A Real-Time Rollover Prevention MPC Controller With a Predictive Load Transfer Ratio

Eric Hu

TU Delft

A Real-Time Rollover Prevention MPC Controller With a Predictive Load Transfer Ratio

by

Eric Hu

to obtain the degree of Master of Science in Mechanical Engineering

Track: Vehicle Engineering

Specialisation in Vehicle Dynamics and Control

at the Delft University of Technology.

Student number:	4497279	
Thesis committee:	Dr. ir. R. Happee,	TU Delft, Chair, Associate Professor, CoR-IV
	Dr. B. Shyrokau,	TU Delft, Supervisor, Assistant Professor, CoR-IV
	Dr. L. Ferranti,	TU Delft, External, Assistant Professor, CoR-LAC

Acknowledgements

Here I want to show my appreciation for all the people that have helped me get to this point and complete this thesis. First of all I would like to thank my supervisor Barys Shyrokau. It was a long process, but he was always patient and willing to help. When I needed it he would give me a push with feedback so that I could make great progress on this work. Besides his supervisory role, I also want to thank him for his role as a teacher. His courses on vehicle dynamics were extremely interesting and well-taught. With that said, I also want to thank all the other professors that taught me many invaluable things during my time at TU Delft.

The courses in Delft are certainly tough, and I wouldn't have been able to get through it without my great classmates. Whether it was working on projects together or trying to solve a problem together, nothing was impossible as long as we worked together. Their drive and ambition also pushed me to be the best I could be. So I want to thank all my fellow students during my time here in Delft.

Finally, I want to thank my friends and family for always supporting me. This was not the easiest period to work on a Master's thesis and without them motivating me this would not have been possible. They were with me through thick and thin, and I will forever be grateful to them.

Eric Hu
Delft, May 2022

Abstract

Over the past few decades vehicles have become more safe than ever. Not only has the chassis design improved substantially, the addition of things such as airbags and seat belts has significantly reduced the fatality rate of passengers involved in traffic accidents. A more recent development are driver assistance systems such as the anti-lock braking system (ABS) and the electronic stability program (ESP). ABS prevents wheel lock during emergency braking and ESP preserves lateral stability. These are active systems designed to help the driver maintain control of the vehicle and prevent dangerous situations from occurring.

One type of accident that has not been addressed enough by these driver assistance systems is the rollover accident. Though a relatively infrequent occurrence, rollover accidents are extremely dangerous. According to statistics from the NHTSA the passenger fatality rate for rollover accidents is almost fourteen times as high compared to the overall accident fatality rate. In fact, despite its infrequent occurrence, rollover accidents account for 33% of all passenger vehicle fatalities. This highlights the need for a rollover prevention system in modern vehicles.

The aim of this thesis is to design an accurate yet computationally light rollover prevention controller. Previous research shows that measuring or estimating rollover state is incredibly difficult. Several methods exist, each with their own advantages and limitations. In this work, the novel Predictive Load Transfer Ratio (PLTR) is used to estimate rollover state. This rollover detection method has good accuracy and does not require special measuring equipment. Moreover, the PLTR has slight predictive properties as well. This is then coupled with a model predictive controller (MPC) in order to actively prevent rollover, as opposed to a more reactive feedback controller. The controller is designed using differential equations derived from a two-track vehicle model in order to keep it as computationally light as possible. A benchmark controller is also designed using the same MPC framework with another rollover detection method.

The controllers are then tested in a virtual environment, with the test vehicle performing the fishhook maneuver which induces rollover in the vehicle. Analysis from the results of the simulation show that the proposed rollover prevention controller has better rollover prevention properties compared to the benchmark controller while being more efficient as well. However, this comes at the cost of reference yaw rate tracking. Based on these results, future work can be done to merge this controller with an ESP system where it only activates in critical rollover scenarios.

Nomenclature

α	Tire side slip angle
β	Vehicle side slip angle
δ_d	Steering wheel angle
δ	Tire steering angle
$\dot{\delta}_d$	Steering wheel angular velocity
$\dot{\phi}$	Vehicle roll angular velocity
$\dot{r}$	Yaw acceleration
SR	Vehicle steering ratio
ϕ	Vehicle roll angle
a_x	Longitudinal acceleration
a_y	Lateral acceleration
$C_{\alpha f}$	Front tire cornering stiffness
$C_{\alpha r}$	Rear tire cornering stiffness
$C_{\dot{\phi}}$	Roll damping coefficient
F_{br}	Braking force
F_L	Normal force left tires
F_R	Normal force right tires
F_y	Lateral force
g	Gravitational constant
g	Vehicle track width
h_{cg}	Center of gravity height
I_z	Moment of inertia about z-axis
K_{ϕ}	Roll stiffness coefficient
l_f	Length of vehicle center of gravity to vehicle front
l_r	Length of vehicle center of gravity to vehicle rear
L	Vehicle length
M_z	Moment about z-axis
m	Vehicle mass
R_w	Tire radius

