these tyre-road interactions and generate the forces acting on the tyres are developed [13, 14]. Each model has it's pros and cons about whether the formulation is derived from physical properties or empirically designed, and whether it's complexity and accuracy fits the rate at which the forces are to be computed. In this analysis, the Magic Formula (MF) is considered to compute the forces from this tyre-road interaction. [15] compares the Dugoff, a modified Dugoff and the MF method to describe force generation. As comparable results are concluded in this analysis for the low slip-angle regions, at the higher slip-angle regions both Dugoff methods could not compare to the accuracy of the MF. Besides this, an 'if' condition exists in the Dugoff method to compute forces. The higher accuracy of the MF at high slip-angles and this non-smooth property within the Dugoff method provides two reasons why is chosen for the Magic Formula.

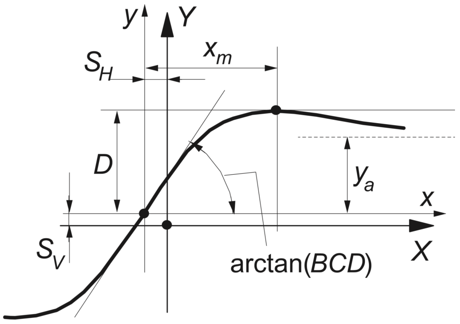

Figure 2-3: Curve produced by the original Magic Formula (2-4a), source [2]

2-2-1 The Magic Formula tyre model

Longitudinal and lateral forces acting on the front and rear tyres ($F_{i,j}$) are computed using the so-called Magic Formula (2-4)[2]. This semi-empirical model can be used for control system design purposes as the complexity is low, but accuracy can be achieved when fitted with enough empirical data [16]. Having a low complexity benefits high-frequency computation while the sin(arctan) formula provides excellent fit for F_x and F_y curves as depicted in Figure 2-3. These properties are critical elements for vehicle dynamics control.

The complete Magic Formula tyre model consists of over 20 formulas with over 100 parameters to compute longitudinal forces, lateral forces and self-aligning moments acting on the tyres. However this is not beneficial in the case of a high-rate of computations, and therefore a simplified model is chosen which has proven successful in the field of autonomous drift control [17].

$$y(x) = D\text{sin}[C\tan^{-1}\{Bx - E(Bx - \tan^{-1}(Bx))\}] \tag{2-4a}$$

with

$$Y(X) = y(x) + S_v \tag{2-4b}$$

$$x = X + S_h \tag{2-4c}$$

In this Magic Formula, X and Y are the input and output variable respectively. This simplified formulation contains six variables; B being the stiffness factor, C is the shape factor, D is peak value, E is the curvature factor, S_h defines horizontal shift and S_v vertical shift. The variables describing the Magic Formula can be chosen to be directional dependent for longitudinal and lateral forces. However, isotropic behavior is assumed in this analysis.

$$\mu_{i,j} = \frac{\sigma_{i,j}}{\sigma_j} D\text{sin}(C\text{atan}(\sigma_j B - E(\sigma_j B - \text{atan}(\sigma_j B)))) \tag{2-5a}$$

$$F_{i,j} = \mu_{i,j} F_{z,j} \tag{2-5b}$$

Formulation of the Magic Formula (2-5) is done as described by D. Tavernini [3] to compute the slip coefficient μ, where coupling between longitudinal and lateral forces is as described

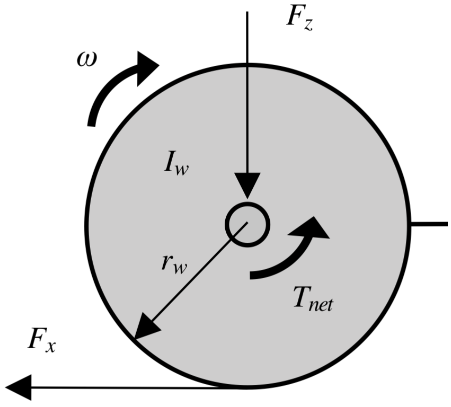

Figure 2-4: Wheel dynamics model during braking

in [2] using the theoretical slip quantities (2-9). Horizontal and vertical shift of the friction curve is neglected in this formulation. The analysis of this tyre model is continued after the introduction of slip and wheel dynamics (Section 2-2-3). The computation of the theoretical slip is done when analyzing the wheel dynamics which describe the relative motion between the tyre and the road.

2-2-2 Slip and Wheel dynamics

Slip occurs when there is a relative motion present between two surfaces. In the case of tyre-road interaction, this slip can be found when looking at the velocity at which the contact area of the tyre is traveling due to the angular velocity of the wheel, and comparing this to the velocity at which the road surface is moving with respect to the vehicle. Assumed here is that the vehicle is a rigid body and that the tyre is perfectly round, depicted in Figure 2-4. Here, a (braking) torque T_{net} is applied to the wheel which will result decrease of the angular velocity ω. The dynamical equation describing this behavior is given in (2-6).

$$\dot{\omega} = \frac{1}{I_w}(-F_x r_w + T_{\text{net}}) \tag{2-6}$$

This difference in velocities of the vehicle and tyre is commonly described by the longitudinal wheel slip coefficient λ.

$$\lambda = \frac{r_w\omega - v_x}{\max(r_w\omega, v_x)} \tag{2-7}$$

Besides slip in the longitudinal direction, the angle of the wheel with respect to the vehicle velocity results in lateral wheel slip α_j. As the front wheel can be subjected to a steering angle, the formulation of these descriptions varies. Also, the partial velocity generated by the yaw rate of the vehicle accounts for the difference between descriptions.

$$\alpha_f = \delta - \tan^{-1}(\frac{rl_f}{v_x} + \frac{v_y}{v_x}) \tag{2-8a}$$

$$\alpha_r = -\tan^{-1}(\frac{-rl_r}{v_x} + \frac{v_y}{v_x}) \tag{2-8b}$$

As these two slip quantities are to be combined into a homogeneous slip description, theoretical slip quantities (2-9) are used to quantify the slip angle and longitudinal slip coefficient in directional vectors. When computed, these theoretical slip quantities can be used by Tavernini's deviation of the Magic Formula mentioned before.

$$\sigma_{x,j} = \frac{\lambda_j}{1+\lambda_j}, \quad \sigma_{y,j} = \frac{\tan(\alpha_j)}{1+\lambda_j}, \quad \sigma_j = \sqrt{\sigma_{x,j}^2 + \sigma_{y,j}^2} \tag{2-9}$$

Due to the non-smooth nature of the max-function in (2-7) gradient based solvers experience difficulty working with this equation. To approximate the max-function a smooth alternative has been used [18]. The soft maximum approaches the hard maximum when the difference between the two values compared increases. Addition of the parameter ρ increases separation of the two compared values for $\rho > 1$. However, as the difference between $r_w\omega$ and v_x decreases, so does the nominator of the fraction defining the slip, meaning that the absolute value of λ (and thus also the absolute error due to the approximation) decreases.

$$\tilde{\lambda}_j = \frac{r_w\omega - v_x}{\frac{1}{\rho}\log(e^{\rho r_w \omega} + e^{\rho v_x})} \tag{2-10a}$$

$$\tilde{\lambda} \to \lambda \quad \text{for} \quad \rho \to \infty \tag{2-10b}$$

2-2-3 Tyre Variations

The MF contains four parameters $(B - E)$ which can be set to vary the tyre-road interaction. Tavernini defined four different tyres (Table 2-1) to simulate various surface behaviors ranging from dry asphalt to soft off-road surfaces [3]. As this covers a wide range of scenario's, and due to the fact that these tyres are considered in various other researches on automobile drifting these four variations are initially considered in this research. Their respective friction ellipses can be observed in Figure 2-5.

These four tyre types can be split into two groups, the first of which contains convex slip properties where the slip properties of the other are non-convex. These properties can be observed with the help of the friction ellipse, where only $\frac{1}{4}$ of each is depicted as the tyres are assumed isotropic in x- and y directions. Convex slip properties for the tyre mean that there is only one $\alpha - \lambda$ pair which determines a specific $\mu_x - \mu_y$ pair. The tyre types 3 and 4 can be identified as having these convex properties. The tyre sets with non-convex slip properties (tyre 1 and 2 in Figure 2-5) show us that there are two intersections between α and λ curves. For a certain force to be generated from the tyre-road interaction, multiple $\alpha - \lambda$ pairs suffice. This multiplicity of steady-states capable of completing the same corner tells us that there are multiple driving modes available for this tyre type.

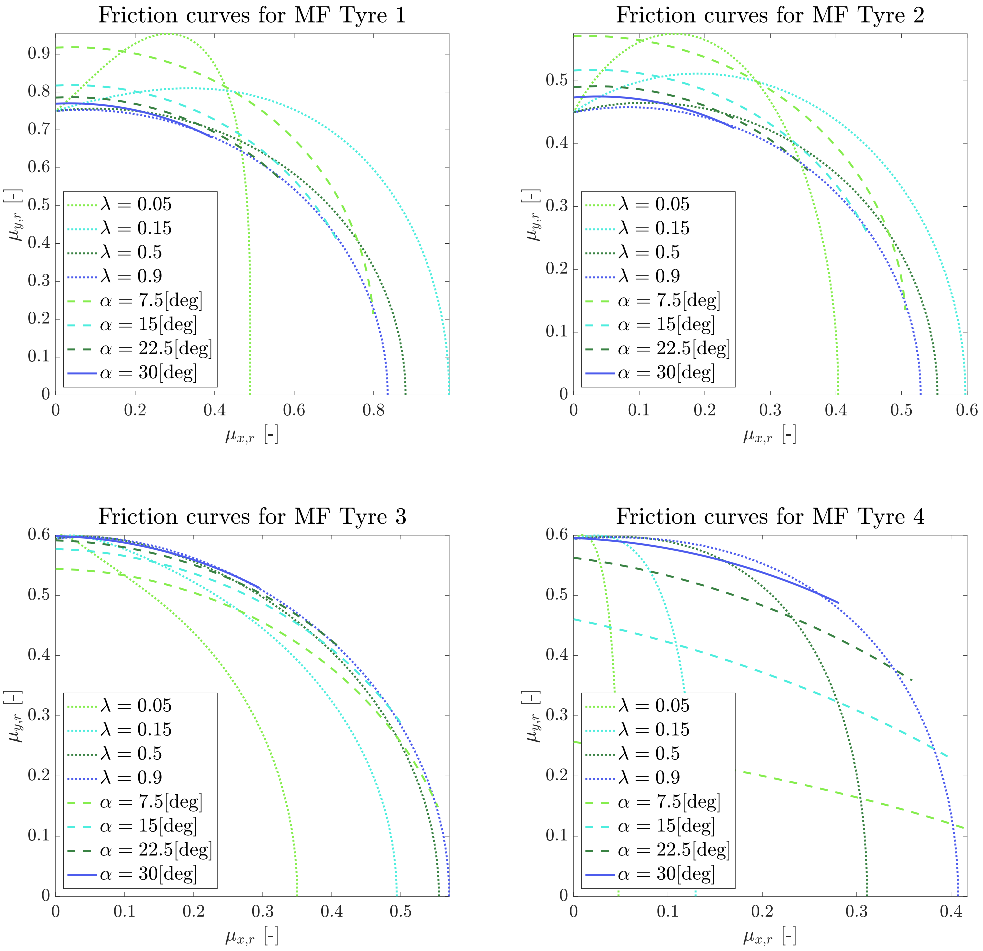

Figure 2-5: $\frac{1}{4}$ Friction ellipses for four tyre types (Table 2-1)

Table 2-1: MF parameters for various road conditions, source [3]

Tyre	B	C	D	E
1	6.8488	1.4601	1.0	-3.6121
2	11.415	1.4601	0.6	-0.20939
3	15.289	1.0901	0.6	0.86215
4	1.5289	1.0901	0.6	-0.95084

2-3 Summary

The vehicle as a physical system has been the topic of discussion in this chapter. Differential equations make it possible to describe vehicle dynamics in a general manner, allowing all sorts of vehicles to be modeled when taking into account various assumptions and simplifications. Simplifications of these sorts have brought forward the three-state bicycle model of a vehicle which, in further chapters of this thesis, will be taken as the method to describe vehicle dynamics. The dynamics of the wheel bring forward a fourth differential equation, also in a simplified form, to conclude the description of the system states. A slight modification to the calculation of the longitudinal wheel slip ratio λ is made by substituting the *max*-function for a smooth alternative. The interaction between the tyres of the vehicle and road-surface has been described in a simplified manner using an isotropic Magic Formula tyre model. With these mathematical formulations, the forces acting on the vehicle can be computed, and result in the simulation of a moving vehicle. This simulation can be done with the vehicle performing a large range of maneuvers, traveling along a large range of paths. The following chapter discusses in more detail the drifting maneuver, ranging from initiation of the drift to stabilization hereof, and defining the paths along which the vehicle should drift.

Chapter 3

Analysis of drifting equilibria and reference trajectories

The characterizing movement of a vehicle with a high body side-slip angle portrays the image of a drifting vehicle. Rally drivers are commonly known to be skilled at controlling this maneuver, and when they do they make it seem as if it were an easy task. However, the motion of a drifting vehicle is highly unstable, meaning that without proper control the vehicle will either return to normal driving (low body side-slip angle) state or continue spinning without the possibility of regaining control. Being able to keep the vehicle motion stable requires pursuit of (drifting) equilibria, analysis of these equilibrium points is the subject of the following chapter.

3-1 The art of drifting

Stability of drifting equilibria comes from balancing forces, acting from the wheels on the chassis of the vehicle. Regarding the bicycle model described in Section 2-1 equilibrium occurs when the forces acting on the rear and front tyre are in balance, resulting in no accelerations and allowing control of the body side-slip angle [19]. However steady-state drifting does not occur instantaneously and requires a series of actions for one to describe a successful drift [20];

1. Drift initiation
   The vehicle approaches a turn and by saturating the rear tyres a moment around the body produces a yaw rate increase, inducing a side-slip angle to the vehicle body.

2. Steady-state drifting
   When the vehicle is in a drifting state it is necessary to stabilize yaw rate and side-slip and control these to achieve path following properties.

3. Low side-slip stabilization
   After the turn is made the vehicle will exit the drift by desaturating the tyres and re-entering normal driving conditions with low side-slip.

After initiating the drift by performing a steering wheel maneuver a yaw moment is generated which determines the direction of the drift. Hereafter it is necessary to saturate the rear tyre to increase this yaw moment further and obtain a high side-slip angle.

Tyre saturation

There is a limit on forces that tyres can exert on the road, defining the saturation limit. The forces that saturate the tyres are composed of lateral and longitudinal force directions. This composition of forces can be found in the $\frac{1}{4}-$friction ellipses in the previous chapter. At the intersection of the curves defined by the longitudinal slip λ and the lateral slip α_j lies a combination of μ_i values which directly correlate to the force composition (2-5).

The longitudinal force is the result of torque applied to the axis connecting the engine and wheel of the vehicle and, under normal driving conditions, results in acceleration of the vehicle. The lateral force is generated when the vehicle is in cornering state, thus when changing the heading of the vehicle, or when changing the steering angle of the front wheel. The drift is properly initiated after this saturation of the rear-tyres, something that can be accomplished by either reducing or accelerating the angular motion of the wheels by braking or applying extra throttle. This results in an increase in the amount of wheel slip $\sigma_{i,j}$ applied to the tyres and because of this the velocity of which the rear tyres move increases. This (generally) unstable mode results in the rear wheels trying to surpass the front wheels, and with it generates an increase of the yaw motion of the vehicle. This yaw motion defines the side slip angle and naturally the yaw rate of the vehicle.

Side-slip and yaw rate

Once the vehicle enters the drifting mode by saturating the rear tyres the side-slip angle increases due to the rear wheels having a higher velocity than the front wheels. From here on it is necessary to control the forces acting on the tyres and with it stabilize the yaw rate of the vehicle. This can be done by either redirecting the forces of the front tyre by actuating the steering wheel or by applying a different amount of force on the rear wheel by reapplying the throttle or brake. Actuating the steering wheel redirects the front wheel forces. To balance out the yaw moment the steering wheel is to be directed towards a direction opposite to the cornering direction, an action called counter steering.

Counter steering

Redirection of the front wheel counters the yaw moment of the vehicle applied after initiating the drift. To do this the steering wheel, and with it, the front wheel is turned in the direction opposite to the cornering direction [19]. For low side-slip angles, a small change of angle must be taken. For larger angles, the steering wheel is turned much further. When the steering action changes direction, the action is defined as counter steering, essential for high side-slip control.

3-2 Equilibrium Analysis

The stability of Rear Wheel Drive (RWD) vehicles when drifting is an unstable motion that needs to be controlled by the driver or a control system [21]. However is unstable behavior contains a number of equilibria, and for certain steering wheel inputs unstable saddle points describe two or one steady-state drifting modes as well as one normal driving mode [22]. In his analysis, Gerdes analyses the stability of drifting vehicles and distinguishes three types of equilibria [23]. The first defines 'normal' driving (neutral steer), where the other two describe drifting due to extreme oversteer and extreme understeer behavior. Over- and understeer happens when there is a difference in the front and rear slip angle α_j.

The working points for steady-state cornering are determined by analysis of the equilibrium points ($\dot{x} = 0$). When cornering a vehicle there are two properties to keep in mind, namely the desired velocity and the desired curvature. Setting a grid of points with velocity and curvature on the axis will allow finding equilibrium points for each possible driving state. The dynamics defining these states come from the algebraic description of the vehicle.

3-2-1 Definition of the Differential-Algebraic set of Equations (DAE)

The differential states of (2-2) - (2-10) are considered. For equilibria, the differential states are equal to zero, which tells us that there is a constant yaw rate. This is only possible when the moment creating this angular velocity is equal to zero. The lateral forces from the tyres acting on the vehicle body define this balance, and as the condition $\lambda_f = 0$ is chosen as assumed is that the front wheel can roll freely, the sole lateral force on the front of the vehicle is defined by the slip angle α_f. The force is defined by the tyre-road properties, discussed in Section 2-2, and from this can be said that for the non-convex tyre properties there exist multiple equilibria for certain radius-velocity pairs. With the dynamics of the vehicle constraining the nonlinear system, it is possible to solve this set of DAE to find equilibrium points.

3-2-2 Solving the DAE

An initial guess sets the starting point for the algorithm to search from. As there may be a multiplicity in solutions it is crucial that the initial guess is a good estimate of the solution. As for 'normal driving conditions' the body side-slip angle β is low, the initial guess for the longitudinal velocity is estimated to be the total vehicle velocity. In the event of having found a suitable equilibrium, as the search advances to a slightly higher velocity, the solving process is warm-started by using the previous solution as the initial guess.

In the case of finding drifting equilibria, an extra constraint is added to the set of equations to be solved, described in (3-1c). Initially was thought that this would constrain the search to conditions where the tyres were saturated, though mathematically this constraint is always fulfilled. However interesting enough the solution will not converge successfully towards drifting equilibria without this constraint.

The MATLAB function *fsolve* is used to solve this nonlinear set of equations. This function uses the 'trust-region dogleg' algorithm which incorporates the Powell dogleg procedure for

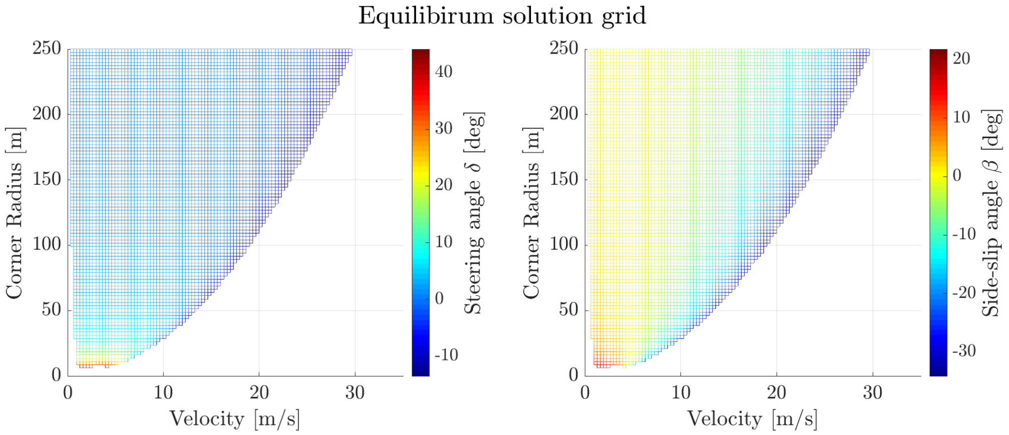

Figure 3-1: Equilibrium solutions for Tyre 4

computing the step. Though this gradient based solver is not able to identify global minima, a good initial guess and multiple searches allow finding each equilibrium set.

$$\kappa = \frac{r}{V} \tag{3-1a}$$

$$V = \sqrt{v_x^2 + v_y^2} \tag{3-1b}$$

$$F_{z,r}\sqrt{\mu_{x,r}^2 + \mu_{y,r}^2} = \sqrt{F_{x,r}^2 + F_{y,r}^2} \tag{3-1c}$$

From this search a grid of equilibrium points arises.