r	Yaw rate
T_{br}	Brake torque
v_x	Longitudinal velocity
v	Lateral velocity

List of Figures

List of Tables

Contents

1 Introduction

1.1. Background

Over the years vehicles have become progressively safer with additional passive safety systems such as the seat belt and active safety systems such as the anti-lock braking system (ABS), and electronic stability program (ESP). Among the types of accidents, vehicular rollover is an important one to consider. Comprising about 2% of all traffic accidents, rollover is a relatively infrequent occurrence. However, while the overall passenger fatality rate for all accidents is 0.2%, the passenger fatality rate for rollover accidents is almost fourteen times higher at 2.7%. This means that despite the infrequent occurrence of rollover, 33% of all passenger vehicle fatalities were due to a rollover accident [1]. The dangerous nature of rollover accidents highlights the importance of systems to prevent or mitigate rollover.

Figure 1.1: Rollover Accident [2]

The risk of rollover differs per vehicle and is primarily influenced by its dimensions. A higher center of gravity (CoG) and/or a more narrow track width translates to an increased risk of rollover. This propensity for rollover is represented by a dimensionless metric called the Static Stability Factor (SSF), given by the following equation:

$$SSF = \frac{T}{2h_{cg}} \tag{1.1}$$

Lower values for the SSF mean the vehicle in question has more potential for rollover. As such, vehicles with higher CoGs such as SUVs, pickup trucks, delivery vans, and commercial trucks are more at risk of rollover accidents. The market for SUVs has grown substantially over the past 10 years. In 2010, 31% of new vehicles sold in the United States were SUVs, whereas in 2020 this figure is now 50%, representing a 61 % increase. If pickups trucks are accounted for as well, that means that over 70% of new vehicles sold in 2020 are vehicles that are at increased risk of rollover accidents. Furthermore, the market for SUVs is expected to grow more in the future. The increasing popularity of SUVs reiterates the need to focus on the issue of rollover.

Rollover may arise due to an extreme steering maneuver. A sharp turn can cause the vehicle to lean to much to one side, resulting in the vehicle tipping over. Another possible cause for rollover is an aggressive steering maneuver in combination with an obstacle such a curb or patch of grass. This is referred to as a tripped rollover, while the former is referred to as an untripped rollover.

1.2. Previous Research

In order to prevent or mitigate rollover, a controller is needed, similar to how an electronic stability program (ESP) prevents vehicular lateral instability. For this, a review of existing methods in the literature used for this purpose was performed. There are two main components in an active rollover prevention controller: the detection of rollover followed by the active prevention of rollover. The detection step is essential and a prerequisite for active prevention step. The findings from the literature review show that the detection step is a difficult issue with several solutions that each have their pros and cons.

1.2.1. Rollover Detection

The first and most critical step for preventing rollover incidents is being able to recognize when it is occurring or about to occur. An accurate measure for rollover detection is necessary in order to take preventative measures. Knowing that a vehicle is approaching rollover will allow the controller activate in a timely manner and prevent the accident from occurring. It is also important to only take action when it is strictly necessary to ensure a smooth driving experience. In the literature, the common methods for rollover detection are summarized below.

Load Transfer Ratio

The load transfer ratio (LTR) is a commonly used metric to detect rollover. Odenthal et al. [3] make use this coefficient for rollover avoidance, which is given by equation 1.2:

$$LTR = \frac{F_L - F_R}{F_L + F_R} \tag{1.2}$$

This gives the load distribution between the two sides of the vehicle and reaches a critical value at -1 or 1. At this point, one of the wheels is lifting off the ground, which is a sign of impending rollover. In theory this is an accurate indicator for rollover, but the issue is that directly measuring forces on the wheels is not currently feasible. Odenthal et al utilize the equilibrium of vertical forces and a set of assumptions to rewrite the equation. The aforementioned assumptions are that the sprung vehicle mass is much larger than the