3-2-3 Analyzing found equilibria

From Figure 3-1 it can be seen that there exists a large range of possibilities for a vehicle to take a corner with a certain radius. For each corner, there exists a range of velocities accompanying steady-state cornering conditions, limited by a maximum velocity. When approaching the maximum velocity for each cornering radius the body side-slip angle decreases to high-negative values. Approaching these high side-slip conditions the steering wheel angle crosses the zero-line and decreases to negative values indicating counter steering behavior.

The non-convex tyre has a slightly more complex equilibrium grid as there is a multiplicity for certain driving states, visible in Figure 3-2. Looking at the top view one will see that the profile shape of the equilibrium solutions is somewhat equal, though this tyre allows higher velocity cornering. Not all the properties are equal, as there exists a second plane with lower (negative) body side-slip angles. These drifting equilibria contain both oversteer and understeer areas where understeer allows more extreme side-slip angles.

Centripetal acceleration is defined as $a_c = \frac{V^2}{R}$, telling that either taking a corner at a maximum velocity or taking the smallest radius corner at a given velocity will bring maximum centripetal acceleration. Therefore the southeast edge of this top view determines the states with maximum centripetal acceleration. This edge contains almost only states that belong to normal cornering properties, however, an extra region which looks like an elongation of

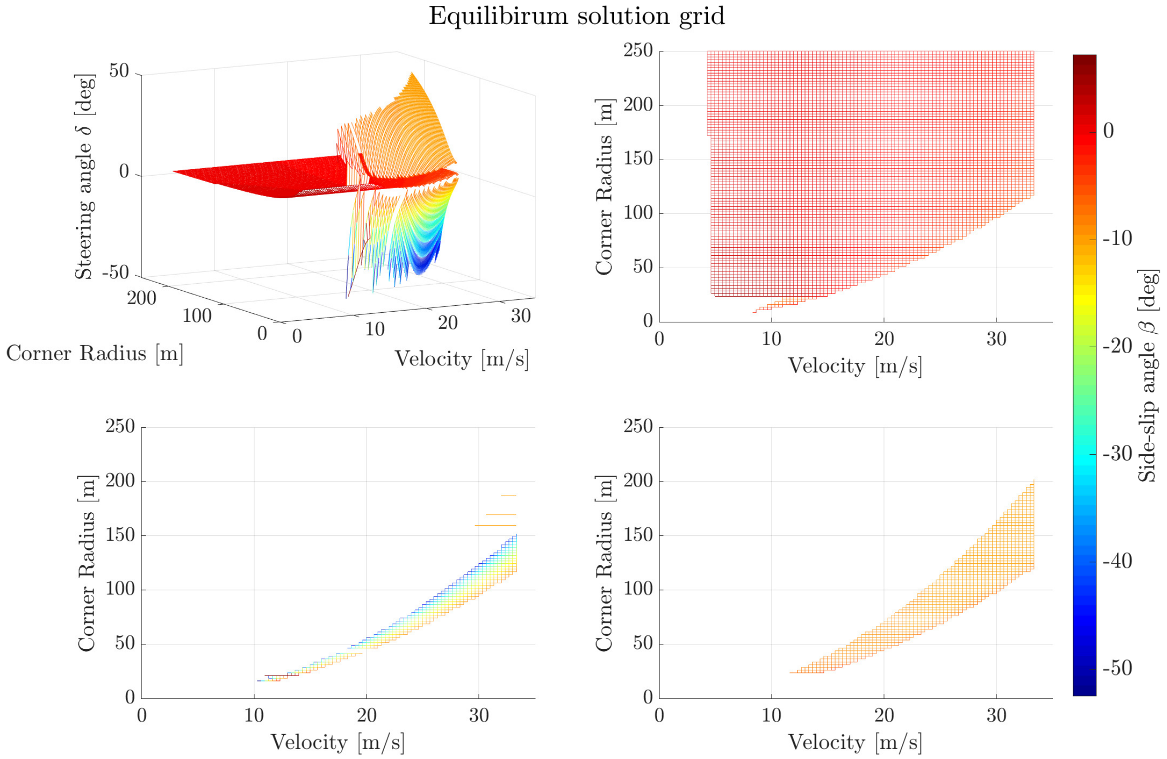

Figure 3-2: Equilibrium solutions for Tyre 1

this edge belongs to drifting equilibria. This tells that there exists drifting equilibria during which smaller radius cornering is possible, an area which is particularly interesting for obstacle avoidance situations.

With the driving equilibria found and with it the boundaries set it can be concluded that there is a limited range of corners which the vehicle is able to take at a set velocity. The next section discusses the formulation of the possible trajectories according to these corner radii.

3-3 Track definition and deviation

Looking at a track from above will instantly reveal the Cartesian coordinate system, which has a fixed axis on the track geometry. However, the axis of a vehicle is fixed to the body of the vehicle which, when moving along a curved trajectory, will rotate with respect to the Cartesian coordinates. To evaluate whether the vehicle follows a predefined track it is much easier quantified using curvilinear coordinates. These tracks can be defined by their curvature and definition hereof is the subject of this section.

3-3-1 Track segments

Constant radius circle

With this coordinate system in mind, a track layout can be rewritten from Cartesian to a curvilinear description. The curvature of a track segment is defined by the inverse of the radius of the arc that the segment produces. Circles with a constant radius, therefore, have a constant curvature, depicted in Figure 3-3. This track type is interesting because it shows whether the control scheme is able to stabilize the vehicle around the proposed equilibrium.

Clothoid transition

Once the ability to stabilize the vehicle has been confirmed, the arc radius of the curve can be varied over its length linearly (Figure 3-4). This results in a clothoid track segment, similar to an Euler spiral. A clothoid track type is interesting because the controller's ability to track a moving equilibrium is accessed, concluding the cornering ability of the controller. Clothoid segments are implemented in highway and road designs [24].

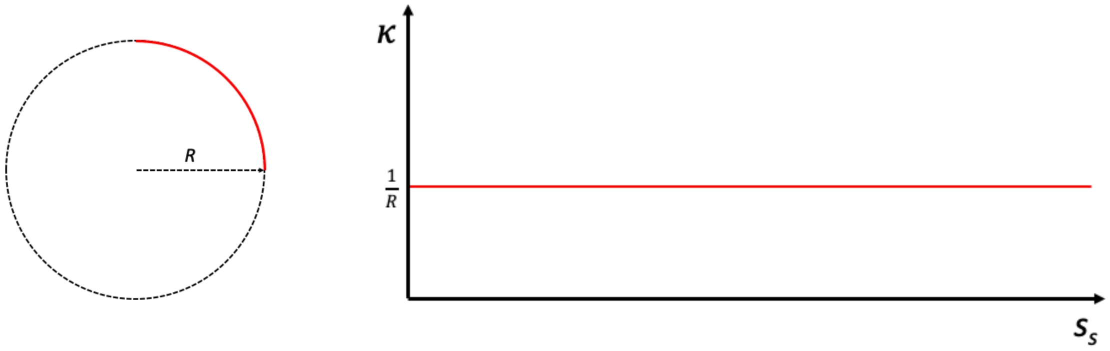

Figure 3-3: Constant radius track segment

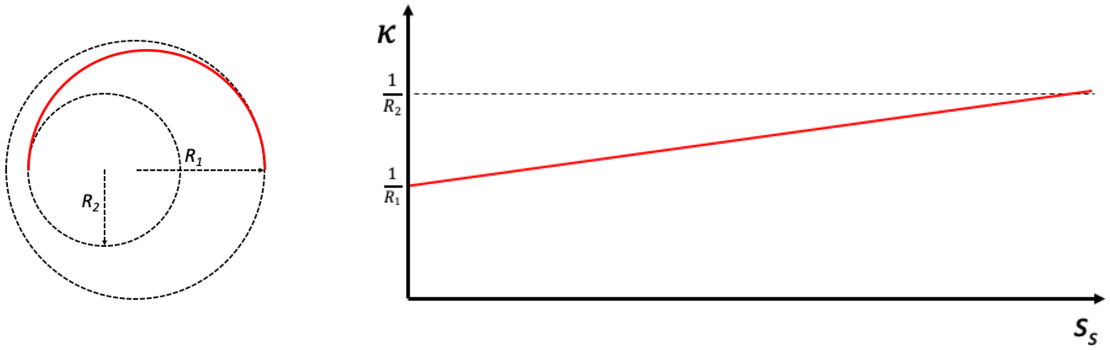

Figure 3-4: Clothoid track segment

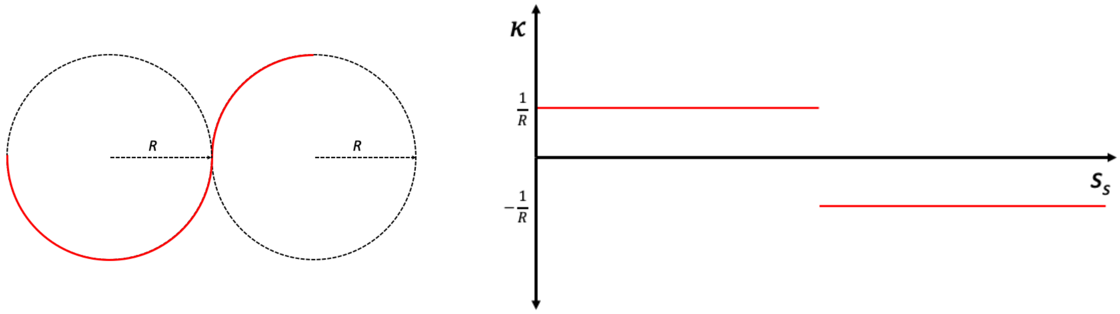

Figure 3-5: Transition in cornering direction

Directional transition

As corners have to be taken in both directions the ability to switch between two directions is analyzed. Changing the turning direction of the corner will not change its radius. However, a distinction has to be made to determine whether the vehicle must turn left or right. That is why in this analysis taking left-handed turns is defined by a positive curvature signal and right-handed turn defined by a negative κ, shown in Figure 3-5. This complies with the description of the curvature which a vehicle is taken ($\kappa = \frac{r}{V}$), where clockwise rotation implies a negative yaw rate (r) and counter-clockwise rotation a positive yaw rate. The transition houses a zero-crossing at the instance when the cornering direction is switched, and whether the controller is able to seamlessly handle such situations is something that is evaluated using this track type.

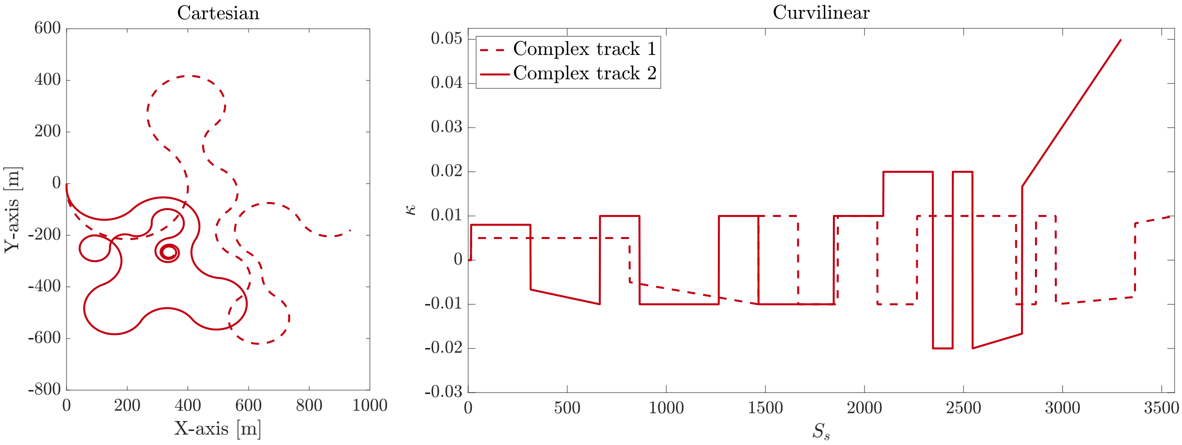

Figure 3-6: Two complex tracks

Complex tracks

Furthermore, two tracks are composed consisting of multiple constant radius circles, clothoids, and transitions. The first of these tracks contain a large sequence of turns with approximate equal radius, ideal for testing scenarios with a constant reference velocity. The second track contains steeper transitions and smaller radius corners where the dynamic reference velocity must be enabled. In the case that this is not enabled, setting a reference velocity for the vehicle may result in the system demanding the vehicle to take a corner which radius is too small for the vehicle to take at the set velocity. The layout of the tracks in Cartesian coordinates and the curvilinear description hereof can be seen in Figure 3-6.

3-3-2 Curvilinear coordinates

This coordinate system (3-2) is primarily defined by a curved axis and the orthogonal direction at each point along the line. Besides this axis a rotational velocity based on the tangent vector of the curve is evaluated, defining the change of heading.

$$\dot{s}_s = \frac{V\cos(\alpha+\beta)}{1-s_n\kappa} \tag{3-2a}$$

$$\dot{s}_n = V\sin(\alpha+\beta) \tag{3-2b}$$

$$\dot{\alpha} = r - \kappa\frac{V\cos(\alpha+\beta)}{1-s_\mathrm{n}\kappa} \tag{3-2c}$$

Measurement of the vehicles ability to follow the path begins by measurement of the system states. The body side-slip angle β is a difficult to measure quantity due to physical and economic reasons as it requires precise measurement of the decomposed vehicle velocity states (2-3).

3-3-3 Error definition

To determine path following capabilities the curvilinear coordinate system (3-2) as described previously is adopted. The use of these coordinates can be done with knowledge of the three system states $x = [v_\mathrm{x} \quad v_\mathrm{y} \quad r]$. In this coordinate system it is possible to determine the distance traveled along the desired path s_s, a property which is particularly useful when the path layout is known beforehand. Besides this, the lateral deviation from the desired path s_n can be found (Figure 3-7). As measurement of this signal is done in a discrete manner, quantification of the path following ability of the system can be done by taking the Root-Mean-Square Error (RMSE) value of this deviation.

Tracking of the desired path is achieved when $\epsilon = \alpha + \beta = 0$. The reason that these two properties have opposite signs is that the heading angle is taken with respect to the velocity vector, where the side-slip angle is taken with respect to the vehicle center-line. The error between the desired heading and side-slip angle can be seen in Figure 3-8.

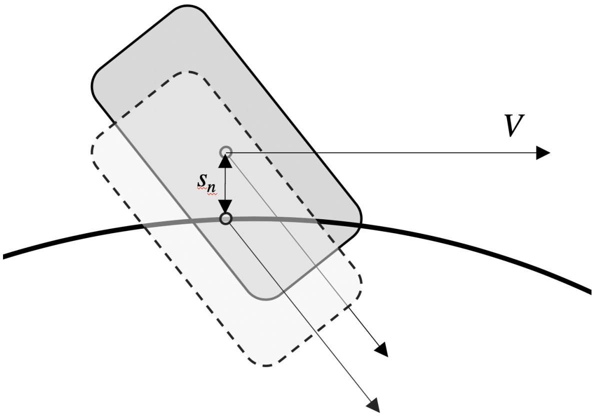

Figure 3-7: Lateral path deviation s_n

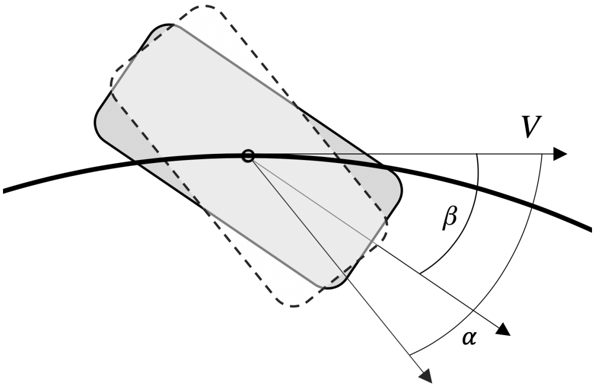

Figure 3-8: Desired vehicle heading α and side-slip angle β

3-4 Summary

Rally drivers are capable of cornering vehicles with a controlled execution of a drifting maneuver. During this maneuver the tyres are saturated, the body side-slip angle is much higher than during normal driving conditions and the driver must counter-steer to maintain the desired yaw rate. When these situations occur the vehicle operates at a highly unstable mode. However, this unstable mode still contains equilibria which, when steered toward correctly, can be balanced upon. The differential equations governing the vehicle dynamics are the key to finding these equilibria. An entire grid with potential driving modes can be found on the $R - V$ plane, with each point on this grid having a set of equilibrium states and control inputs. Different tyre-road descriptions show different $R - V$ planes where high-friction surfaces contain multiple driving modes and for low-friction surfaces, there is a smooth transition between normal driving- and drifting conditions. The path to be taken is described using the curvilinear coordinate system, having a κ_{ref} value for each point on the path. Movement on the $R - V$ grid, regarding the reference curvature, brings forward equilibrium trajectories for the vehicle states to follow and provide control signals for the vehicle. Steering the vehicle towards these equilibrium states is to be done using a control system which is designed in the following chapter. The ability to correctly follow the desired path can also be measured using the curvilinear coordinate system and provides useful information which is used to expand the control system design.

Chapter 4

Control System Design

Design of the control system can be done using various control strategies. Optimization-based methods include Linear Quadratic Regulator (LQR), Model Predictive Control (MPC) and State Dependent Ricatti Equation (SDRE) methods which all have been used for various autonomous drift control purposes [11, 25, 26]. As MPC methods are found to be too computational intensive for implementation in vehicle dynamics control, due to the high operating rate required and computational demanding optimization problem needed to be solved, the Ricatti-based control strategies are generally used. Using a MPC strategy for autonomous drifting brings forth the challenge to find a formulation which allows solving of the Optimal Control Problem (OCP) fast enough. This chapter describes the MPC-based method used in this thesis.

A general overview of the control scheme can be seen in Figure 4-1. The previous sections have covered the offline computation of drift equilibria and the vehicle as a physical system. This section covers the working of the different mechanisms acting together to compute the control signals to enable drifting of the vehicle.

The system is built using four components; one low-level Model Predictive Controller to generate the control signals for the vehicle, one Reference generator to compute the reference horizon for the MPC and two high-level controllers which pre-process the reference signals sent to the Reference Generator.

4-1 Path following controller

From the measurement of the vehicle states and the equations describing the curvilinear coordinate system, the lateral path deviation s_n, and desired vehicle heading α can be determined. In their turn, these values can be used to manipulate the desired path curvature κ_{ref} to obtain path-following properties. The difference between desired and actual vehicle heading is found as; $\epsilon = \alpha + \beta$. A Proportional Integral Derivative (PID) control strategy, initially tuned to underdamp s_n, is chosen as the vehicle cannot instantly change its direction. Therefore it is prone to initially taking a larger radius corner than desired. This larger radius corner will

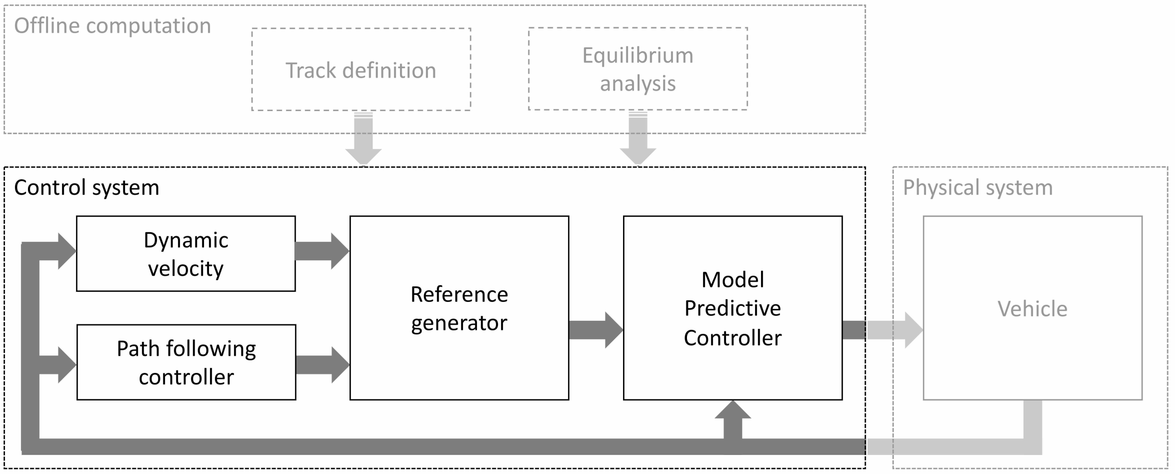

Figure 4-1: General control system overview

always fall within the possible $R-V$ equilibrium solutions when choosing constant velocity. However, when the taken corner radius is smaller than desired, the PID controller will enlarge the desired radius, potentially leaving the possible driving envelope. The weights are static, and therefore the performance of the controller is different for varying vehicle velocities and curvatures. The general description of the controller is as follows;

$$\Delta\kappa = K_{p1}s_n + K_{i1}\int_0^t s_n + K_{d1}\frac{\Delta s_n}{\Delta t} + K_{p2}\epsilon + K_{i2}\int_0^t \epsilon + K_{d2}\frac{\Delta\epsilon}{\Delta t} \tag{4-1a}$$

The computed control signal is then added to the reference signal (4-2), converted to cornering radius ($R_{\text{ref}} = \frac{1}{\kappa_{\text{ref}}}$) which is to be processed by the Reference Generator.

$$\bar{\kappa}_{\text{ref}} = \Delta\kappa + \kappa_{\text{ref}} \tag{4-2a}$$

Tuning of this controller preferably is done for each varying $R-V$ pair as the response of s_n is very dependent of the vehicle velocity. The desired response of the controller should not result in exceeding the maximum velocity. To make sure that this does not happen a dynamic reference controller for the velocity V is designed. This will allow a more rigorous response from the PID controller as overshooting the maximum velocity will not happen anymore.

4-2 Dynamic reference velocity

Choosing a fixed vehicle velocity as a reference allows only a certain minimum radius corner to be taken. For the vehicle to take smaller radius corners, the reference velocity must be lowered. Also, when the desired corner has a larger radius than the minimum allowable corner, for the fixed velocity, there lies a range of greater velocities at which the corner can be taken. Allowing a controller to dynamically set the reference velocity according to the cornering radius will allow for a larger set of corners to be taken while cornering at a higher velocity.

The maximum allowable velocity for a certain corner is determined offline during the computation of the equilibrium grid, and the driving velocity is assumed to be measured.

As the robustness against initial condition variation (further discussed in section 5-1) indicates that for desired velocity increase there mustn't be too large a difference, where for the desired velocity decreases the algorithm is much more robust. Therefore a graduate increasing reference velocity is chosen while a hard decrease in velocity taken (4-3a). Hereafter a second check is done to verify whether the new reference doesn't exceed the maximum velocity (4-3b). The coefficient $c \in [0\ 1]$ is used as a tuning parameter to slightly reduce the reference velocity for robustness reasons.

$$V_{\text{ref}} = \begin{cases} \frac{V_{\max}+V}{2}c, & \text{if } V < V_{\max} \\ V_{\max}c, & \text{else} \end{cases} \tag{4-3a}$$

$$V_{\text{ref}} = V_{\max}c, \quad \text{if } V_{\text{ref}} > V_{\max} \tag{4-3b}$$

4-3 Reference generator

The Reference Generator uses the two reference signals computed by the PID-, and dynamic velocity controller to determine the vehicle equilibrium states. These states form the reference signal for the MPC, containing all algebraic, differential and control states. The grid filled with equilibria are pre-computed, as done in Section 3-2, and allow look-up of these reference vehicle equilibrium states. This lookup is done by interpolating between points on the grid, resulting in a smooth signal produced by the reference generator. This grid contains a finite set of points, without reference for $\kappa = 0[\text{m}^{-1}$, and with it $R = \infty[\text{m}]$. Therefore an extra set of equilibrium values is computed for $R = 10^{99}[\text{m}]$. This allows for a smoother transition when κ_{ref} transitions from positive to negative values, and vice versa.

$$x^{\text{eq}} = \begin{bmatrix} \kappa^{\text{eq}} & V^{\text{eq}} & v_x^{\text{eq}} & v_y^{\text{eq}} & r^{\text{eq}} & \omega^{\text{eq}} & \delta_{\text{swa}}^{\text{eq}} & T_{\text{net},r}^{\text{eq}} \end{bmatrix}' \tag{4-4}$$

The reference signal generated is added at the end of the prediction horizon at each timestep, which requires a prediction of the distance traveled along the s_s-axis at the end of the horizon. The N^{th} κ_{ref} value is found by integrating the current velocity over the horizon length. This approximates the location along the s_s-axis and will be sufficiently accurate as long as the velocity doesn't change too much.

The multiplicity of equilibria planes generated for the non-convex tyres implicate a choice to be made between 'normal' driving and drifting. A high-level controller can determine which equilibria to look-up for the reference $R - V$ pair, however this is not further analyzed in this research.

These acquired reference signals are sent from the reference generator to the MPC controller, which determines the actual control signals needed to direct the vehicle towards the desired equilibrium states.

4-4 MPC formulation

Linear and non-linear MPC techniques have already been addressed to vehicle dynamics control in the application of autonomous drifting [27, 9]. Though some argue that this method is too computational intensive it has proven its effectiveness in terms of the ability to achieve steady-state drifting. The system is required to generate a control input and provide this to the vehicle. By predicting the future states according to the differential equations of a model one can construct a control signal which is optimal to steer the model towards the desired system states. This MPC strategy computes these control inputs by solving the OCP. Automatic Control and Dynamic Optimization (ACADO) provides a code generation toolkit to construct a solver for the OCP which in its turn determines the control inputs for the system.

4-4-1 OCP

The OCP is described as a normal discretized linear-least squares problem. This Quadratic Problem (QP) can normally be solved using various algorithms, in this analysis Sequential Quadratic Programming (SQP) is applied, which means that the problem is solved sequentially and approaches the optimum by iteration. The constraints of this problem are the system dynamics and a set of bounds to the control inputs.

$$J = \sum_{k=0}^{N-1} \left\| \begin{matrix} x_k - x_k^{\text{eq}} \\ u_k - u_k^{\text{eq}} \end{matrix} \right\|_W^2 + \|x_N - x_N^{\text{eq}}\|_{W_N}^2 \tag{4-5a}$$

$$\min \quad J \tag{4-5b}$$

$$\text{s.t.} \quad \begin{cases} x_{k+1} = f(x_k) + g(x_k, u_k) \\ y = h(x_k) \\ u_{\text{lb}} < u_k < u_{\text{ub}} \end{cases} \tag{4-5c}$$

The system states are chosen as $x = [\kappa \quad V \quad v_x \quad v_y \quad r \quad \omega_r]$, the control inputs as $u = [\delta \quad T_{\text{net},r}]$, and the output signals of the system are chosen as $y = [v_x \quad v_y \quad r \quad \omega_r]$. With this set of states describing the Differential-Algebraic set of Equations (DAE) it is possible to choose the weighting matrices in such a manner that emphasis is placed on the two algebraic states (κ and V) to steer the vehicle motion to tracking behaviour of the predefined path.

The horizon N tells us how many steps we are predicting future state, to be able to solve this optimal control problem the continuous time dynamics are to be discretized along this prediction horizon. The method used to do this is by integrating the continuous time dynamics and with it find the system states at each timestep along the horizon. The ACADO toolkit allows for various integration methods to do this.

4-4-2 Integration methods

The MPC strategy operates in the discrete time domain while the dynamical equations describing the system are given in the continuous time domain. Thus to predict the future states the differential equations are integrated over time to find the predicted values hereof. Integration can be done in various methods both explicit and implicit. ACADO provides a variety of methods whom largely based on the Runge-Kutta (RK) methods, for explicit RK methods the general description is as follows;

$$\begin{aligned}
y_{n+1} &= y_n + h\sum_{i=1}^{s} b_i k_i && \text{(4-6a)}\\
\text{Where} & && \\
k_1 &= f(t_n, y_n) && \text{(4-6b)}\\
k_2 &= f(t_n + c_2 h, y_n + h(a_{21}k_1)) && \text{(4-6c)}\\
k_3 &= f(t_n + c_3 h, y_n + h(a_{31}k_1 + a_{32}k_2)) && \text{(4-6d)}\\
&\vdots && \\
k_s &= f(t_n + c_s h, y_n + h(a_{s1}k_1 + a_{s2}k_2 + \dots a_{s\ s-1}k_{s-1}) && \text{(4-6e)}
\end{aligned}$$

The parameter s determines the integration order. Choice of the exact method requires definition of the parameters s, $a_{i,j}$, b_i and c_i for $i \in [0, 1, ..., s]$. These parameters can be represented in a Butcher tableau (Tab: 4-1). The integrating technique used in this analysis is an implicit RK Radau IIA method of order three (Tab: 4-2). Implicit RK methods have the form of;

$$\begin{aligned}
y_{n+1} &= h\sum_{i=1}^{s} b_i k_i && \text{(4-7a)}\\
\text{Where} & && \\
k_i &= f\left(t_n + c_i h,\ y_n + h\sum_{j=1}^{s} a_{ij}k_j\right), \quad \text{for } i, j \in [1, ..., s] && \text{(4-7b)}
\end{aligned}$$

Table 4-1: General Butcher tableau

c	A
	b^T

Table 4-2: RK Radau IIA3 tableau

1/3	5/12	-1/12
1	3/4	1/4
	3/4	1/4

4-4-3 Solving the OCP

As the OCP is the most computationally intensive part of the controller a series of techniques is implemented into the ACADO software to reduce the computational demand. The SQP algorithm iterates the QP multiple times using Newton's method in a Real-Time-Iteration (RTI) scheme [28]. For this analysis, the Gauss-Newton Hessian approximation method is applied to decrease computation time. Multiple shooting allows for the finite-time optimization horizon to be divided into multiple segments. The discretized system then undergoes condensing to construct a smaller-scale dense problem from the large and sparse QP. This sparse problem is then given to the chosen solver, in this analysis the *qpOASES* solver is taken which employs the Online Active SEt Strategy (OASES) [29]. An alternative for the ACADO toolkit to solve the OCP is the FalcOpt toolkit [30], however, experimentation with this toolkit quickly showed that it was not able to match the speed with which the ACADO toolkit was able to solve the optimization problem. Besides using *qpOASES* as solver ACADO is also capable of using *qpDUNES* or *ForcesPro* as solver. These solvers have not been tested in this thesis as the *qpOASES* was able to quickly solve the OCP.

At each timestep of 10[ms] the horizon of the OCP shifts with an equal step, demanding a new of N^{th} equilibrium state-set to be added to the optimization horizon. Instead of solving the entire SQP only one iteration is performed, this RTI method allows for faster computation while allowing the QP to converge to the optimum over the period of a few timesteps.

Performance of the solving algorithm can be measured using two indicators. First from the Karush-Kuhn-Tucker (KKT)-value, one can determine if an 'optimal solution' is found and whether this point is a KKT optimal point as described in [31, 32]. As the RTI scheme sets a fixed amount of SQP iterations there is no tolerance for which the algorithm stops and therefore no evaluation is done whether a KKT criterion is met. One can therefore only say that performance is *better* when a lower KKT value is found, indicating that there is less fluctuation from the desired state. This value is commonly measured by taking the $\log_{10}$(KKT-value).

As a second performance indicator, the execution time t_{exec} required for the solver to perform a SQP step and with it compute the desired control signal is measured. When one would implement this control strategy onto an embedded system this would mainly decide the frequency at which could be run. Having a lower execution time than the discrete timestep set for the OCP is therefore seen as an acceptable performance.

4-5 Summary

A control strategy for which a MPC controller forms the basis is designed in this chapter. The reference states are pre-computed as done in the previous chapter. Using a dynamic reference velocity and a path following PID controller to determine the reference curvature allows generation of a reference trajectory for the MPC controller. The MPC requires solving of the OCP which is described in a discrete manner. To find this discrete state description the differential equations describing the vehicle dynamics can be integrated using one of the described RK-based integration methods available for the ACADO toolkit. The OCP can be solved using various solvers, wherein this analysis the *qpOASES* solver is used. To measure performance of the solving algorithm two performance indicators are used namely the $\log_{10}$(KKT-value) and the required execution time t_{exec}. These quantities indicate if solving can be done within the desired and in which degree the solution computed is optimal. This describes the groundwork for the control system, and the next task is to test this control strategy and evaluate its performance.

Chapter 5

Control System evaluation

Evaluation of the control scheme is done by analyzing vehicle behavior and algorithm convergence during various maneuvers. Assumed is that the track is known beforehand, and is described by a reference curvature κ_{ref}. Each track is composed of a small straight track segment, followed by a curvature. This straight segment allows the controller to initiate the solving process before being demanded to take a corner.

Simulation settings and performance indicators

The system is simulated using MATLAB & Simulink 2018b running macOS 10.14.2 using a 2,6 GHz Intel Core i7 processor with 16 GB 2133 MHz LPDDR3 memory and a Radeon Pro 450 2048 MB graphics card.

The optimization is run using a 100-step horizon with 10[ms] steps, thus with a prediction horizon of 1 second. The desired rate at which the optimization algorithm must be run to allow for shifting of the horizon by one step per optimization is thus 100Hz, desiring the execution time below 10[ms].

Each simulation regarding the analysis of the computational time required for the solver in this chapter is repeated 10 times to analyze how the optimization algorithm performs. The convergence to the reference states is equal for each repetition, however, the time required for this computation varies.

The vehicle simulated requires a set of parameters for system dynamics. These parameters are taken from a Porsche 911 Carrera S, obtained from the *IPG Carmaker* software. The Magic Formula (MF) parameters are taken from Tyre 4 (Figure 5-1), simulating behaviour of a low-friction surface. Assumed is that the model used by the Model Predictive Control (MPC) matches the simulated vehicle perfectly. Though this is not a realistic scenario, from this the ability of the control system to solve the Optimal Control Problem (OCP) and generate control signals which steer the dynamic model towards the desired equilibrium can be concluded. Besides this, the capabilities of the path-following Proportional Integral Derivative (PID) and the dynamic velocity controller can be evaluated.

Table 5-1: Vehicle parameters

Symbol	Value
m	$1593.1[\mathrm{kg}]$
l_f	$2.383[\mathrm{m}]$
l_r	$2.43[\mathrm{m}]$
I_z	$2575.9[\mathrm{kg\ m}^2]$
r_w	$0.508[\mathrm{m}]$
I_w	$3.916[\mathrm{kg\ m}^2]$

Table 5-2: Parameters for MF tyre 4

Parameter	Value
B	1.5289
C	1.0901
D	0.6
E	-0.9508

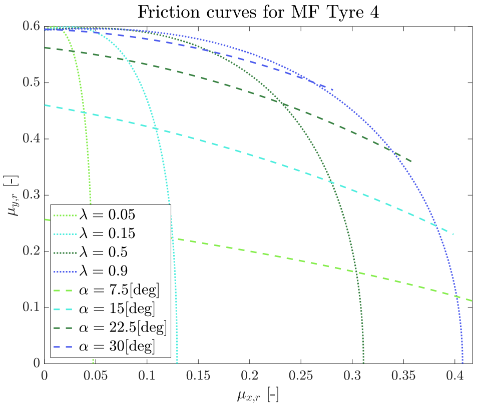

Figure 5-1: $\frac{1}{4}$ friction ellipse for MF tyre 4

5-1 MPC for tracking of drift equilibria

The MPC controller alone should be able to bring the vehicle into the desired reference state. To verify, this controller is tested for simulation of a vehicle on three different track types (described in Section 3-3-1). The first essential element for proper system response is the initiation of the drift, where after steady-state drifting is the goal. These maneuvers must be performed while the MPC controller is able to compute solutions to the OCP within the timestep length.

Due to the initialization of the controller, the reference signal is zero for the first prediction horizon, visible in Figure 5-2 describing the reference curvatures for three track segments. After each timestep, the reference signal is continuously updated with additional reference states at the end of the horizon.

A corner of a constant radius is initially considered. The curvature reference signal of the circle with continuous radius is determined by a reference-step change of magnitude $\frac{1}{R_{\text{ref}}}$.

The controller responds by exciting the steering wheel angle δ, forcing the vehicle to turn rapidly and initiate the drift. The vehicle is then guided to its reference states to achieve the desired steady-state velocity and curvature. Due to the fact that this step cannot be achieved instantaneously by the vehicle the intermediate time will result in an offset of the path, and with it, a lateral road error is generated, visible in Figure 5-2. For the directional transition, quite equal behavior is observed. Even with the zero-crossing of the reference signal, there is no exceptional decrease in computational performance.

Interesting is that for the clothoid track segment the path-following performance is best. Likely due to the much smaller initial reference step-change, the lateral deviation is much smaller and the track is well followed until the reference corner radius falls below the minimum due to the fixed desired velocity. The Karush-Kuhn-Tucker (KKT) value rises significantly due to the constant change of reference states. Computational demands from the controller can be observed in Figure 5-3 and show that the great majority hereof fall well beneath the timestep length. There are maximum computation time values that exceed the timestep length, bringing forward the question of what impact this would have on an actual implementation of the control system. The simulation environment only indicates this potential problem as it is able to simulate the system at a slower rate and thus doesn't provide an answer to the question.

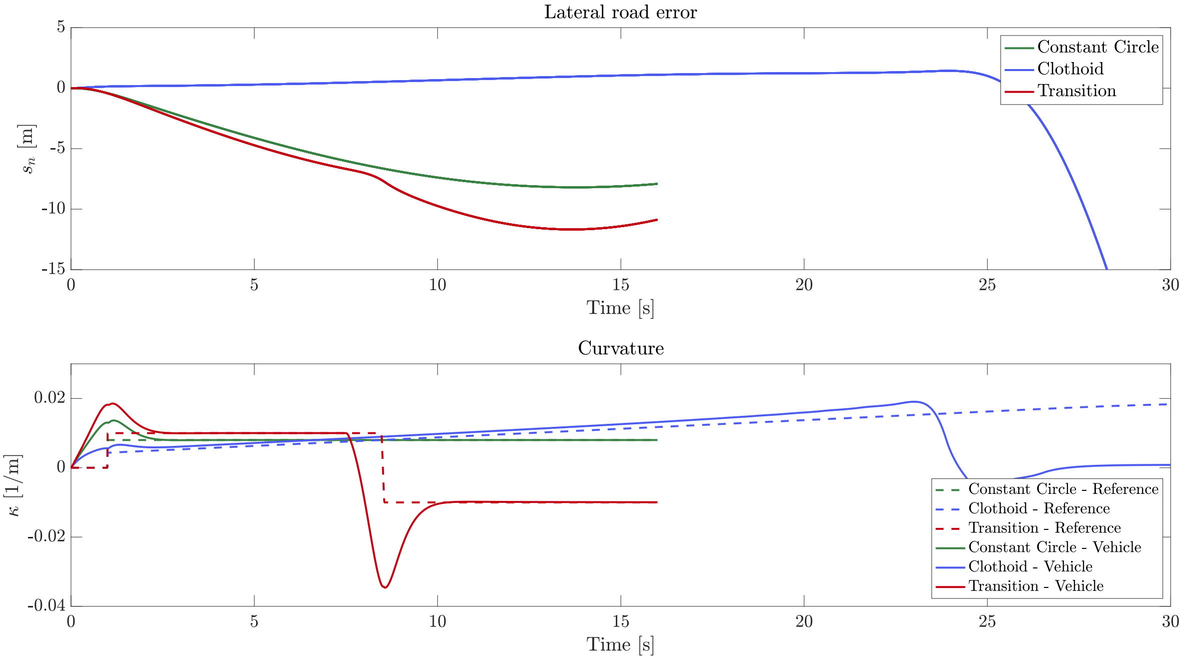

Figure 5-2: MPC performance

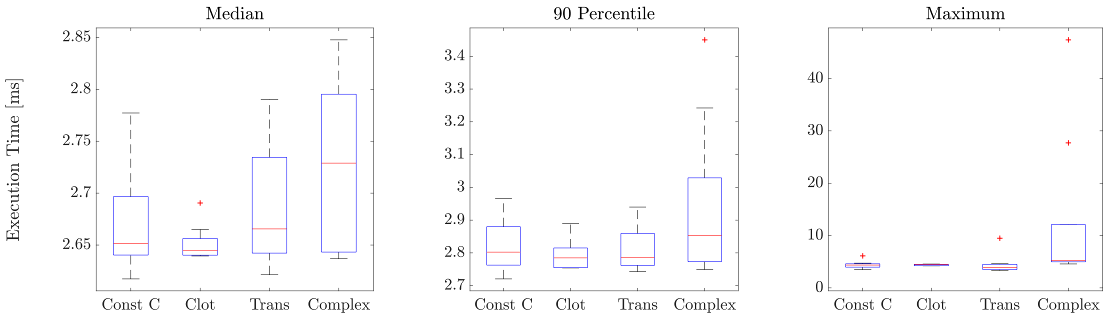

Figure 5-3: MPC execution time

Table 5-3: MPC convergence

MPC	$\log_{10}$(KKT-value)					RMSE
	Minimum	Median	90 percentile	95 percentile	99 percentile	s_n
Const C	-13.6555	-12.8598	-10.7228	-7.1826	-2.3394	6.051
Clothoid	-7.4132	-6.0359	-4.1670	-3.6373	0.0078	16.771
Transition	-12.8590	-8.3123	-4.9961	-2.2519	-0.5929	8.020
Complex 1	-15.1694	-9.1928	-6.1330	-5.1949	-0.1874	6.932

Initial condition variation

The simulations are done with the vehicle already having a certain initial velocity. However, the controller could be needed when the vehicle is at a different state, potentially further away from the desired equilibrium state. Figure 5-4 shows how the vehicle states behave when the controller is activated while the vehicle is at a range of initial states. The green circles indicate whether the controller is able to stabilize the vehicle as the optimization algorithm converges the system states towards the desired equilibrium. The states varied are v_x, v_y and r. However, the controller allowed convergence towards the desired equilibrium for each initial yaw rate, and therefore the phase-plane is portrayed in 2D. Convergence from initial velocities which are higher than the desired velocity shows much better result than when initiating from a lower velocity. This aspect was taken into account when designing the dynamic velocity controller.