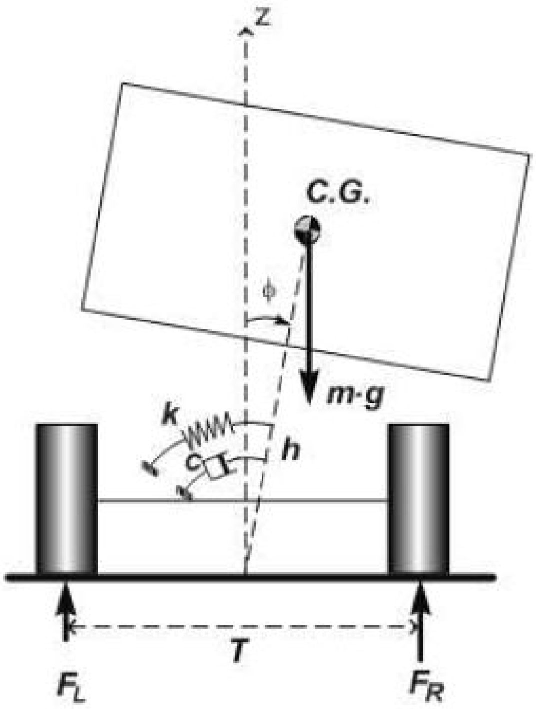


Figure 1.2: Linear model with roll degree of freedom

unsprung mass and that the roll angle is small. These assumptions with the moment balance lead to the static load transfer ratio ($\mathrm{LTR_s}$), given by equation 1.3:

$$LTR_s = 2a_y \frac{h_{cg}}{gT} \tag{1.3}$$

Here, if it is assumed that the CoG stays constant, only the lateral acceleration is necessary for real-time calculation of the $\mathrm{LTR_s}$. Since lateral acceleration is used for the ESP as well, it can be assumed that this signal is available. However, as stated by Solmaz et al [4], the $\mathrm{LTR_s}$ only works well in steady-state conditions and not when transient dynamics come into play.

To combat this issue, a torque balance equation is used to derive the dynamic load transfer ratio ($\mathrm{LTR_d}$) [4] [5], which is given by equation 1.4:

$$LTR_d = 2\frac{C_{\dot{\phi}}\dot{\phi} + K_{\phi}\phi}{mgT} \tag{1.4}$$

Here, roll dynamics are taken into account, which gives a better approximation of the LTR when compared to the $\mathrm{LTR_s}$. Instead of the lateral acceleration, what is needed here are the roll angle and the roll rate. The roll angle cannot be measured directly and needs to be obtained in another way in order to make use of the $\mathrm{LTR_d}$. Dahmani et al [5] designed an observer to estimate the roll angle. This is a common method for obtaining the roll angle in the literature [4] [6] [7]. Another option, as shown by Phanomchoeng and Rajamani [8], is to estimate the roll angle using roll angle from the tilt-angle sensor and the gyroscope roll rate.

Energy

A rollover index derived from energy considerations is another possibility. Choi [9] derives a rollover index based on the minimum potential energy of the vehicle required for rollover. Choi states that as long as the lateral kinetic energy, which can be quickly converted to potential energy, does not exceed the minimum potential energy threshold the vehicle is safe from rollover. With this, Choi defines a rollover index (equation 1.5) that requires lateral acceleration, longitudinal velocity, and sideslip angle.

$$\Phi_0 = \frac{1}{2}|v_x\beta|^2 - \sqrt{g^2 + a_y^2}\sqrt{(\frac{1}{2}T)^2 + h_{cg}^2} + \frac{1}{2}Ta_y + h_{cg}g \tag{1.5}$$

Johansson and Gafvert [10] developed a rollover index on the basis of a critical energy threshold. Similar to the concept of the LTR, they define that the moment the wheels of a vehicle lift off on one side is critical. The critical energy threshold is calculated with an optimization step in the transient case, which is then compared to the roll energy, which is a combination of kinetic and potential energy of the vehicle. The index, called 'Warning: Wheel Lift Off' ($\mathrm{W_{WLO}}$) is given by equation 1.6:

$$W_{WLO} = \frac{E_{crit} - E_{roll}}{E_{crit}} \tag{1.6}$$

If the $\mathrm{W_{WLO}}$ is less than 0, it indicates that one side of the vehicle has lifted off. It should be noted that to calculate the roll energy, the roll angle is needed, which can present problems as discussed in section 1.2.1

Similar to Johansson and Gafvert, Chen [11] developed a rollover index based on current energy and critical energy. In addition to that, Chen's rollover-detection index (RDI) also includes a critical energy rate and speed dependency, shown in equation 1.7:

$$RDI = c_1\frac{E}{E_{crit}} + c_2\frac{\dot{E}}{\dot{E}_{crit}} \tag{1.7}$$

Here c_1 and c_2 are speed-dependent constants.

The reviewed energy methods for rollover detection require the three main vehicle states roll angle, roll rate, and lateral acceleration as well. Specifically, the method described by Johansson and Gafvert [10] operates on the same principle as that of the LTR. That is to say that the rollover index is meant to detect the moment one side of the vehicle lifts off the ground.

Alternative Detection Methods

There are several alternative detection methods not based explicitly on either the LTR or energy. Some of these methods are outlined in this section.

Yoon and Yi [12] propose a rollover index derived through a roll angle and roll rate phase plane analysis, shown in Figure 1.3. Thresholds and critical values are determined for the roll angle, roll rate, and lateral acceleration. The resulting rollover index (equation 1.8) is a function of these thresholds, the lateral acceleration, roll angle, and roll rate. Yoon and Yi designed a model-based estimator to obtain the roll angle.

$$RI_{Yoon} = f(a_y, \phi, \dot{\phi}, a_{ycrit}, \phi_{thresh}, \dot{\phi}_{thresh}) \tag{1.8}$$

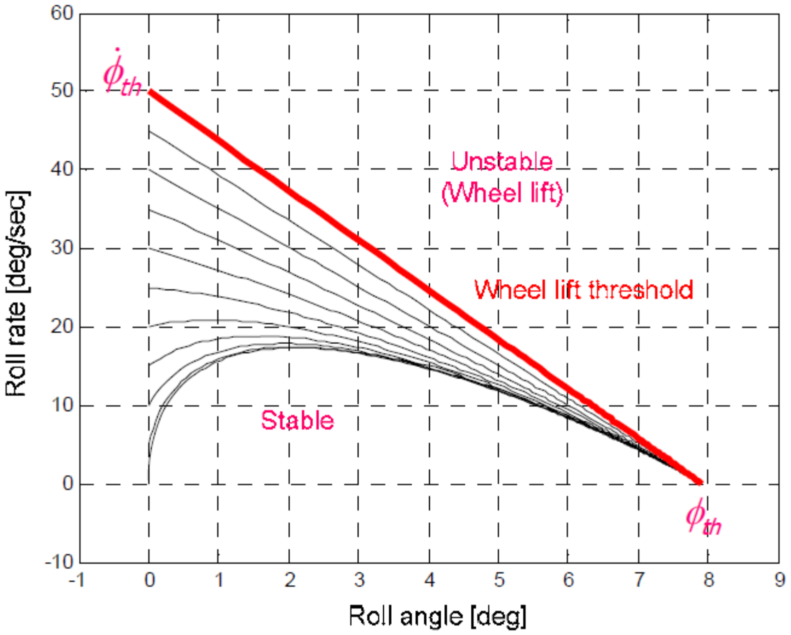


Figure 1.3: Wheel lift threshold phase plane [12]

Similar to Yoon and Yi, Hac et al [13] make use of a rollover index that is a function of the roll angle, roll rate, and lateral acceleration (equation 1.9). The shape of the function depends on vehicle characteristics such as the maximum lateral acceleration achievable on a dry surface in a steady-state turn. The parameters are fine-tuned through simulation and testing. Figure 1.4 shows this rollover index during a Fishhook maneuver. During normal driving conditions, the value is zero and gets higher as the danger of rollover increases. The activation level for chassis control was set at 0.2 in this case.

$$RI_{Hac} = f(a_y, \phi, \dot{\phi}) \tag{1.9}$$

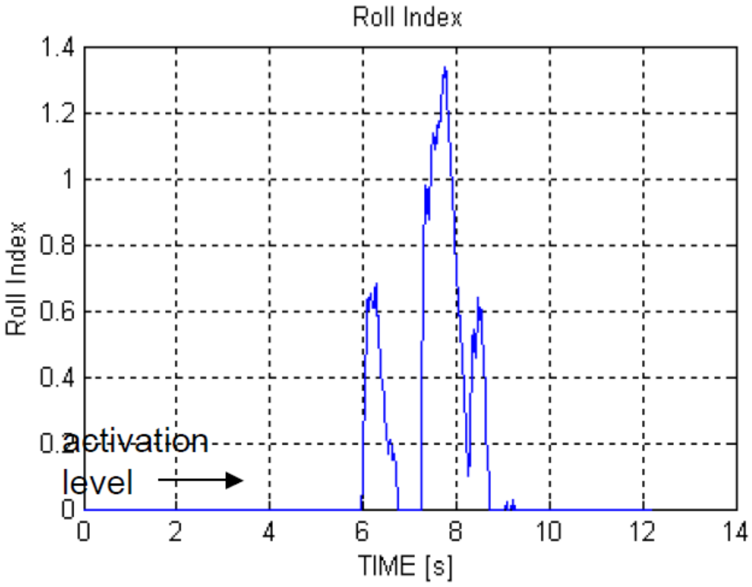


Figure 1.4: Rollover index from Hac et al during a Fishhook maneuver [13]

Carlson and Gerdes [14] developed a controller for rollover prevention using Model Predictive Control. The objective of the controller is to limit the peak roll angle, which is what acts as the indicator for rollover. Carlson and Gerdes estimate the roll angle with a GPS attitude system with gyroscopic measurements. Using the roll angle directly as an indicator for rollover is also a method utilized by Cherian et al [15].

Trent and Greene [16] use a critical tire deflection as an indicator for rollover. A genetic algorithm predictor estimates a tire deflection value that will lead to rollover. This is compared against value of the tire deflection calculated with vehicle speed, lateral acceleration, roll rate, and roll angle. If the calculated value is higher than the estimated critical value, it indicates the onset of rollover.