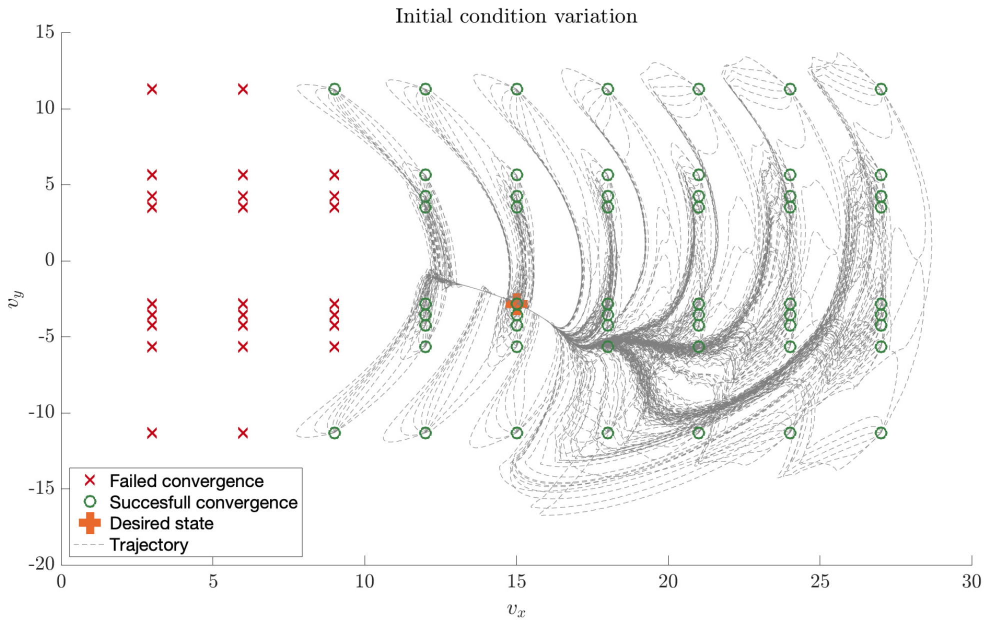

Figure 5-4: Initial condition variation for constant radius cornering

It can be concluded that the MPC controller is able to successfully initiate- and stabilize the drifting maneuver. The next task is to add the path-following ability to the control system.

5-2 Path following

The reference signal constructed from the desired curvature is a non-smooth signal, demanding instant switches between equilibrium states. However, as this is not possible, the path taken by the vehicle will deviate from the desired path. To steer the vehicle back onto the desired path the path following PID controller from Chapter 4-1 is added to the control system. The resulting change of the reference signal, which was constant before the introduction of the PID controller, is visible in Figure 5-5.

Introduction of this PID controller instantly shows improvement of the path following ability of the vehicle as the Root-Mean-Square Error (RMSE) value for lateral deviation has dropped significantly (Table 5-5). For behavior on the constant radius circle and directional transition track segments stabilization around zero lateral road deviation is achieved while converging to the desired equilibrium states. Impact on execution time is minimal and the ability of the algorithm to convergence drops slightly, due to the constant change of reference states. Even though this increase of $\log_{10}$(KKT-value) it can be concluded that the controller is able to successfully drift the vehicle for the desired road curvatures. Analyzing the computational impact of this varying reference curvature on the MPC, visible in Figure 5-6, shows that there is little difference though slightly reduced computation time required.

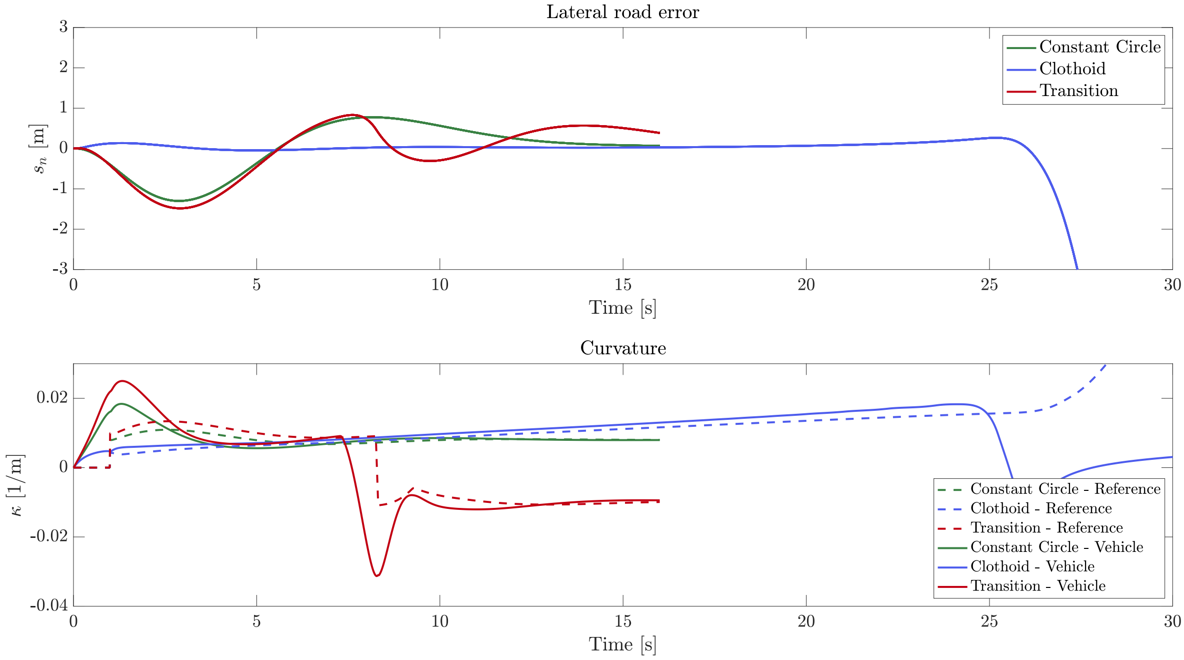

Figure 5-5: MPC|PID performance

5-3 Dynamic velocity driving

After the change in reference curvature, the reference vehicle velocity is also dynamically set. This maximum allowable velocity for the reference curvature is determined according to (4-3a). For the constant radius circle and directional transition track segments, the vehicle's track

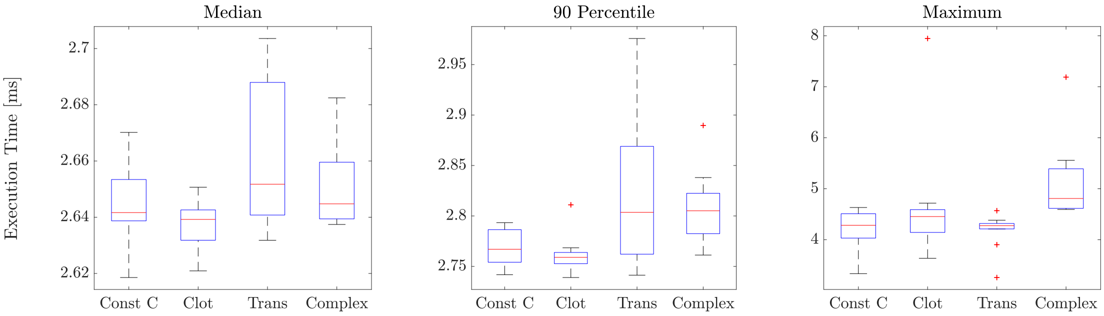

Figure 5-6: MPC|PID execution time

Table 5-4: MPC|PID convergence

MPC|PID	log$_{10}$(KKT-value)					RMSE
	Minimum	Median	90 percentile	95 percentile	99 percentile	s_n
Const C	-12.8324	-7.6314	-3.7475	-2.7347	-1.7090	0.622
Clothoid	-7.2267	-5.3909	-3.4678	-2.9351	0.6758	12.288
Transition	-10.0550	-5.0681	-1.4290	-0.9586	0.5767	0.682
Complex 1	-14.5729	-9.1099	-3.8206	-2.6966	-0.2461	0.389

following performance shows practically no change (Figure 5-7). For the clothoid segment, one can observe two phenomena. Where first the lateral deviation was close to zero at the start of the track there is a significant increase while using the dynamic velocity reference signal. However, the other observation is that the vehicle is able to pass the point where previously the reference cornering radius became too small, thereby allowing a greater range of corners able to be taken. The RMSE values describing the path following ability all fall below 1[m] indicating that the control system is able to successfully drift a wide range of curved paths and follow the desired path. The computational impact of the added dynamic velocity controller shows that there is a slight increase in computational demand though generally, the required time for these computations falls below the timestep length.

Table 5-5: MPC|PID|DV convergence

MPC|PID|DV	log$_{10}$(KKT-value)					RMSE
	Minimum	Median	90 percentile	95 percentile	99 percentile	s_n
Const C	-6.2400	-4.8801	-3.7252	-3.1468	-1.2552	0.571
Clothoid	-8.1852	-4.2423	-0.9064	6.7330	9.6778	0.900
Transition	-8.6850	-4.8339	-1.7857	-0.8055	0.8503	0.686
Complex 1	-10.7984	-4.6713	0.6551	1.7604	4.0072	0.730

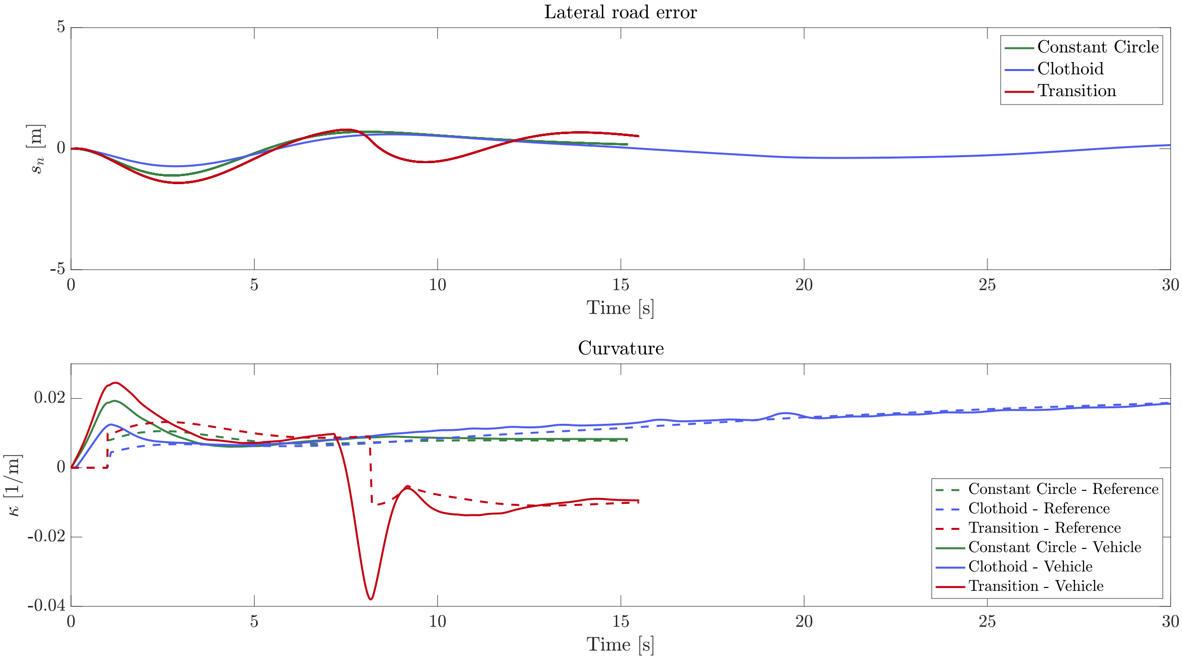

Figure 5-7: MPC|PID|DV performance

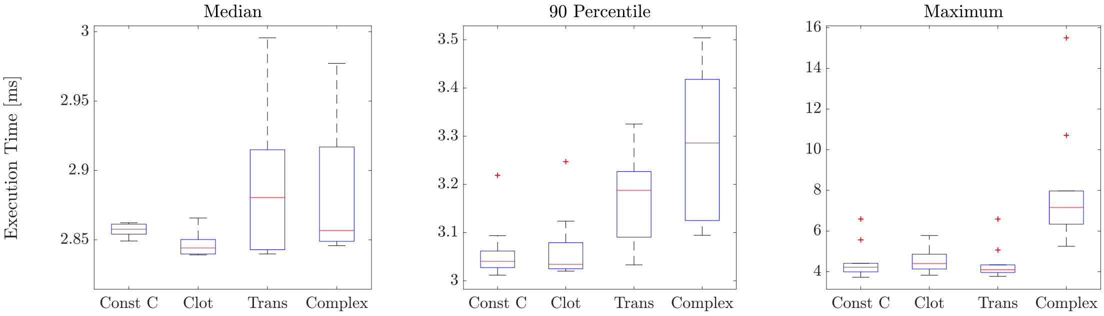

Figure 5-8: MPC|PID|DV execution time

Computational impact of various integrators

There are a large number of integrators which can be used by the Automatic Control and Dynamic Optimization (ACADO) controller to discretize the continuous time system. These integrators vary in order/complexity and method, though all family of the Runge-Kutta (RK) integration methods. Figure 5-9 shows how the execution times of the controller compare when using these different integrators for different controller combinations described in Table 5-6. Each scenario is run 10 times to get a good impression of the execution times. It can immediately be concluded that the implicit integration methods require much more computation time (up to 2x) when compared to the explicit RK method. Convergence of the algorithm does not vary greatly between integrators, indicating that it does not improve performance when taking a higher order integrator. Also, the lateral deviation of the drifting vehicle does not change greatly when varying between integration methods. The best choice would be to take the simplest integration method (RK-12, which is the forward Euler method), however further in this analysis, the implicit Radau IIA of order 3 (RIIA3) integrator is used.

Table 5-6: Integrator and controller scenario index

	Runge-Kutta				Radau IIA			Gauss-Legendre			
	1/2	2/3	4/5	7/8	1	3	5	2	4	6	8
MPC	1	4	7	10	13	16	19	22	25	28	31
MPC|PID	2	5	8	11	14	17	20	23	26	29	32
MPC|PID|DV	3	6	9	12	15	18	21	24	27	30	33

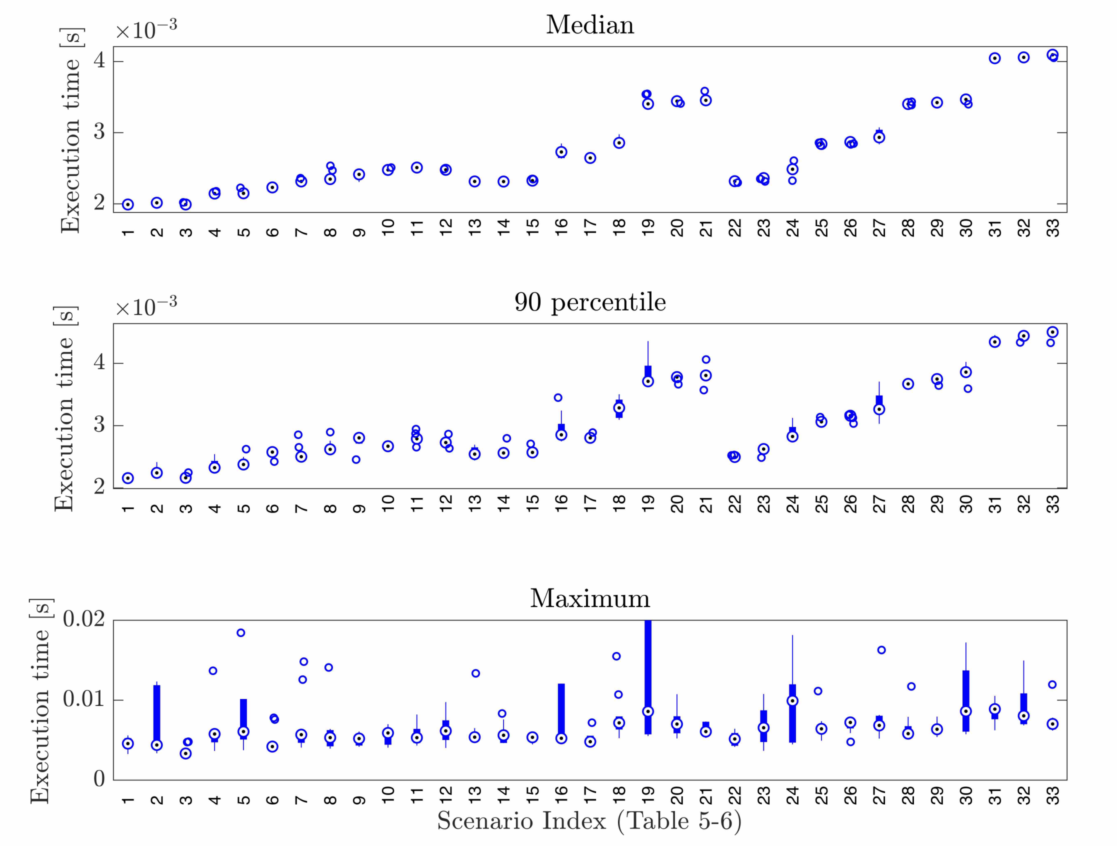

Figure 5-9: Computational impact of integrators

Table 5-7: Convergence for various integrators (Index Table 5-6)

	$\log_{10}$(KKT-value)					RMSE
Index	Minimum	Median	90 percentile	95 percentile	99 percentile	s_n
1	-15.041	-9.212	-6.052	-5.168	-0.151	6.880
2	-14.591	-9.066	-3.785	-2.661	-0.248	0.386
3	-10.881	-4.532	0.689	1.825	3.903	0.707
4	-15.172	-9.237	-6.100	-5.163	-0.149	6.932
5	-14.573	-9.118	-3.805	-2.678	-0.244	0.389
6	-10.967	-4.565	0.596	1.760	3.864	0.667
7	-15.169	-9.237	-6.099	-5.163	-0.149	6.932
8	-14.573	-9.117	-3.805	-2.678	-0.244	0.389
9	-10.812	-4.666	0.639	1.772	3.984	0.728
10	-15.169	-9.237	-6.099	-5.163	-0.149	6.932
11	-14.573	-9.117	-3.805	-2.678	-0.244	0.389
12	-10.833	-4.672	0.640	1.768	4.001	0.729
13	-15.293	-9.261	-6.145	-5.149	-0.174	6.959
14	-14.554	-9.167	-3.834	-2.700	-0.240	0.392
15	-11.042	-4.671	0.650	1.825	3.893	0.682
16	-15.169	-9.193	-6.133	-5.195	-0.187	6.908
17	-14.573	-9.110	-3.821	-2.697	-0.246	0.388
18	-10.798	-4.671	0.655	1.760	4.007	0.730
19	-15.169	-9.237	-6.099	-5.163	-0.149	6.932
20	-14.573	-9.117	-3.806	-2.678	-0.244	0.389
21	-10.812	-4.662	0.646	1.773	3.974	0.730
22	-15.168	-9.237	-6.099	-5.163	-0.149	6.932
23	-14.573	-9.117	-3.806	-2.678	-0.244	0.389
24	-10.832	-4.661	0.637	1.798	3.961	0.735
25	-15.169	-9.237	-6.099	-5.163	-0.149	6.933
26	-14.573	-9.117	-3.805	-2.678	-0.244	0.389
27	-10.799	-4.666	0.657	1.763	3.990	0.730
28	-15.169	-9.237	-6.099	-5.163	-0.149	6.932
29	-14.573	-9.117	-3.806	-2.678	-0.244	0.390
30	-10.797	-4.665	0.656	1.760	3.995	0.730
31	-15.169	-9.237	-6.100	-5.163	-0.149	6.933
32	-14.573	-9.117	-3.805	-2.678	-0.244	0.389
33	-10.816	-4.664	0.630	1.749	3.979	0.728

Complex track 2

With the implementation of the path following and the dynamic velocity properties onto the control system the controller can be tested on the second complex track (Figure 3-6). There are a few things to be noticed in this figure, namely that there is an oscillatory behavior present at each switch of cornering radius, something that can be tuned for different response by altering the PID path following controller and the weighting matrices of the MPC controller. However, this tuning is sensitive to the vehicle velocity and the desired driving curvature. It would be a nice improvement to implement a dynamic weighting matrix strategy for this and also a nonlinear PID strategy. Also, there can be spikes observed in the reference velocity, which arise due to the fact that the switch from one direction to the other does not change the reference curvature instantaneously to the other value but MATLAB interpolates this into a continuous signal. Therefore when this curvature approaches the zero crossing, the reference velocity shoots upward as low curvature corners have a high allowed maximum velocity. The controller allows the vehicle to follow the desired trajectory with a lateral deviation smaller than 3[m] which is interpreted as a good result regarding the vehicle velocity.