These rollover detection methods also make use of the three main vehicle states governing rollover behavior: lateral acceleration, roll angle, and roll rate. Yoon and Yi [12] and Hac et al [13] both devised rollover indices as a function of roll angle, roll rate, and lateral acceleration. Another, simpler, possibility is to use only the roll angle similar to Carlson and Gerdes [14]. In most scenarios using only the roll angle can be sufficient for rollover detection provided the critical roll angle is well defined. However, this method is not as robust as the other discussed methods because it neglects the effects of lateral acceleration and roll rate. An example of a scenario where roll angle is not sufficient is given by Hac et al as when a vehicle driven straight hits a bump on one side. This can cause the peak roll angle to reach critical values but does not mean it is in danger of rollover.

1.2.2. Rollover Prediction

Rollover prediction methods exist and have advantages over rollover detection methods. Predicting the onset of rollover serves as an early warning and thereby gives either the driver or the controller additional time to act.

Model-based methods

Chen and Peng [6] developed the time-to-rollover (TTR) metric for rollover prediction. The future vehicle state is predicted using a model and current state information such as the steering wheel angle and vehicle speed. The predicted states are then evaluated with a rollover indicator such as the LTR to check for possible rollover. This returns the TTR, which can serve as an early indicator of rollover. The algorithm for the TTR is represented with a flow chart in Figure 1.5. This method is often used in the literature for various control schemes [7] [17] [18]. Zhu et al [19] expand upon the TTR by utilizing a neural network to correct the estimated TTR. Figure 1.6 shows the TTR algorithm flow chart from Zhu et al, which uses the LTR for rollover detection.

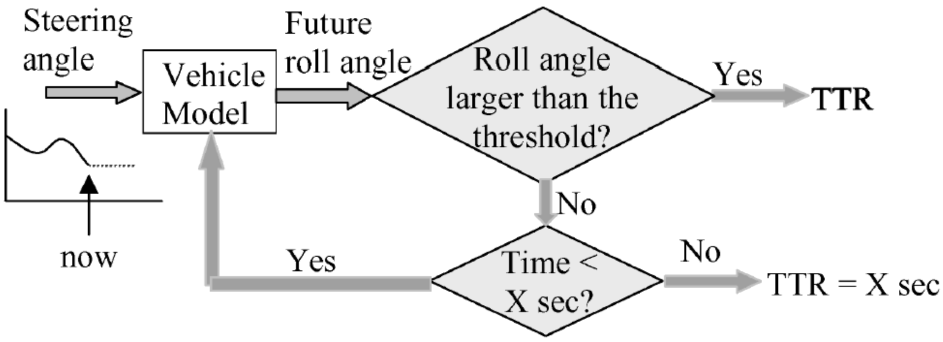


Figure 1.5: TTR algorithm flow chart [6]

A similar method is employed by Sanchez et al [20], where a neural network is used for a state predictor.