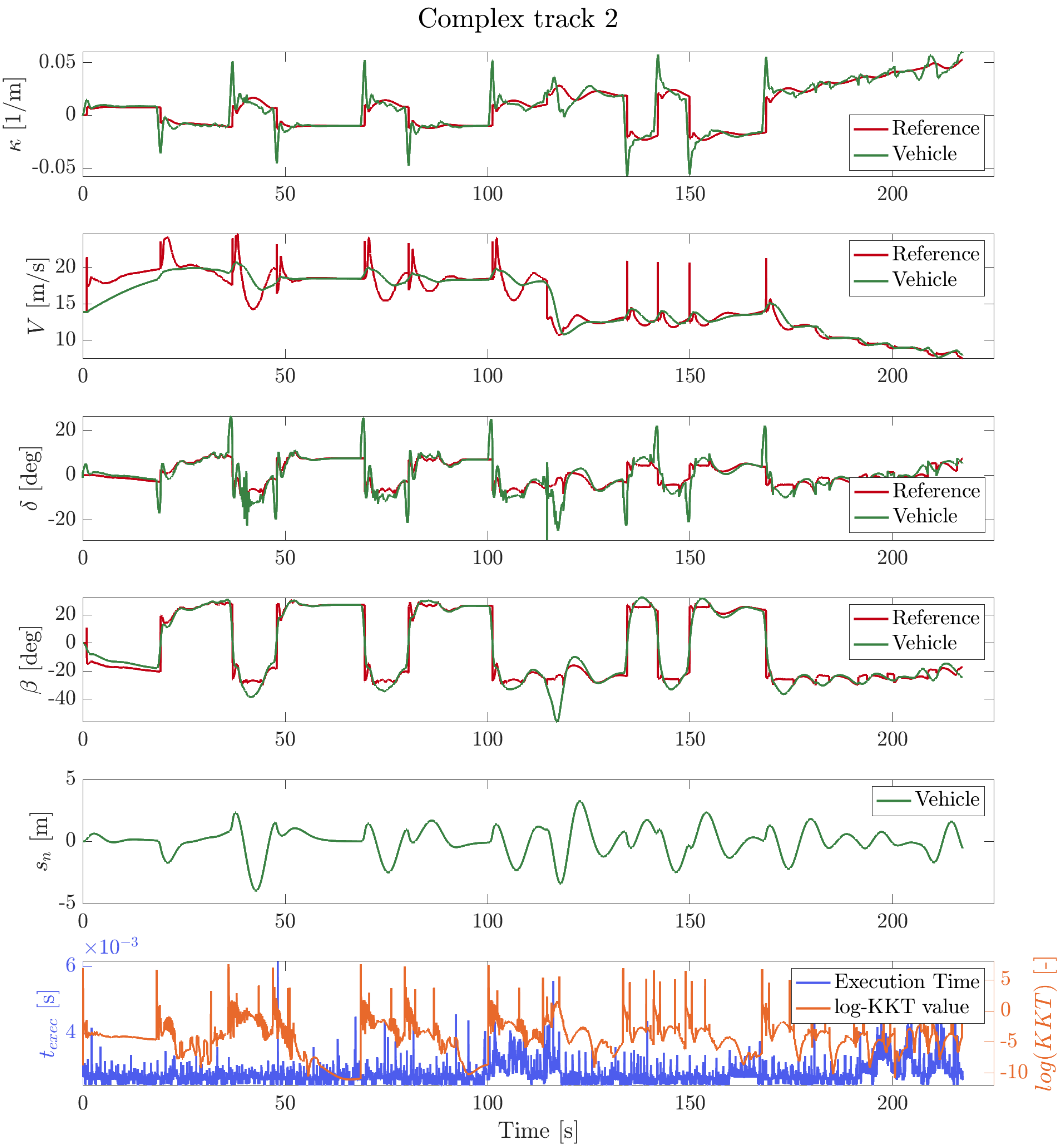

Figure 5-10: Simulating controller on complex track 2

5-4 Summary

The MPC controller is able to steer the states towards the desired equilibrium values. However, the transition towards this equilibrium results in the vehicle deviating from the desired trajectory. The controller is able to solve the OCP in a fraction of the desired length showing that this controller is feasible. The convergence of the algorithm shows that it is able to converge towards the desired equilibria. Interesting is to see that the constantly changing reference signal of the clothoid track results in a 'worse' algorithm convergence. The MPC controller shows robustness when activated in cases where the vehicle velocity is larger than the equilibrium value. Though, for velocities smaller than the desired equilibrium it fails to converge to an optimal solution. The path following PID controller allows for a dynamic curvature while also including a dynamic reference velocity allows for tracking of the desired paths. With the reference trajectory constantly changing the algorithm has a more difficult task to find the solution to the OCP. With this 'worse' optimality, the controller is still able to control the vehicle to achieve the autonomous drift for all tracks. The choice of integrator leads to believe that the simplest integration method available is the best option. Deviation from the desired path is hereby not influenced much, and the ability for the algorithm to converge also stays generally equal while reducing computation times to the minimum. The control system designed is thus able to successfully control the vehicle and steer the states towards the desired equilibria. The next step towards implementation is to see if this is applicable to a 1:10 scaled vehicle. This is done by simulating this scenario, and with it identify challenges and limitations which arise.

Chapter 6

Feasibility analysis for experimental validation

This simulation environment is built in Matlab/Simulink, allowing experimentation with the control system and bridge the distance to potential experimental validation. Though this implementation and validation is not part of this project, an analysis is made to take the first step in this direction. Certain challenges arise when moving from simulation to implementation. These challenges are discussed, attempted to be recreated, and, overcome in this chapter.

6-1 Delft Center for Systems and Control (DCSC) Experimental platform

The Network Embedded Robotics DCSC lab (NERDlab) contains an experimental platform on which autonomous driving can be tested. Inside this lab lies a Motion Capture 'arena' provides position tracking inside a contained area. This tracking data is sent to a computer which runs Robotic Operating Software (ROS), able to read this data and communicate between a remote-controlled scaled vehicle and other local software e.g. Matlab/Simulink. This remote controlled scaled vehicle equipped with various sensors can then be controlled via ROS.

6-1-1 Scaled vehicle

The platform on which vehicle dynamics controller can be tested is a 1:10 scaled remote controlled vehicle. The dynamics describing this vehicle motion are roughly the same, though due to the difference in dimensions the vehicle behaves differently. The parameters describing the vehicle and it's tyres can be found in Table 6-1, Table 6-2 and Figure 6-1.

Table 6-1: Scaled vehicle parameters

Symbol	Value
m	2.90 [kg]
l_f	0.129 [m]
l_r	0.129 [m]
I_z	0.04 [km m^2]
r_w	0.029 [m]
I_w	0.0004 [kg m^2]

Table 6-2: MF parameters for scaled vehicle tyre

Parameter	Value
B	0.710
C	1.057
D	0.494
E	-0.2

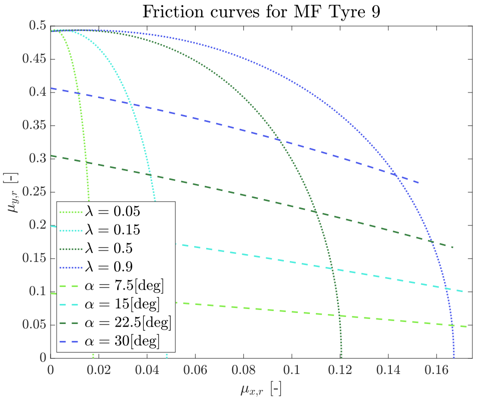

Figure 6-1: $\frac{1}{4}$ friction ellipse for scaled vehicle tyre

With the new parameters describing the scaled vehicle, it is possible to compute the equilibrium solutions as done in Section 3-2-2. This results in a grid of equilibrium points for the scaled vehicle, shown in Figure 6-2. The equilibrium grid shows behavior similar to the grid computed for the full-scale vehicle. Steering angles decrease greatly when higher radius corners are taken, while for increasing velocity these steering angles are reduced further to reach negative values. For these negative steering angles, the vehicle side-slip angles become highly negative, indicating drifting behavior.

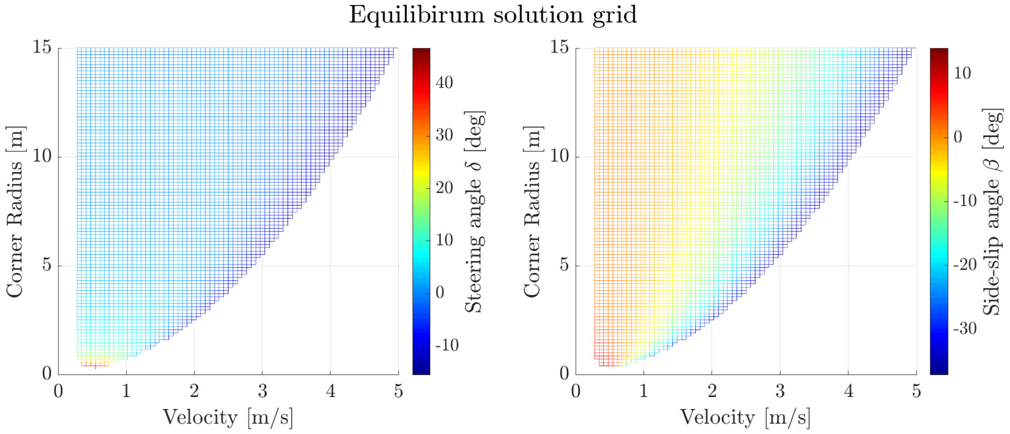

Figure 6-2: Equilibrium solutions for the scaled vehicle model

For the linearized 2-state bicycle model system described as in [22] one can determine the poles (p_i) of the system as in (6-1), deducted from a linearized two-state bicycle model ($x = [\beta \quad r]$).

$$
\begin{aligned}
p_{1,2} &= -\frac{1}{2}\left(c_1 \pm \sqrt{c_1^2 + 4c_0}\right) && \text{(6-1a)}\\
c_0 &= \frac{l_f C_{y,f}}{I_z} && \text{(6-1b)}\\
c_1 &= \left(\frac{l_f^2 C_{y,f}}{I_z v_x^{\text{eq}}} + \frac{C_{y,f}}{m v_x^{\text{eq}}}\right) && \text{(6-1c)}
\end{aligned}
$$

The lateral front-tyre cornering stiffness $C_{y,f}$ can be determined by taking the tangent of the Magic Formula (MF) curve describing the relation between $F_{y,f}$ and the slip angle α_f using the differential equations (2-5),(2-8),(2-9) and the state measurements. Having determined this quantity it is possible to find the poles of the system dynamics as done in [22] for each drifting equilibria. These poles are used further when investigating parametric uncertainty of the model, inaccuracies when measuring the vehicle parameters or estimating tyre parameters.

6-1-2 Controller adjustment

To redesign the controller the new vehicle and tyre-parameters are used to redefine the differential equations. These differential equations are used to compute a new set of driving equilibria and define the model used by the Model Predictive Control (MPC) strategy.

Each of the controllers is re-tuned for the system to achieve stable drifting behavior on the three test-track types. This includes a new weighting matrix, re-tuning the PID-controller and new coefficients for the Dynamic Velocity controller. General controller performance can be seen in Figure 6-3.

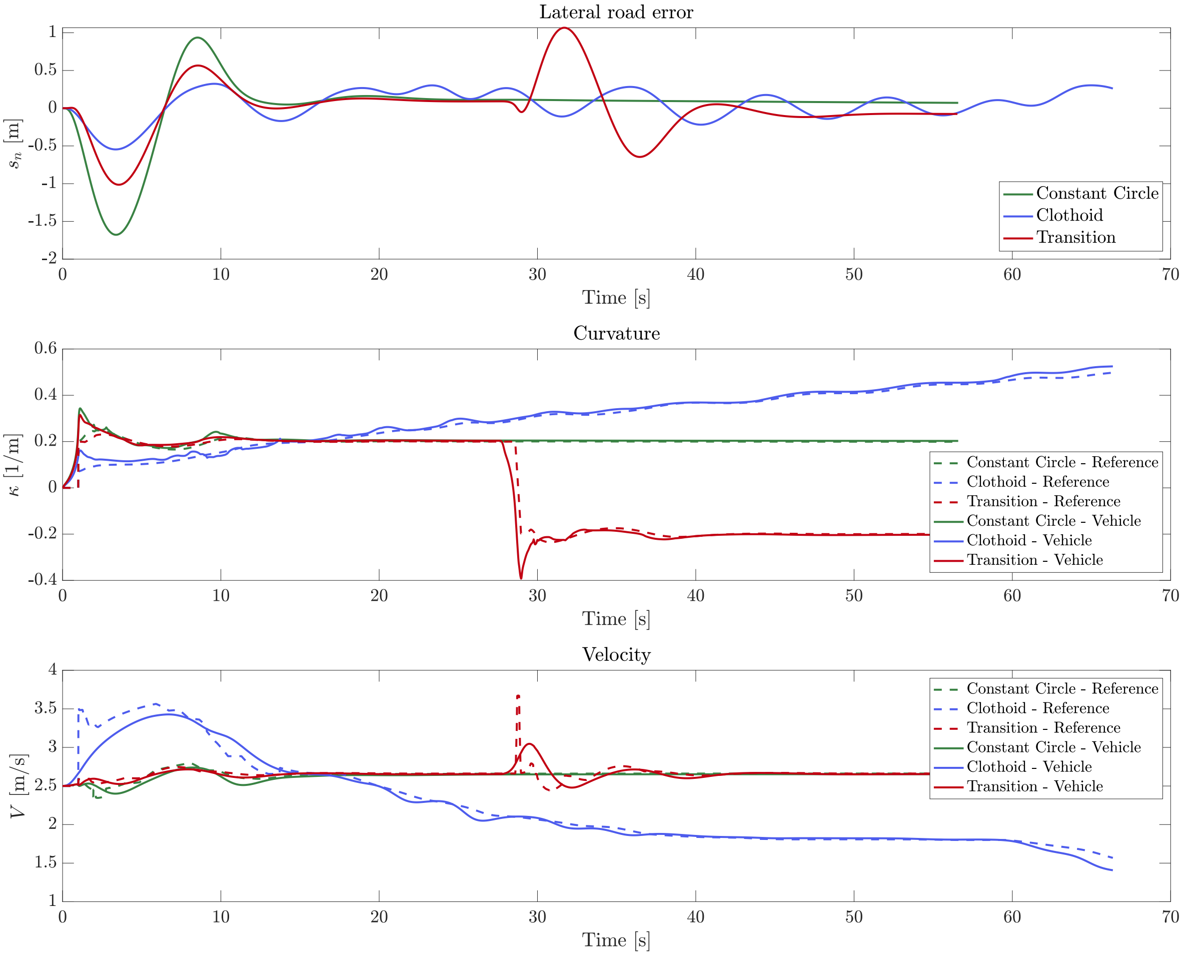

Figure 6-3: Scaled vehicle - general controller performance

The behavior of the scaled vehicle shows oscillation before settling to the reference curvatures. The lateral deviation of the vehicle with respect to the desired path has become relatively a much higher value as the reference radius of the corner equals 5[m] and the lateral deviation overshoots initially to 1.5[m]. Furthermore, the controller is able to stabilize the vehicle around the desired equilibrium well.

6-2 Uncertainties and imperfections

As the real world does not behave exactly as modeled, these non-ideal factors are to be accounted for. To test whether the designed control system would still perform when confronted with these factors, these non-ideal factors are simulated. The measurements done to identify the system states does not happen instantaneously and without error. The delay and noise acting on the system are analyzed and recreated.

6-2-1 Modeling of delay and noise

Process delay

A Bachelor Thesis group working on the implementation for autonomous drifting has identified the delay acting on the experimental platform, quantifying this as in 6-3. Computation of control signals was not done onboard the remote-controlled vehicle but on a PC connected to the Motion Capture (MoCap) arena. The delay between the measurement of the vehicle states and receiving this on the PC is defined as L_{wireless}. The delay between the measurements done by the MoCap system and receiving the information on the PC is given by L_{wired} and the latency between the computation of the control signal and the actuation of the vehicle is defined as L_{control}.

Table 6-3: Measured latency in the NERDlab

Parameter	Mean	σ	95% confidence interval
L_{wireless}	239 ms	27 ms	[184 ; 293] ms
L_{wired}	131 ns	31 ns	[69 ; 193] ns
L_{control}	66.7 ms	-	-
Total latency	306 ms	27 ms	[252 ; 360] ms

This delay acting on the system simulated by placing a constant 250[ms] delay on the measurement of the states. Also, a delay of 65[ms] is placed on actuation of the steering wheel angle and application of torque on the rear wheel. These constant time delays should provide insight into the robustness of the control system to these delay sizes. Though not included in this research, further analysis on variable time delays could be done for more accurate simulation.

The drift controller is initially desired to achieve vehicle drifting in a constant radius circle. After the delay was introduced to the system the controller was unable to converge to a steady-state equilibrium, resulting in termination of the simulation. For the controller to be able to work all the controllers are re-tuned, and the performance of the new system can be seen in Figure 6-4. Interesting is that the performance of the controller after re-tuning has a much better path following performance than the response where there was no delay present. This better performance is most likely due to the much lower (16%) steady-state velocity, which also results in a lower side-slip angle.

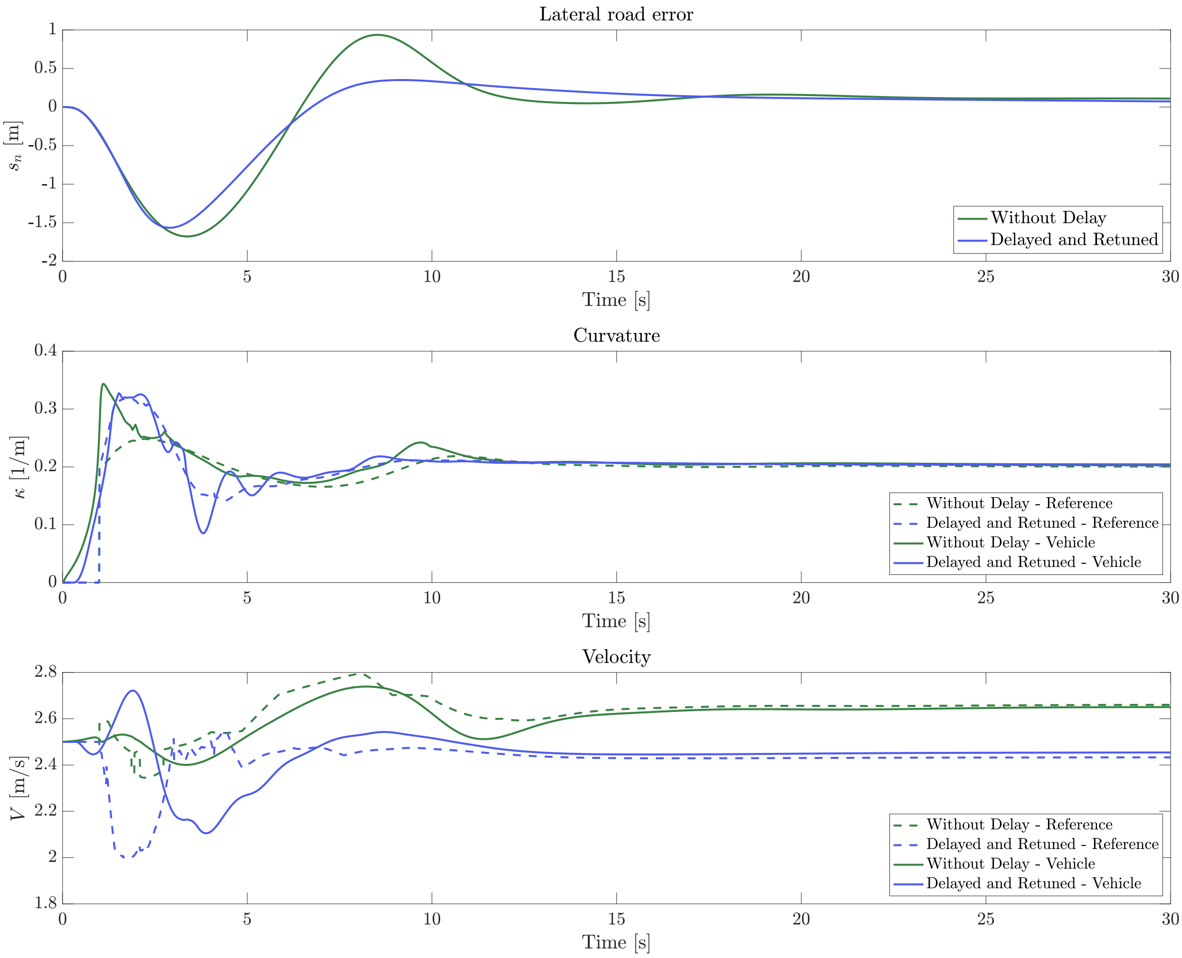

Figure 6-4: Results after introducing- and re-tuning for delay

Measurement noise

A second factor influencing the system is noise measured by the Inertial Measurement Unit (IMU), containing two measurement types to measure the longitudinal/lateral acceleration and yaw rate. The accelerations can be measured by accelerometers and the rotational velocities can be measured by the gyroscope. Some IMU's may contain a magnetometer to measure the magnetic field surrounding the body. In this case the longitudinal and lateral acceleration $\tilde{a}$ are measured (6-2) including additive zero-mean Gaussian noise η_a;

$$\tilde{a} = a^{(l)} + \eta_a, \quad \eta_a \sim \mathcal{N}(0, \sigma^2_{\text{acc}}) \tag{6-2}$$

The gyroscope is capable of measuring the yaw rate $\tilde{r}$ (6-3);

$$\tilde{r} = r + b + \eta_g, \quad \eta_g \sim \mathcal{N}(0, \sigma^2_{\text{gyro}}) \tag{6-3}$$

This is composed by the true yaw rate r, a bias b and additive zero-mean Gaussian noise η_g. The bias is temperature-dependant and may change over time though can be approximated as a constant. Even when the bias is known, integration of these measured values results in parameter drift.