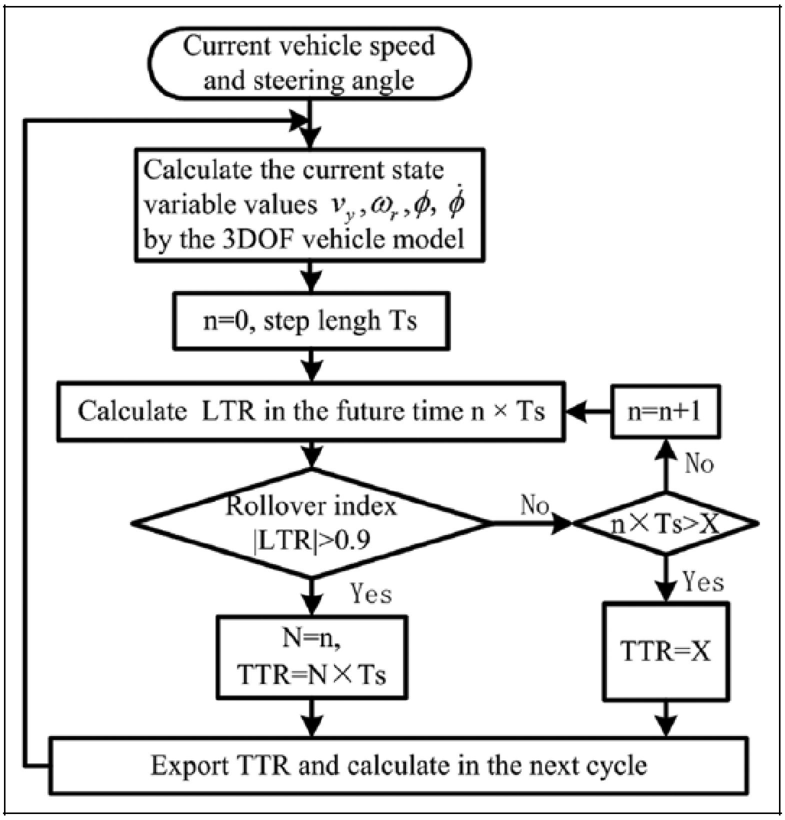

Current vehicle speed and steering angle
Calculate the current state variable values $v_y, \omega_r, \phi, \dot{\phi}$ by the 3DOF vehicle model
n=0, step lengh Ts
Calculate LTR in the future time n × Ts
n=n+1
No
Rollover index |LTR|>0.9
No
n×Ts>X
Yes
Yes
N=n, TTR=N×Ts
TTR=X
Export TTR and calculate in the next cycle

Figure 1.6: TTR algorithm flow chart using LTR [19]

The predicted values for roll angle and lateral acceleration is used for rollover prevention control. Results from Sanchez et al, seen in Figure 1.7, are promising and can potentially be used to detect for rollover.

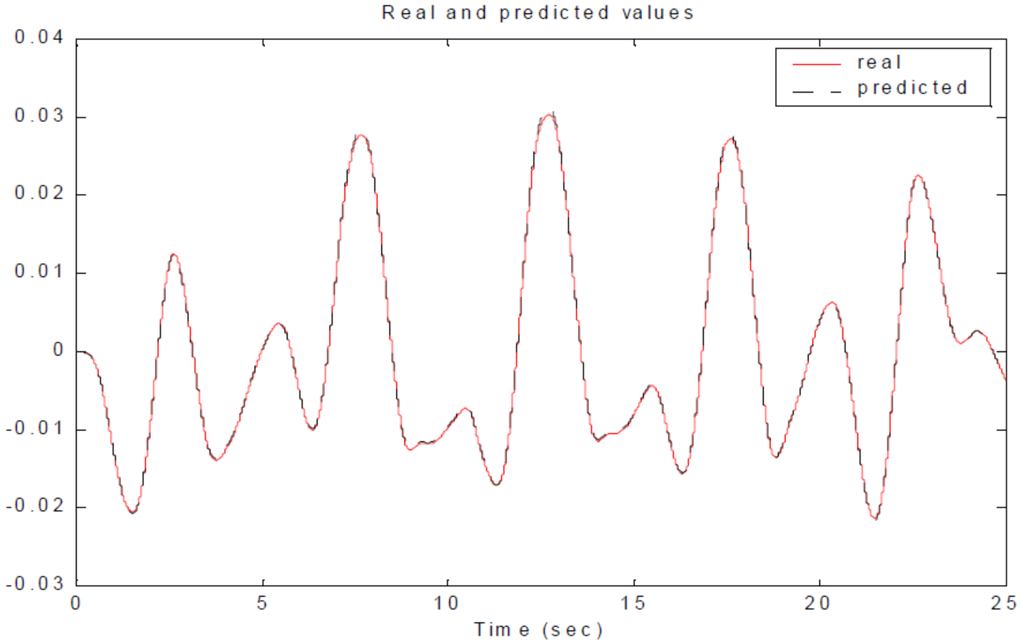


Figure 1.7: Simulated and predicted roll angle with a neural network state predictor [20]

Another variation of this idea is one by Chen and Huang [21]. Here, the vehicle states are predicted using not only current vehicle states, but also the road geometry and a driver model. The rollover index developed by Yoon and Yi [12] is used to detect rollover with the predicted vehicle states.

Finally, as mentioned in section 1.2.1, the method shown by Trent and Greene [16] uses a genetic algorithm predictor to estimate a critical tire deflection value that will lead to rollover several timesteps into the future.

In essence, all these methods develop a model of the vehicle that is capable of predicting future vehicle states based on the current conditions. The predicted future states then detect for rollover with one of the aforementioned methods, which in that case would predict the onset of a rollover accident. The issue with this is that the model needs to run much faster than real-time, depending on the prediction horizon. However, simplifying the model significantly to reduce computational load can lead to inaccuracies. A linearized model is relatively light in terms of computational load, but rollover accidents often occur in the nonlinear region of handling, which can cause the model to incorrectly predict rollover. These model-based methods have the potential to be useful as long as the right balance between complexity and speed is found.