The final state to be measured is the angular velocity of the rear wheel ω_r. Various methods to do this include a tachometer, or a laser source and patterned reflective surface on the wheel rims.

The DC-motors powering the wheels generate magnetic fields, this influences the signals generated by the IMU as the metal parts of which it is composed are acted upon. To account for this behavior it is possible to use an IMU which measures the magnetic field and use this knowledge to post-process the signal. Another possibility is to surround the IMU with a magnetic field fending material.

When using the MoCap system to identify vehicle velocity, taken trajectory and desired curvature assumed is that there is no noise and only a one-timestep-delay (10[ms]) present on the measurements and computation.

Analyzing the convergence of the algorithm indicates the ability of the vehicle to achieve stable sustained drifting. This shows that the increase of noise, results in more oscillations at 'steady-state', thus being less steady. The control input now generated by the controller is a much higher frequency oscillating signal. This signal is then sent to the system which may cause a problem as the actuator may not have the ability to handle such high rate oscillations of the steering angle.

The system copes much better with the variance of accelerometer data than to variance of the yaw rate. In this case, the longitudinal and lateral accelerations are directly integrated to obtain the state values v_x and v_y. Also, there is no bias on the accelerometer data, though this noisy signal still gives rise to parameter drift due to direct integration. To account for this unwanted situation a Kalman Filter is constructed to estimate the system states by measurement of the IMU and MoCap signals and (partially) eliminate the noise.

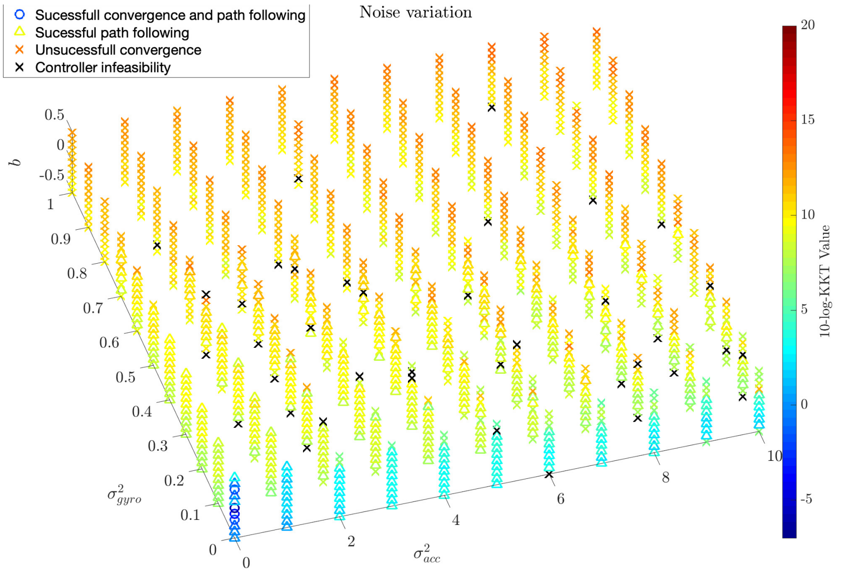

Figure 6-5: Algorithm convergence response to measurement noise

Filtering the measurement noise and state estimation

Many methods have been developed to estimate vehicle states and parameters [33, 34, 35]. In this analysis to filter the additive noise for this system an Extended Kalman Filter (EKF) is designed. This discrete model of system dynamics is obtained by a first order linearization of the continuous time bicycle model and by adding additive process and measurement noise terms w_k and v_k. As the IMU measures accelerations rather than velocities the state description of the system is extended to be $x = \begin{bmatrix} v_x & a_x & v_y & a_y & r & \omega \end{bmatrix}'$.

$$\begin{aligned} x_{k+1} &= f(x_k, u_k) + w_k, \quad w_k \sim \mathcal{N}(0,0) && \text{(6-4a)} \\ y_k &= h(x_k, u_k) + v_k, \quad v_k \sim \mathcal{N}(0, \sigma^2_{\text{acc/gyro}}) && \text{(6-4b)} \end{aligned}$$

To model various delays in the system gives rise to the state description (6-5).

$$\begin{bmatrix} x_{k+1+D_1} \\ \vdots \\ x_{k+2} \\ x_{k+1} \\ u_k \\ \vdots \\ u_{k-D_2+1} \end{bmatrix} = \begin{bmatrix} x_{k+D_1} \\ \vdots \\ x_{k+1} \\ f(x_k, u_k) + w_k \\ u_{k-1} \\ \vdots \\ u_{k-D_2} \end{bmatrix} \tag{6-5a}$$

The measurement functions $g(x_k, u_k)$ presents the output map of the IMU, MoCap system and the Wheel speed encoder. As the measurements done by the IMU and the MoCap systems are provided with a different (smaller) delay, their measurement can be incorporated in their

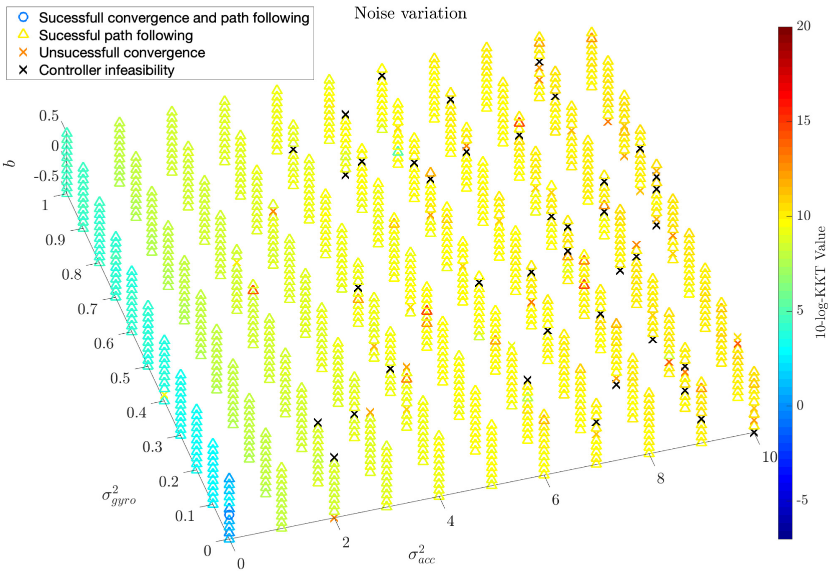

Figure 6-6: Algorithm convergence response to measurement noise after Kalman Filter introduction

(delayed) state. This gives us the following measurement functions;

$$y^{\text{IMU}}_{k+D_1} = C^{\text{IMU}} x_{k+D_1} + v^{(1)}_{k+D_1} \tag{6-6a}$$

$$y^{\text{MoCap}}_{k+D_3} = C^{\text{MoCap}} x_{k+D_3} + v^{(2)}_{k+D_3} \tag{6-6b}$$

$$y^{\text{Wheel}}_{k+D_4} = C^{\text{Wheel}} x_{k+D_4} + v^{(3)}_{k+D_4} \tag{6-6c}$$

The IMU is capable of measuring accelerations and rotations due to the accelerometer and gyroscope; ($C^{\text{IMU}} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}'$). The MoCap is capable of measuring the vehicle velocity and yaw rotation ($C^{\text{MoCap}} \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}'$). The angular velocity of the rear wheel is measured with a wheel speed encoder; ($C^{\text{Wheel}} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}'$).

Differentiating the state- and measurement functions along the states bring forward their respective Jacobian;

$$F_k = \left.\frac{\partial f(x,u)}{\partial x}\right|_{\hat{x}_{k-1|k-1}, u_k} \qquad H_k = \left.\frac{\partial h(x)}{\partial x}\right|_{\hat{x}_{k|k-1}} \tag{6-7}$$

The EKF estimates the states $\hat{x}$ according to the following recursive algorithm;

$$
\begin{array}{rcl}
 & \text{Predict} & \\
\hat{x}_{k|k-1} & = & f(\hat{x}_{k-1|k-1}, u_k) \\
P_{k|k-1} & = & F_k P_{k-1|k-1} F_k^T + Q_k \\
 & \text{Update} & \\
\tilde{y_k} & = & y_k - H(\hat{x}_{k|k-1}) \\
S_k & = & H_k P_{k|k-1} H_k^T + R_k \\
K_k & = & P_{k|k-1} H_k^T S_k^{-1} \\
\hat{x}_{k|k} & = & \hat{x}_{k|k-1} + K_k \tilde{y}_k \\
P_{k|k} & = & (I - K_k H_k) P_{k|k-1}
\end{array}
$$

Assumed is that $v^{(2)}$ and $v^{(3)} \sim \mathcal{N}(0,0)$, with $D_3 = 1$ and $D_4 = 0$. Here the MoCap measurement is delayed only one timestep to the process ($D_3 = 1$). Where the wheel speed encoder has direct measurement ($D_4 = 0$). The output of the Kalman filter produces state estimates for the controller to use, and using the state description previously mentioned it is possible to estimate each state, including x_{k+1}, making it possible to counteract on the delay brought forth by the IMU.

The Variance Accounted For (VAF) measures the similarity between two signals. If the VAF is 100% the signals are the same, and when they are different the VAF value will be lower.

$$
\text{VAF} = (1 - \frac{var(y - \hat{y})}{var(y)}) * 100\% \tag{6-9}
$$

The following four scenarios (Ref. 1-4) indicate variations of the defined measurement functions (6-6) simulated;

Table 6-4: Filtered measurements scenarios

Ref.	C^{MoCap}	D_1	D_3
1	$\begin{bmatrix}1 & 0 & 1 & 0 & 1 & 0\end{bmatrix}$	25	1
2	$\begin{bmatrix}1 & 0 & 1 & 0 & 0 & 0\end{bmatrix}$	25	1
3	$\begin{bmatrix}0 & 0 & 0 & 0 & 1 & 0\end{bmatrix}$	25	1
4	$\begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}$	25	1

Without the implementation of the filter, a large range of noise descriptions results in a noisy reference signal generated. The algorithm then doesn't converge properly, resulting in a large $\log_{10}$(KKT-Value), visible in Figure 6-5. Implementation of the Kalman Filter, using the information available from the IMU, MoCap (Scenario 2 from 6-4) and wheel speed encoder is simulated. This results in great range increase the controller is able to steer the vehicle successfully towards the desired equilibrium. This stabilizes the drifting motion for a great range of noisy signals, as can be seen in Figure 6-6.

Table 6-5: Scenario results

Ref.	VAF-value			$\log_{10}$(KKT-value)		RMSE
	v_x	v_y	r	Median	95p	s_n
1	99.9788	99.9936	99.9858	-5.50	3.54	0.531
2	99.9904	99.9964	83.6720	4.86	7.08	0.586
3	91.8800	82.9110	99.0615	12.69	15.15	0.564
4	95.4483	69.0882	88.7909	12.89	15.444	0.825

The results of the second scenario are portrayed in Figure 6-7. This image shows clearly that the noise acting on the system has a direct negative effect on the reference signals generated.

To further test the control system an IMU is chosen as case study [36]. The gyroscope of this sensor has a Power Spectral Density (PSD) of $0.01[\text{deg/s}/\sqrt{\text{Hz}}]$. Running at 100[Hz] this produces a noise output with a variance of $1.75 * 10^{-2}[\text{rad/s}]$. The accelerometer has a PSD of $300 * 10^{-6}[g/\sqrt{\text{Hz}}]$, producing a noise with a variance of $3 * 10^{-2}[\text{m/s}^2]$. This sensor is simulated in the remainder of this thesis to generate noise for the following simulations. The sensor-bias is ignored in these cases.

State estimation is found to be an important factor for the controller and algorithmic performance, especially as these states are also used by the reference generator. The noisy signal entering the reference generator also results in a noisy signal entering the MPC controller, which greatly reduces the $\log_{10}$(KKT-Value). This noisy signal also results in noisy control signals, which may impose problems when implementing the proposed strategy on actual actuators. Another problem which arises when implementing this is the computational demand of the filter. Though no actual indication is made in the quantity of time which this would cost extra this is something potentially restricting implementation on a scaled vehicle. Besides measurement, inaccuracy of the states further differences between the ideal vehicle model and the actual scaled vehicle are discussed.

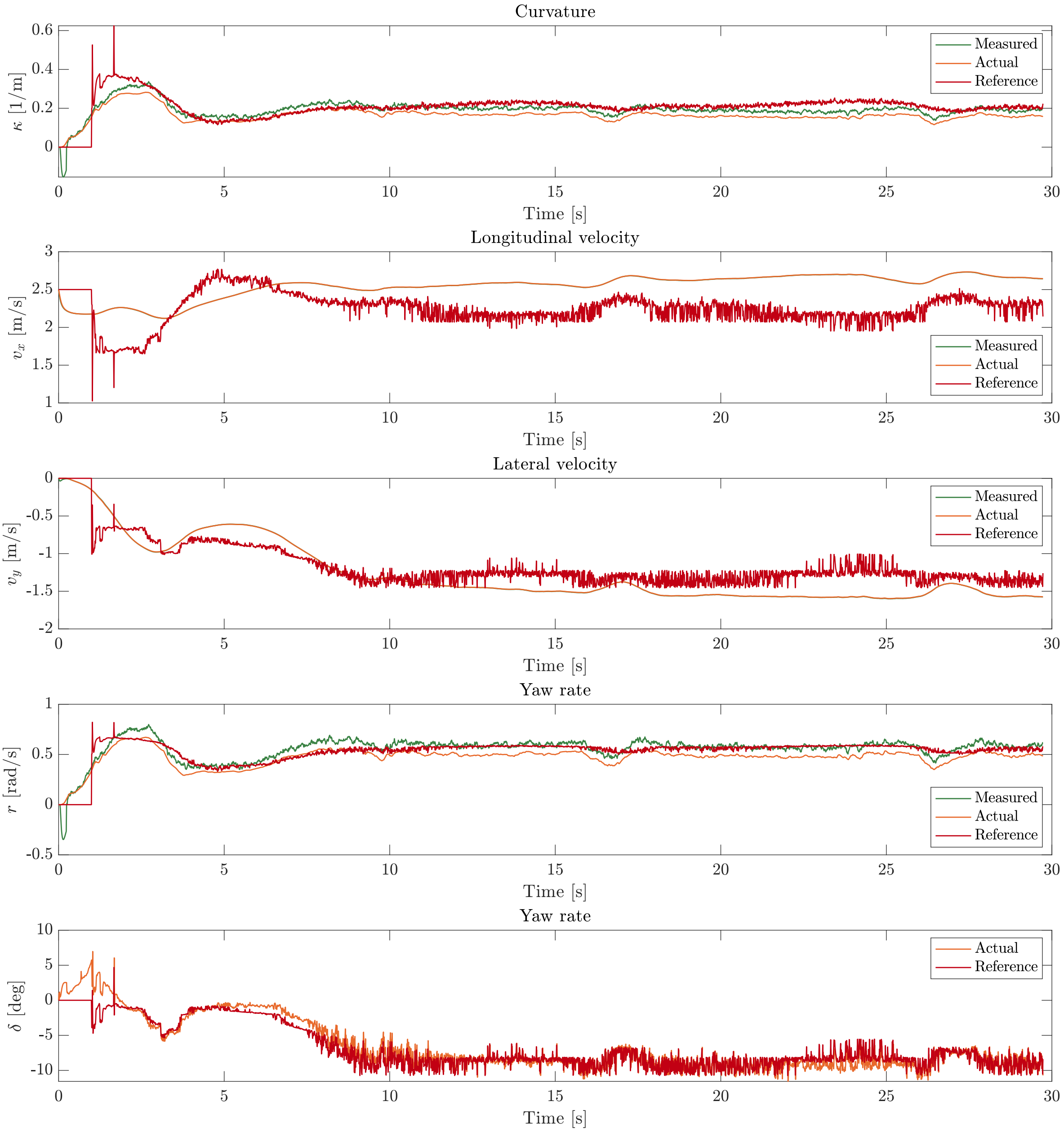

Figure 6-7: System response after implementation of Ref. 2 (see Table 6-4)

6-2-2 Actuator dynamics

Even though the control system runs at a frequency of 100[Hz] there is the possibility that the actuators do not have the same rate. Therefore it scenario's are considered that the actuator sampling time t_{act} may differ. The impact of these different sampling times can be found in Figure 6-8.

The effect that this variation of actuator frequency brings is described in Table 6-6. This shows that there is a slight decrease in the path following ability when the actuator sampling time is greater than 0.03[s]. The ability to estimate system states does not rigorously change as does the ability for the optimization problem to be solved.

Table 6-6: Actuator sampling time variance

t_{act}	VAF-value			$\log_{10}$(KKT-value)		RMSE
	v_x	v_y	r	Median	95p	s_n
0.01 [s]	99.9904	99.9964	83.6720	4.86	7.08	0.586
0.02 [s]	99.9905	99.9964	83.8383	4.86	7.10	0.586
0.03 [s]	99.9897	99.9956	85.8648	4.87	7.06	0.628
0.04 [s]	99.9894	99.9956	86.6473	4.88	7.14	0.636
0.05 [s]	99.9905	99.9958	85.9719	4.85	7.09	0.647

Besides the possible difference in the sampling time of the actuators there is a limit in how much the actuator can move within a given time. This gives rise to the potential problem that the desired control inputs may not (fully) as desired. These limitations for the actuator are to be investigated in further research.

The input for actuators may not be equal to the control signal computed, e.g. the servo actuating the steering angle will not have the angle δ as an input. A method to convert the signals to ones actually usable for such servo must be designed.

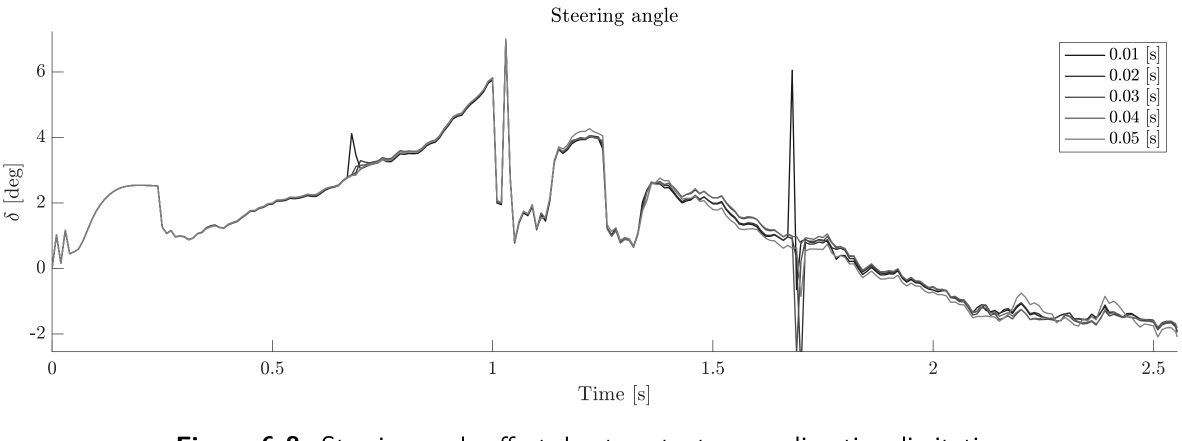

Figure 6-8: Steering angle effect due to actuator sampling time limitation

6-2-3 Parametric uncertainty

Measurement of the vehicle parameters (Table 6-1 and tyre parameters 6-2) may not be completely accurate when performed in the lab. Therefore an analysis is done to observe system behavior when a discrepancy occurs between model- and actual parameters. Assumed is that a minimum accuracy of 90% can be obtained for dimensional measurements of vehicle parameters and that inertial values and Magic Formula Tyre parameters can be determined with a 75% accuracy.

Using Monte Carlo simulation with random numbers to define parameter variance for each simulation, provides an insight into the effects that parametric uncertainty have. The ability of the vehicle to maintain a stable drifting motion is analyzed. To quantify the performance the ability to track the desired curvature is analyzed and the ability for the controller to follow the desired path is evaluated. Lateral deviation (oscillating) beyond 5[m] defines the threshold for undesired path following properties. For each situation the poles of the system, as done at the beginning of this section, are computed. As this varies according to the vehicle parameters a color-scale is applied for deviation from the baseline. For a more clear representation which parameter combinations lead to successful drift control, the transparency is increased towards the mean value, clustering the 'slowest' and 'fastest' pole combinations per color, allowing to compare differences between these sets.