Predictive LTR

Larish et al [22] proposed a predictive LTR (PLTR) that can predict the LTR accurately up to 100 ms in advance. The PLTR is composed of the LTR and the derivative of the LTR (equation 1.10). The derivative is multiplied with Δt, which determines the length of the preview time. Larish et al utilize linear approximation and small angle assumption to rewrite the equation into a form that requires the roll angle and steering wheel angle, along with easy obtainable signals such as the lateral acceleration, shown in equation 1.11.

$$PLTR(\Delta t) = LTR + L\dot{T}R \cdot \Delta t \tag{1.10}$$

$$\begin{aligned} PLTR(\Delta t) = \frac{2h_{cg}}{T}(\frac{a_y}{g} + sin\phi) + \frac{2h_{cg}}{Tg}[a_y + \frac{-C_0(a_y - rv_x) - C_1\dot{r}}{mv_x} + \ldots \\ \ldots \frac{2Cf}{m}\frac{1}{SR}\dot{\delta}_d + g\dot{\phi}] \cdot \Delta t \end{aligned} \tag{1.11}$$

The equation from Larish et al also includes terms used to smooth out noisy signals, which has been left out for simplicity. The C_0 and C_1 terms in equation 1.11 are given by:

$$\begin{aligned} C_0 &= 2C_{\alpha f} + 2C_{\alpha r} \\ C_1 &= 2l_f C_{\alpha f} - 2l_r C_{\alpha r} \end{aligned} \tag{1.12}$$

1.2.3. Rollover Prevention Controllers

Utilizing one of the aforementioned rollover detection or prediction methods, one can design a controller to prevent the onset of rollover. With the imminent onset of rollover known or predicted, a controller can act on this information and take preventative action. Several controller designs and stabilization methods have been proposed in the literature.

Differential Braking

The most common approach for preventing rollover is the use of active differential braking. Examples include Solmaz et al. [23], Carlson and Gerdes [14], Yoon and Yi [12], and Choi [9]. By utilizing differential braking a controller is able to reduce both the lateral acceleration and add an additional corrective yaw moment to the vehicle. This corrective yaw moment can help counteract the yaw moment from a vehicle turning too aggressively, thereby improving the lateral stability of the vehicle. With the advent of electric vehicles, a variation of differential braking is individual motor torque control or torque vectoring, which is used by Ataei [24] for example. This is possible with electric motors because each wheel can be driven independently. An advantage of this is greater control over the change in lateral acceleration and yaw moment and is also less likely to cause the wheels to lock with aggressive braking.

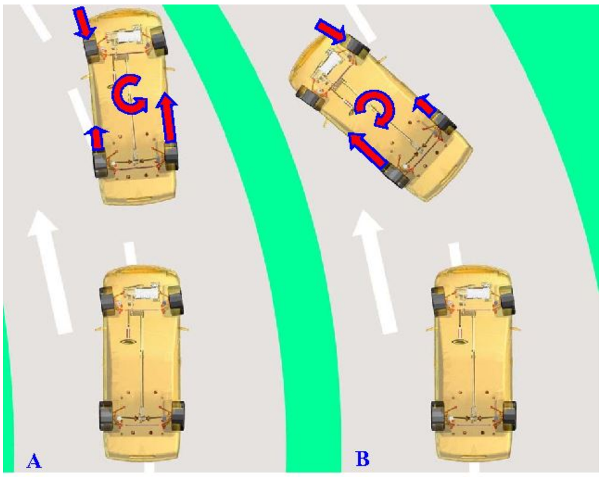


Figure 1.8: Differential Braking Demonstration

Active Steering

Active steering for rollover prevention is proposed by Yim [25], Solmaz et al. [4], and Shao et al. [26]. Instead of reducing tire forces with braking, active steering can change the tire forces by adjusting the wheel's steering angle. It can also work more aggressively by changing the vehicle trajectory altogether to avoid a dangerous situation that could lead to a rollover accident. Active steering for rollover is a more efficient method compared to differential braking [27] and offers more comfort for the passenger. On the other hand, differential braking is easier to implement because the existing hardware in vehicles is enough. Active braking needs to have steer-by-wire system or specially installed equipment to enable it.

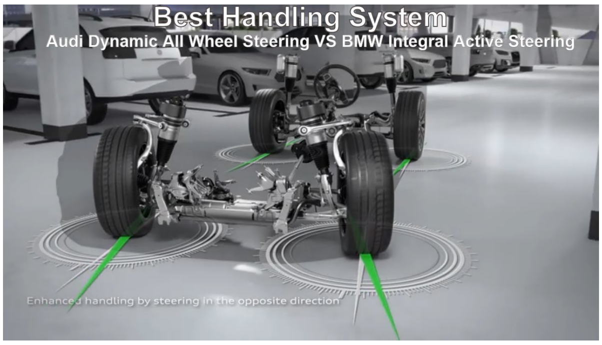

Best Handling System
Audi Dynamic All Wheel Steering VS BMW Integral Active Steering
Enhanced handling by steering in the opposite direction

Figure 1.9: Active Steering System

Active Suspension

Finally, utilising an active suspension as shown by Yim [28] and Riofrio [29] is a possibility as well. The active suspension can influence the roll dynamics of the vehicle by changing the stiffness of the suspension. This can effectively prevent the vehicle from rolling over, but as a result the maneuverability of the vehicle deteriorates. According to Lee et al. [30], utilising an active suspension will make the vehicle tend to oversteer. This is undesirable handling behavior and needs to be compensated for when the active suspension is working.