First, the Proportional Integral Derivative (PID)-path following controller and dynamic velocity controller are deactivated, and a series of 556 simulations are performed with random parameter settings. As there is no path following property for this configuration, this cannot be evaluated. And thus only the curvature κ is analyzed in Figure 6-9.

There are two major clusters identified in, one where the vehicle curvature is slightly below the baseline and one where the curvature is slightly higher. These different behavioral clusters highlight two specific parameter combinations occurring for these situations.

There are a number of parameters which tend to influence vehicle behavior more than others. When looking at the mass of the vehicle m there tends to be 'slow' and 'fast' poles displaced along the varying parameter. This indicates that this is not one of the critical parameters influencing system behavior. The same can be said for the lengths from the Center of Gravity (CoG) to the axles l_j, though still for all these parameters correct measurement is of course preferred.

Three parameters describing the tyre-road interaction B, C, and D, and the yaw inertia tensor I_z tend to have a specific influence on the system behavior. State behavior for these clusters can be observed in Figure 6-10. The curvature of all these highlighted behavior patterns are highly comparable, varying only in the steady-state value to which these converge to. The 'faster' cluster (highlighted in green) can be argued to take corners faster and sharper, having a higher BCD pair, indicating a steeper initial slope in the curve produced by the original Magic Formula (Figure 2-3) and a higher road friction coefficient (due to a higher D value) resulting in more acceleration for same steering and torque input. Also due to the lower yaw inertia tensor, the vehicle will be prone to faster rotational acceleration than modeled. The controller responds adequately by counter steering less stabilizes the drifting motion in a slightly larger radius circle. The exact opposite can be said for the 'slower' cluster (highlighted in blue). However, there are a few scenarios where the velocities do not stabilize indicating that having a vehicle which turns slower than expected may run into instability issues faster.

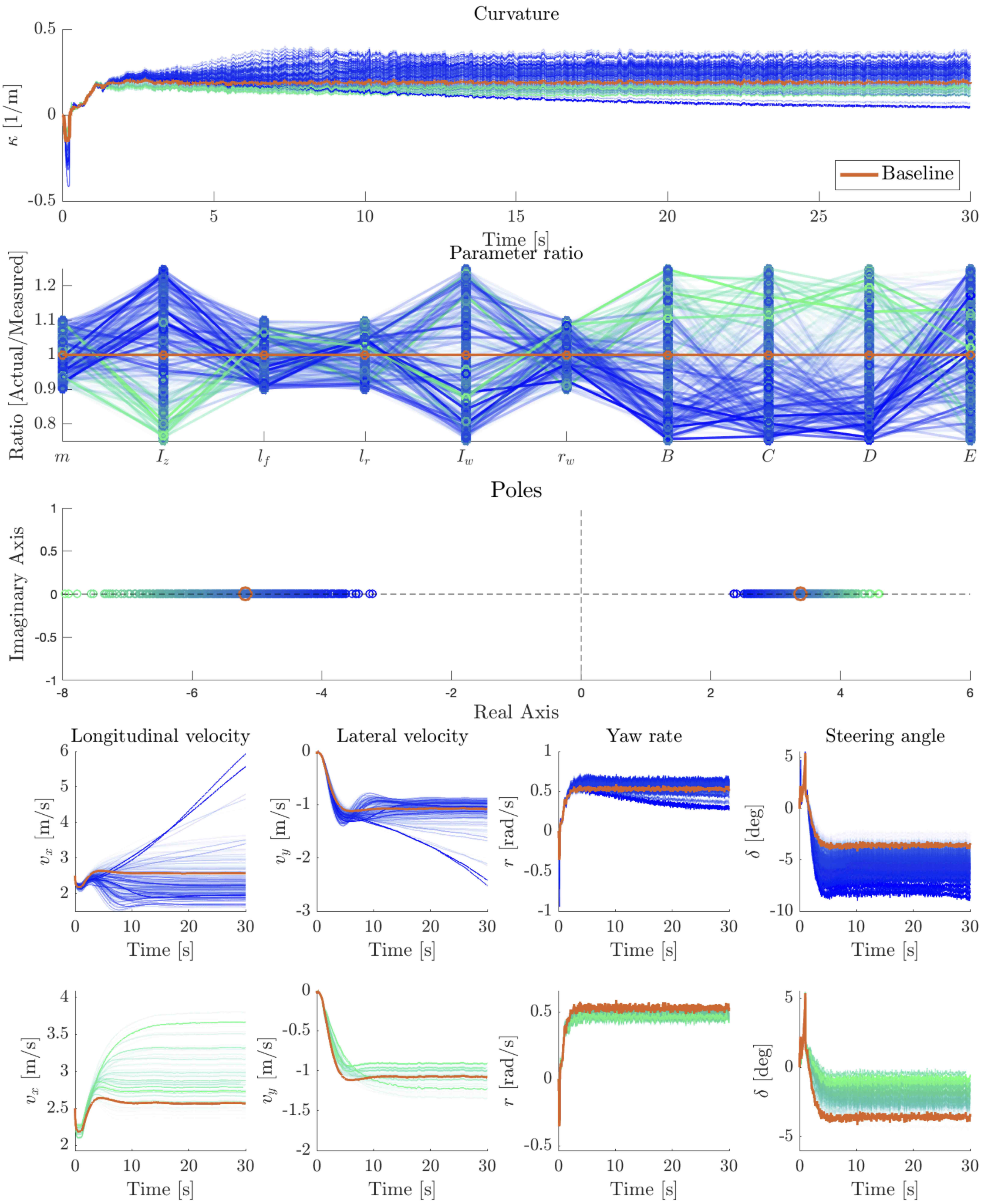

Figure 6-9: Parametric uncertainty for system without path following control

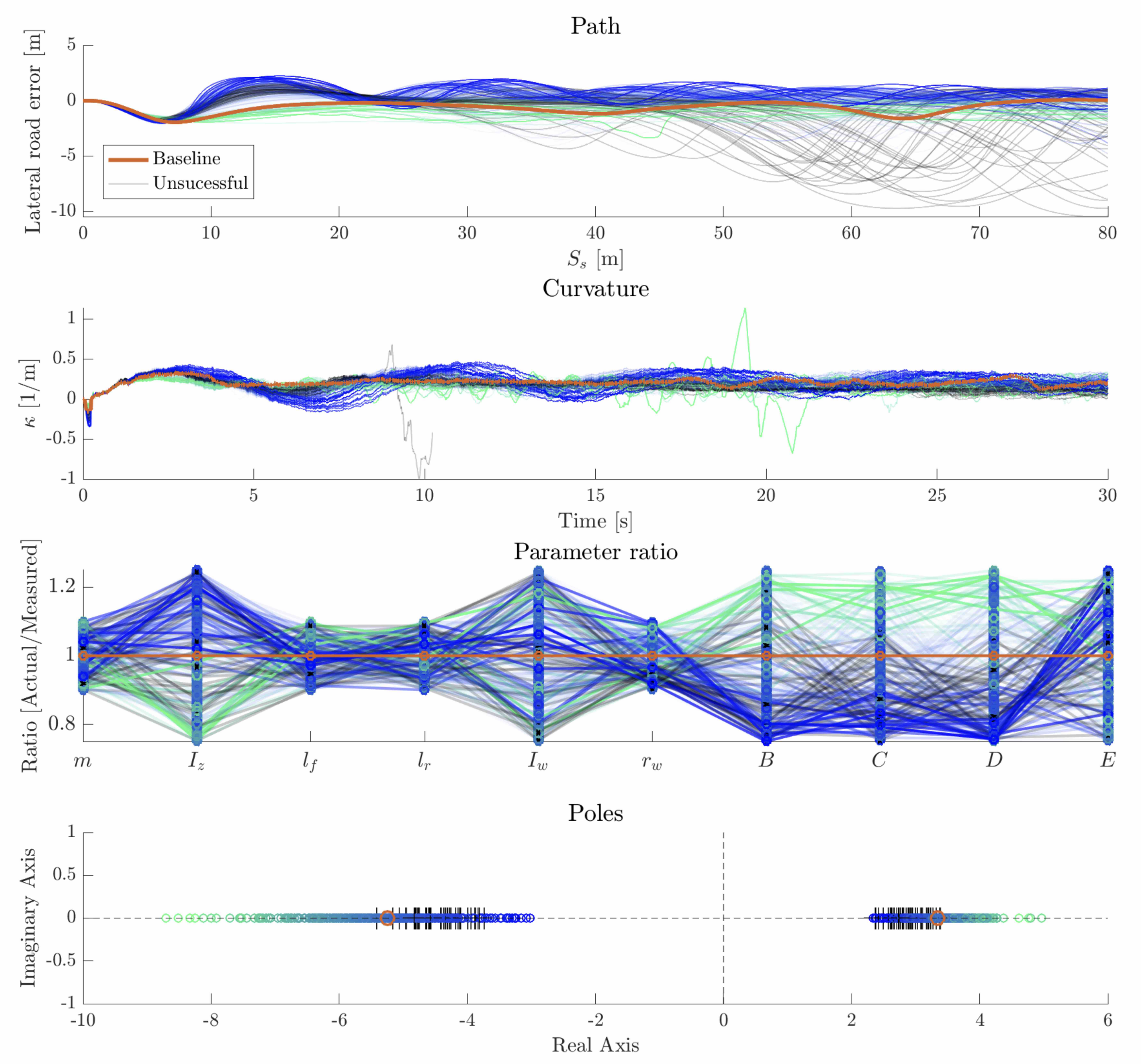

Figure 6-10: System response to parametric uncertainty

After activating the path following- and dynamic velocity controller, 518 simulations are performed. Again using random parameter settings while implementing all uncertainties mentioned before. Figure 6-10 shows how the system behaves regarding these uncertainties.

Many similarities are visible between both cases (with and without added PID|DV control), as the clusters observed practically highlight the same parameter groups. The behavior of the closed-loop system highlights three clusters in Figure 6-10; again on 'faster' (green) cluster and one 'slower' (blue) cluster but now including one which shows unstable behavior where the control system is not able to steer the vehicle towards the desired path.

It initially seems that both the slow- and fast- clusters are able to stabilize the vehicle and are able to follow the desired path. However, when analyzing the behavior of the vehicle states (Figure 6-11), it can be concluded that this is not always the case. The cluster with fast system dynamics cases shows high-frequency oscillations for states and control inputs,

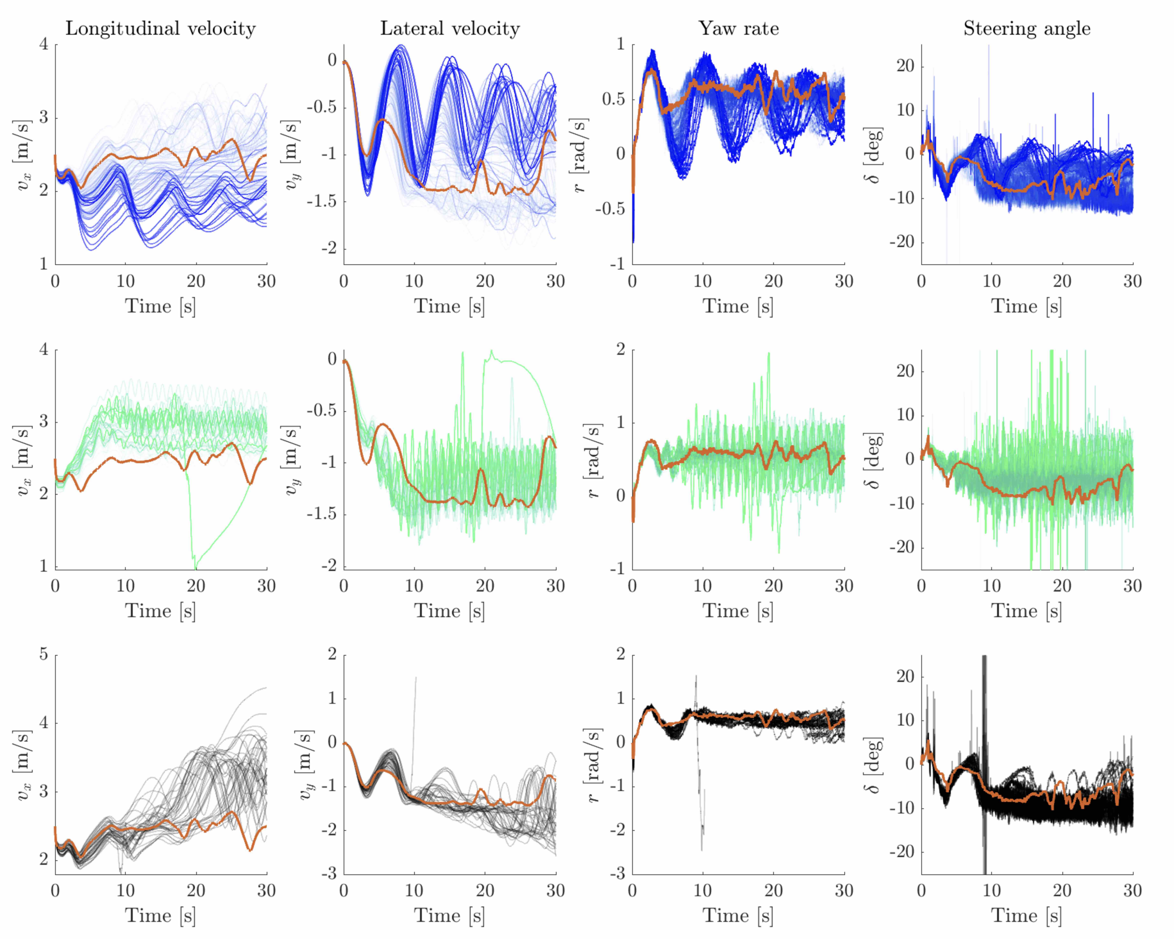

Figure 6-11: State trajectories for parametric uncertain system response clusters

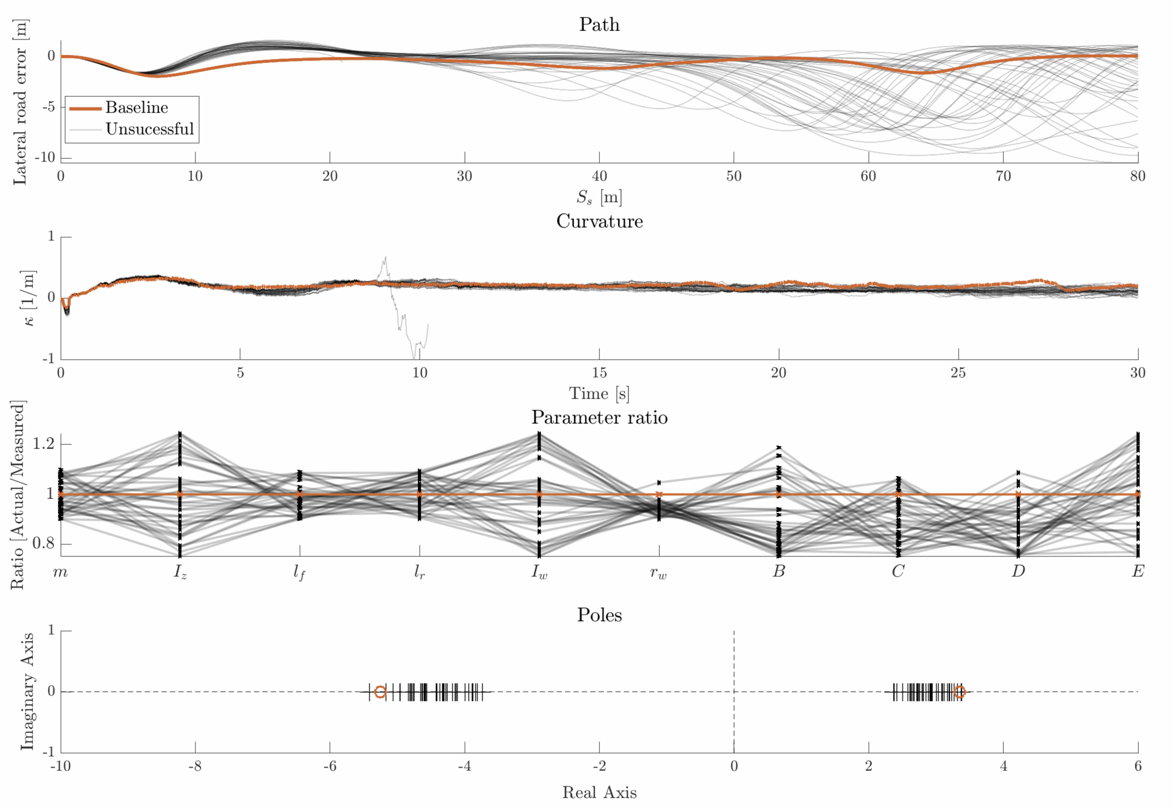

Figure 6-12: Unstable state trajectories for parametric uncertain system response clusters

including for the steering angle δ. The slow cluster shows much more reasonable behavior though the states are not damping nicely. Both forms of oscillations are not desired, though re-tuning of the PID and Dynamic velocity controller could result in a better response.

The cluster with unstable behavior, unsuccessful in stabilizing the vehicle, is highlighted in Figure 6-12. Most interesting is to see that the vehicle poles in the unstable cluster practically are all slower than the baseline and that the smaller rear wheel radius r_w ratio shows a huge dominance towards in the unstable behavior cluster. A smaller ratio than expected, when having the same torque applied, results in a higher longitudinal force directed from the wheels to the chassis. This partially explains why the velocity states (v_x and v_y in Figure 6-11) are generally overall higher than desired, even though there is lower friction surface (due to the lower BCD pair) than expected.

From this analysis, it can be concluded that the system is much more robust to parametric uncertainty when the actual system dynamics are faster than expected than when they are much slower than expected when path following is the key performance indicator. However, this gives rise to high-frequency oscillations in state dynamics and control input, indicating that the controller itself is much less robust in these scenarios. Slower system dynamics than expected should prove easier to compensate for when tuning the controllers. The wheel radius remains a very important property to measure, especially as overestimation may result in system instability.

6-2-4 Initial condition variation

During simulation, the controller is switched on constantly and the simulation is initiated regarding a specific set of initial conditions for the vehicle. When the initial conditions match the desired equilibrium state closely it is much easier for the controller to actuate the system to keep it at this desired equilibrium as when the desired equilibrium lies far apart from the initial states. To find out whether a switching logic may be necessary to determine when the controller could be activated a series of simulations is done with a variety of initial conditions as done in Section 5-1.

From Figure 6-13 it can be said that the same general idea holds as that was made in the previous chapter when regarding the full scaled vehicle, that the controller is more robust when activated at a moment that the vehicle velocity is higher than the equilibrium state. However, the state trajectories behave much more violent than for the full-scale vehicle, likely due to the faster poles of the scaled vehicle. The multiplicity of desired (final) state equilibria is due to the path following- and dynamic velocity controllers manipulating the reference signal.

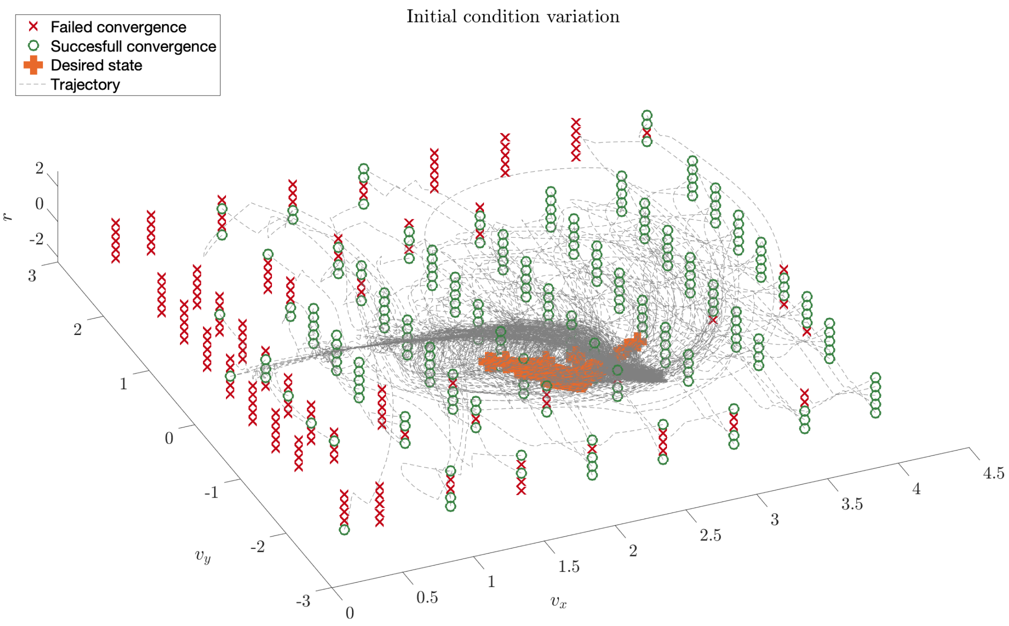

Figure 6-13: Scaled vehicle initial condition variation

Interesting is to see that in comparison to the initial condition variation section (5-1) the control system is able to successfully control the vehicle at initial conditions that are almost zero. During the simulations the situations when the initial conditions where zero, it was not the controller which wasn't able to generate solutions to the Optimal Control Problem (OCP) but rather the dynamic model of the vehicle which tended to reach singularities. This gives reason to believe that the system can be activated with the vehicle at a stand-still condition, a condition favorable when actually implementing the controller on the scaled vehicle.

6-3 Summary

The experimental platform at the NERDlab in the department of DCSC provides a scaled vehicle which in the future can potentially be used to test an autonomous drift controller. To evaluate the feasibility of this idea a simulation study is performed to test whether the designed control system from the previous chapters could be implemented on this 1:10 scaled vehicle. The resulting vehicle dynamics have much 'faster' properties than an actual (full-scale) vehicle, described in the previous sections. After scaling and re-tuning, the control system was still able to initiate and sustain a drifting motion for the vehicle. Uncertainties and imperfections of the system are added to the system to simulate noisy sensor measurements, delayed signals and different operating frequencies for the actuators. The Kalman Filter designed is computationally intensive and able to estimate the vehicle states fairly well when using combined measurements. Simulating different operating frequencies for the actuators show that there is a decrease in performance when $t_{\text{act}} \geq 0.3$[s]. Model parameter mismatches show that there are two situations to be considered; when the system dynamics are slower and faster than of the model description. These situations both have a different impact on the behavior of the closed-loop system. The scenarios where the control system was unable to stabilize the drifting motion occurred when the system dynamics were slower than modeled. The controller can probably be initiated from a stand-still condition, a preferable condition for future implementation. However, the simulation of this scenario proves difficult due to singularity issues of the simulated vehicle.

Chapter 7

Conclusions and recommendations

This chapter contains the conclusions based on this thesis and the recommendations for further research in this field. First, a brief summary will describe the work done in this thesis. Then some conclusions will be drawn regarding the methodology and the proposed simulation model, whereafter the performance of the simulation model and control system is discussed. After this, the focus will be on the potential implementation on a scaled vehicle, the limitations that this brings and the challenges that arise in bringing this control system into reality. After these conclusions, some recommendations are done for further research, investigating where improvement of the simulation model and controller may lie when continuing work on autonomous drift control systems.

7-1 Summary

As autonomous vehicles are in an up-rise the last few years and focus is placed on full vehicle autonomy and safe implementation hereof. Advanced Driver-Assistance Systems (ADAS) are preventing vehicle crashes and shows promising development, including applications for Autonomous Drift Control (ADC). Control of this drifting motion will supplement the ability of autonomous vehicles to drive in all situations and during all conditions.

Model Predictive Control (MPC) requires a description of the system to predict the future behavior of the vehicle. This description is done by modeling the simplified system dynamics as a three-state bicycle model. The tyre-road interaction is modeled with a simplified version of the Magic Formula (MF) to approximate the F_x and F_y curves with respect to the wheel slip. A modified longitudinal slip is proposed to better accommodate gradient-based optimization methods.

The art of drifting is analyzed and three major components arise; saturation of the rear tyres, large body side-slip angles, and counter steering. This behavior characterizes drifting motion and can be found in equilibrium solutions describing stead-state drifting. It is shown that different tyre-road specifications result in two types of $R - V$ equilibria grids, convex tyres providing a single plane of driving equilibria and non-convex tyres providing multiple planes.

This equilibrium grid is used in combination with a curvilinear coordinate system to define path following properties and with it identify deviations hereof.

A control system is designed based on the MPC strategy. The Automatic Control and Dynamic Optimization (ACADO) toolkit provides a flexible environment to efficiently compute control inputs required to steer the vehicle towards desired drifting equilibria. It is found that the simplest integration method is able to predict system dynamics well enough and with it minimize computational demand. An additional path following Proportional Integral Derivative (PID) controller is designed to let the vehicle follow a predetermined path. A dynamic velocity controller is designed to adjust the vehicle velocity to accommodate small radius corners while taking larger corners at a higher velocity. This control system shows the ability of the simulated vehicle to successfully initiate a drifting motion and follow the given path.

To prepare for implementation of such a control system a feasibility study is performed to analyze differences between a full-scale and 1:10 remote-controlled vehicle. It is found that the different properties result in a vehicle with much faster dynamics though simulations show that this is not a problem for the designed control system. Also, various tests modeling noise, delay, and other uncertainties are done to identify potential future challenges.

7-2 Conclusions

In literature, MPC is often regarded too computational intensive for application in settings which require fast responses and computations at a high frequency. However, due to the relatively low complexity of the tree-state bicycle model and application of the Real-Time-Iteration (RTI) strategy to solve the Optimal Control Problem (OCP), application of the MPC strategy for autonomous drift control can be considered feasible.

Modeling vehicle dynamics using the three-state bicycle model with simplified wheel dynamics results in a compact system description which can be used for control purposes. The non-smooth formulation of the wheel slip ratio can successfully be transformed into a smooth approximation, introducing no further control issues.

Extending the MPC strategy with the proposed dynamic velocity controller allows the vehicle to take a larger range of corners compared to a static reference velocity. This also provides robustness for the system as the reference generator only generates feasible equilibria. Combining this with a path-following PID controller lets the vehicle take a large range of paths and adjust the velocity accordingly.

Future implementation will start with testing on a 1:10 scaled vehicle, and therefore a feasibility study is performed. Adjusting the controller shows that the setup regarding vehicle parameters and tyre profile does not pose a threat to the working of the controller. Although the parameters describing the scaled vehicle result in a dynamically ’faster’ system, tuning the controller accordingly allows usage of the proposed strategy.

Delays active in the system are not perceived as a threat to the control system when proper (re-)tuning is done. Measurement of the system states based on the Inertial Measurement Unit (IMU) results in a noisy signal, which can be partially corrected for using an Extended Kalman Filter (EKF). However, using only the IMU specified in the analysis as a measurement source will provide state estimates which are not accurate enough to achieve proper path following drift control.

The parameters describing the vehicle and the tyre-road interaction may deviate slightly from its actual parameters, potentially having a great negative impact on the stability of the closed-loop system. Especially properties linked to the tyres and wheels are identified as critical components, therefore it can be concluded that the correct modeling of the tyres requires much attention before the designed control scheme can be implemented. An analysis is performed to identify these critical vehicle parameters.

Variation of the initial conditions shows that there is only a certain range of situations where the simulation can be switched on. The system is to be initialized with a certain starting velocity, which doesn't have to be larger than 0.1[m/s] for the scaled vehicle, to allow the algorithm to converge properly. This is different than for the actual vehicle, which may be explained due to the addition of the dynamic reference velocity. This additional controller makes the control system more robust to these low-speed initial conditions as the difference between the low initial velocity and a high reference velocity is reduced.

7-3 Future recommendations

There is room for improvement of the simulation environment, as even though there are a lot of uncertainties regarded in this analysis it is still not an exact replica of a real-life scenario. Further simulation-based analysis can include controlling a higher order vehicle model to approach a real-life scenario even better.

The path following PID controller should be re-tuned for varying $R - V$ pairs as the reaction of this controller is greatly dependant the lateral deviation s_n. The current design of the PID controller provides a good option for drifting in a constant radius circle, as the reference velocity does not change much. This desired path should, therefore, be the first goal when implementing the autonomous drift controller. There can be looked into the option of applying a different strategy to achieve the path following properties that depends less on tuning and is more resilient to varying vehicle velocity.

The controller computes two control signals for the system, namely the steering angle δ and the torque that is applied to the rear wheels $T_{\text{net},r}$. These cannot be directly implemented onto a scaled remote-controlled vehicle, as the servos actuating have a different electrical signal as an input. This conversion has to be further identified and a way to bridge this discrepancy is required and must be investigated.

The current delay of 250[ms] acting between the IMU and controller does not cause instability of the system with proper tuning. Though minimization of this delay will allow for higher vehicle speeds and with it higher side-slip angles. In this analysis the delays are assumed to be constant. As this assumption is not realistic it is useful to investigate the impact of delays with varying length. Placing the controller on-board the vehicle will reduce these delays between sensors and controller, allowing a faster response of the MPC controller and with it increase performance. Another option for improvement of the MPC controller is by modeling the system delays inside ACADO integrator. Inside the C-code describing this controller the discretized system is described and can be modified.

When using the designed Kalman Filter and Motion Capture (MoCap) configuration to filter noise from the IMU measurements, the variance of the measurements should not exceed the values presented in Table 7-1. This prevents a noisy signal sent to the actuators and maintains feasibility of the OCP. These variance-values are taken from the actual simulated IMU specifications, as the generated noisy control signal was assumed usable for the actuators. Bias values of ± 0.5, which are very high regarding actual sensors, bring forward no critical issues, and therefore still could be used in this scenario. The state-observer design should be extended or redesigned for use without the MoCap system, as the state-estimation performance of the EKF should be improved to provide higher accuracy and control signals with less noise.

Table 7-1: IMU accuracy recommendations

Parameter	Limit
σ^2_{gyro}	0.0175 [rad/s]
σ^2_{acc}	0.03 [m/s^2]
b	$\pm$0.5 [rad/s]

The uncertainty of the parameters describing the vehicle has shown to bring forward potential instability of the system. The parameters describing the yaw inertial tensor I_z, the effective wheel radius r_w and the parameters describing the tyre-road interaction B,C & D are crucial to ensure stable system behavior. Accuracy recommendations for measuring these parameters can be found in Table 7-2. These are determined specifically for the chosen simulation setup. The yaw inertia tensor can not directly be identified as a parameter which results in instability, however, for the model to rotate in a similar fashion a variation of $\pm 5\%$ is recommended. Variations, where the effective wheel radius of the vehicle is smaller than modeled, brings forward most of the unstable scenarios. Measurement of this radius smaller than the actual value should be avoided. The tyre parameters resulting in 'faster' system dynamics result in less unstable situations. As these values are determined empirically the recommendations should be interpreted as a guideline. As tyre properties tend to be a crucial element in the performance of the closed-loop control system investigated can be if the tyre modeling is accurate enough or whether modeling hereof should be adjusted. Extending the tyre modeling for anisotropic properties or transient properties could be done to capture the actual tyre-road interaction better.

Table 7-2: Parameter accuracy recommendations

Parameter	Upper limit	Lower limit
I_z	+5%	-5%
r_w	+5%	-0%
B	+5%	-5%
C	+10%	-5%
D	+10%	-5%

There will always be differences between the simulation environment and actual implementation. Testing of the control system the scaled vehicle will allow better identification which challenges must be overcome to achieve this optimization based control for autonomous drifting. Exemplified, in this thesis found is that initialization at initial conditions which are all zero brings forward a lot of singularity issues in the simulation of vehicle behavior. This issue will result differently when using an actual scaled vehicle, as there is then no need for a simulated vehicle model.

Though application of autonomous drift controllers in actual sized vehicles to increase safety should be further investigated, the opportunity to control these, exciting, high side-slip drifting maneuvers definitely exists.

Bibliography

[1] European Commission, "Advanced driver assistance systems," November 2016.

[2] H. B. Pacejka, *Tire Characteristics and Vehicle Handling and Stability.* Elsevier Ltd., 2012.

[3] D. Tavernini, M. Massaro, E. Velenis, D. I. Katzourakis, and R. Lot, "Minimum time cornering: the effect of road surface and car transmission layout," *Vehicle System Dynamics*, vol. 51, pp. 1533–1547, oct 2013.

[4] E. Eckermann, *World History of the Automobile.* SAE Press, 2001.

[5] J. Urry, "The System of Automobility," *Theory, Culture & Society*, vol. 21, no. 5, pp. 25–39, 2004.

[6] F. Malin, I. Norros, and S. Innamaa, "Accident risk of road and weather conditions on different road types," *Accident Analysis and Prevention*, vol. 122, no. August 2018, pp. 181–188, 2019.

[7] M. Acosta, S. Kanarachos, and M. Blundell, "Vehicle agile maneuvering: From rally drivers to a finite state machine approach," *IEEE Symposium Series on Computational Intelligence, SSCI*, no. September, 2016.

[8] E. Velenis, P. Tsiotras, and J. Lu, "Optimality Properties and Driver Input Parameterization for Trail-braking Cornering," *European Journal of Control*, vol. 14, pp. 308–320, jan 2008.

[9] M. Acosta, *A Drift-based approach to Last-Moment Accident Avoidance Manoeuvres on Loose Surfaces.* PhD thesis, Coventry University, 2017.

[10] "Autonomous Driving | MCFM | McKinsey & Company." https://www.mckinsey.com/features/mckinsey-center-for-future-mobility/overview/autonomous-driving. Online; Accessed at 2019-02-25.

[11] J. Gonzales, F. Zhang, K. Li, and F. Borrelli, “Autonomous Drifting with Onboard Sensors,” in *Advanced Vehicle Control AVEC’16*, pp. 3–8, 2015.

[12] P. Polack, F. Altch, and B. Andr, “The Kinematic Bicycle Model : a Consistent Model for Planning Feasible Trajectories for Autonomous Vehicles ?,” no. Iv, pp. 6–8, 2017.

[13] H. Dugoff, P. S. Fancher, and L. Segel, “An Analysis of Tire Traction Properties and Their Influence on Vehicle Dynamic Performance,” *International Automobile Safety Conference*, 1970.

[14] E. Fiala, “Seitenkräfte am rollenden luftreifen,” *VDI Zeitschrift*, vol. 96, pp. 973–979, 1954.

[15] A. Bhoraskar and P. Sakthivel, “A Review and a Comparison of Dugoff and Modified Dugoff Formula with Magic Formula,” in *International Conference on Nascent Technologies in the Engineering Field (ICNTE-2017)*, pp. 1–4, 2017.

[16] D. Ammon, “Vehicle dynamics analysis tasks and related tyre simulation challenges,” *Vehicle System Dynamics : International Journal of Vehicle Mechanics and Mobility*, vol. 43:sup1, pp. 30–47, 2005.

[17] M. Acosta, S. Kanarachos, and M. E. Fitzpatrick, “A Hybrid Hierarchical Rally Driver Model for Autonomous Vehicle Agile Maneuvering on Loose Surfaces,” in *International Conference on Informatics in Control, Automation and Robotics*, pp. 216–225, 2017.

[18] J. Cook, “Basic properties of the soft maximum,” *UT MD Anderson Cancer Center Department of Biostatistics Working Paper Series; Working Paper 70; Bepress: Houston, TX, USA*, pp. 1–4, 2011.

[19] M. Abdulrahim, “On the Dynamics of Automobile Drifting,” *SAE Technical Paper*, vol. 1, pp. 10–19, 2006.

[20] M. Acosta and S. Kanarachos, “Highly Skilled Autonomous Off- Road Vehicles,” in *Automotive Electronics Innovation (AESIN) Conference*, (Birmingham), pp. 1–22, 2017.

[21] J. Edelmann and M. Plochl, “Handling Characteristics and Stability of the Steady-State Powerslide Motion of an Automobile,” *Regular and Chaotic Dynamics*, vol. 14, no. 6, pp. 682–692, 2009.

[22] R. Y. Hindiyeh, *Dynamics and Control of Drifting in Automobiles.* PhD thesis, Stanford University, 2013.

[23] J. C. Gerdes, “Equilibrium Analysis of Drifting Vehicles for Control Design,” in *ASME 2009 Dynamic Systems and Control Conference*, no. 1, pp. 1–8, 2014.

[24] K G Baass, “Use of clothoid templates in highway design,” in *Roads and Transportation Association of Canada Conference*, p. 16, 1982.

[25] E. Jelavic, F. Gonzales, and F. Borrelli, “Autonomous Drift Parking using a Switched Control Strategy with Onboard Sensors,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3714–3719, 2017.

[26] E. Wachter, *Lateral Path Tracking in Limit Handling Condition using SDRE Control.* PhD thesis, Chalmers University of Technology, 2016.

[27] T. Keviczky, P. Falcone, F. Borrelli, J. Asgari, and D. Hrovat, "Predictive Control Approach to Autonomous Vehicle Steering," *2006 American Control Conference.*

[28] M. Diehl, H. G. Bock, and J. P. Schlöder, "A Real-Time Iteration Scheme for Nonlinear Optimization in Optimal Feedback Control," *Society for Industrial and Applied Mathematics*, vol. 43, no. 5, pp. 1714–1736, 2005.

[29] H. J. Ferreau, *An Online Active Set Strategy for Fast Solution of Parametric Quadratic Programs with Applications to Predictive Engine Control.* PhD thesis, Ruprecht-Karls-Universität, 2006.

[30] G. Torrisi, S. Grammatico, R. S. Smith, and M. Morari, "A projected gradient and constraint linearization method for nonlinear model predictive control," *SIAM Journal on Control and Optimization*, vol. 56, no. 3, pp. 1968–1999, 2018.

[31] W. Karush, *Minima of Functions of Several Variables with Inequalities as Side Constraints.* PhD thesis, University of Chicago, 1939.

[32] H. W. Kuhn and A. W. Tucker, "Nonlinear programming," in *Berkeley Symposium on Mathematical Statistics and Probability*, pp. 481–492, 1951.

[33] T. a. Wenzel, K. J. Burnham, M. V. Blundell, and R. a. Williams, "Dual extended Kalman filter for vehicle state and parameter estimation," *Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility*, vol. 44, no. 2, pp. 153–171, 2006.

[34] S. Antonov, A. Fehn, and A. Kugi, "Unscented Kalman Filter for Vehicle State Estimation," *Vehicle System Dynamics*, vol. 49, pp. 1497–1520, 2011.

[35] M. Zanon, J. V. Frasch, and M. Diehl, "Nonlinear Moving Horizon Estimation for Combined State and Friction Coefficient Estimation in Autonomous Driving," *European Control Converence (ECC)*, pp. 4130–4135, 2013.

[36] "SparkFun 9DOF Razor IMU M0." `https://www.sparkfun.com/products/14001`. Online; Accessed at 27-02-2019.

Glossary

List of Acronyms

ACADO	Automatic Control and Dynamic Optimization
ADAS	Advanced Driver-Assistance Systems
ADC	Autonomous Drift Control
CoG	Center of Gravity
DAE	Differential-Algebraic set of Equations
DCSC	Delft Center for Systems and Control
EKF	Extended Kalman Filter
IMU	Inertial Measurement Unit
KKT	Karush-Kuhn-Tucker
LQR	Linear Quadratic Regulator
MF	Magic Formula
MoCap	Motion Capture
MPC	Model Predictive Control
NERDlab	Network Embedded Robotics DCSC lab
OASES	Online Active SEt Strategy
OCP	Optimal Control Problem
PID	Proportional Integral Derivative
RK	Runge-Kutta
RMSE	Root-Mean-Square Error

ROS	Robotic Operating Software
RTI	Real-Time-Iteration
RWD	Rear Wheel Drive
QP	Quadratic Problem
SDRE	State Dependent Ricatti Equation
SQP	Sequential Quadratic Programming
VAF	Variance Accounted For

List of Symbols

α	Vehicle Heading [rad]
α_j	Lateral Wheel Slip [-]
β	Body Side-slip Angle [rad]
Δ	Difference operator
δ	Steering Angle [rad]
ϵ	Heading Error [rad]
η	Gaussian Noise
κ	Curvature [1/m]
λ	Longitudinal Wheel Slip [-]
μ_r	Road Friction Coefficient [-]
ω	Angular Wheel Velocity [rad/s]
ρ	Scaling parameter [-]
$\sigma_{i,j}$	Wheel Slip [-]
p_i	System poles
a_c	Centripetal Acceleration [m/s^2]
$\mathrm{C}_{y,f}$	Front lateral cornering stiffness [N/rad]
$\mathrm{F}_{i,j}$	Force [N]
h	Step size [s]
I_z	Yaw Inertia Tensor [kg m^2]
I_w	Wheel Inertia Tensor [kg m^2]
l_j	Length from axle to Center of Gravity (CoG) [m]
m	Mass [kg]
N	Horizon length [-]
Q_k	Process Noise Covariance matrix
R	Radius [m]
r	Yaw Rate [rad/s]

R_k	Measurement Noise Covariance matrix
r_w	Effective Wheel Radius [m]
s	Integration Order
s_s	Longitudinal Distance [m]
s_n	Lateral Distance [m]
T_{net}	Netto Torque [Nm]
u	Control Input
V	Velocity [m/s]
v_k	Measurement Noise
v_x	Longitudinal Velocity [m/s]
v_y	Lateral Velocity [m/s]
w_k	Process Noise
x	States
y	System Output

$_{\text{acc}}$	Accelerometer
$_{\text{gyro}}$	Gyroscope
$_{\text{lb}}$	Lower bound
$_{\text{ub}}$	Upper bound
$_{i}$	Direction, i $\in [x, y, z]$
$_{j}$	Front- or rear, j $\in [f, r]$
$_{k}$	Timestep indicator

$^{\text{eq}}$	Equilibrium
$^{\text{IMU}}$	IMU
$^{\text{MoCap}}$	MoCap
$^{\text{Wheel}}$	Wheel Encoder