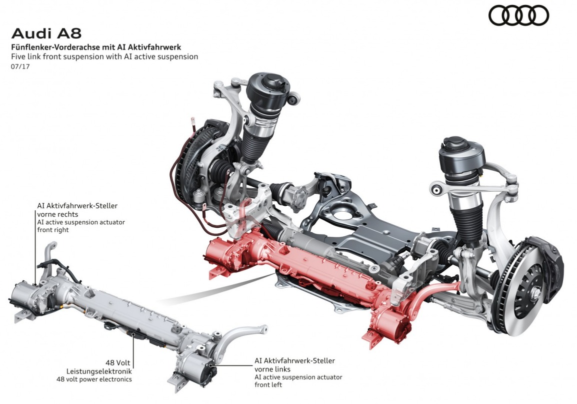

Audi A8
Fünflenker-Vorderachse mit AI Aktivfahrwerk
Five link front suspension with AI active suspension
07/17
AI Aktivfahrwerk-Steller vorne rechts
AI active suspension actuator front right
48 Volt Leistungselektronik
48 volt power electronics
AI Aktivfahrwerk-Steller vorne links
AI active suspension actuator front left

Figure 1.10: Active Suspension System

Control Methods

Regardless of which approach is used to stabilize the vehicle, a control method is needed to determine the timing and magnitude of the control input. In the past, controllers designed for rollover prevention have often used simple PID controllers [11] [23] [22] or LQR controllers [31] [29]. Across the reviewed controllers in the literature, a controller structure that is being used more often in recent years the Model Predictive Controller (MPC) [32] [24]. An MPC controller can offer better, more optimal control compared to PID controllers and LQR controllers. It models the system that is being controlled and uses that to deliver an optimal control sequence keeping in mind the objective and potential future states of the system. By keeping the potential future states in mind, an MPC controller can take action earlier and not just reactively.

1.3. Problem Definition

The previous section shows that it is difficult to have a vehicle rollover prevention controller that can act in a timely manner and is at the same time not too computationally expensive so that it can run in real-time. There is generally a trade-off between accuracy and speed. It is possible to use advanced sensors such as wheel force transducers to eliminate this issue. However, that is impractical and costly to be considered for production vehicles.

Therefore, a controller that uses a simple but accurate detection method and can act ahead of the potential accident is needed. Such a controller can be utilized for production vehicles, especially for taller vehicles such as SUVs, delivery vans, and trucks. As mentioned previously, MPC controllers can deliver optimal control inputs while simultaneously respecting constraints on the vehicle system. An MPC controller is typically more computationally expensive compared to reactive controllers such as a PID controller. In order to maintain real-time performance, a simple detection method is needed. In this case, the predictive load transfer ratio (PLTR) is used for rollover detection. The controller should also use differential braking for simplicity along with the MPC framework. A rollover prevention controller that combines MPC with PLTR should then be capable of real-time rollover prevention. Vehicles with a high propensity for rollover accidents could be much safer with such a rollover prevention system.

1.4. Contributions

The contribution of this thesis is an MPC controller that can prevent rollover incidents with differential braking. The controller combines rollover prediction through PLTR and active control with the MPC. This method of rollover detection allows for a simple model in the controller, thereby improving computational efficiency and cost. This then enables practical and real-time use for the proposed controller.

1.5. Thesis Layout

The paper is structured in the following manner:

- Chapter 1: Provides the background and motivations for the topic of rollover. Rollover detection and rollover controllers are discussed and compared.
- Chapter 2: The dynamical models used for the controller and simulation are introduced. Different tire models are discussed. A linear bicycle model is formulated for the controller's use. The IPG CarMaker model used for the vehicle plant model is also explained.
- Chapter 3: The controller for rollover prevention is discussed. Model predictive control theory is expanded upon. The specific MPC controller design is the formulated. The reference generator and benchmark controller are also presented.
- Chapter 4: The testing method and maneuver are introduced here. Then, simulation results are shown and discussed.
- Chapter 5: Conclusions are given from the results of the experiments. Finally, some recommendations for improvements and future research are given.

2

Vehicle Model

The vehicle model is essential for several reasons. The first is that a mathematical model of the vehicle with dynamic equations is necessary for controller design. This mathematical model is also necessary for the rollover detection. In this thesis, the single-track vehicle model also known as the bicycle model is used for this purpose. The bicycle model is widely used for vehicle modelling for its simplicity. Despite its simplicity, the bicycle model has reasonable accuracy, especially in the linear operating region.

A vehicle plant model is also needed for the simulation and testing of the controller. Simulating the controller's performance in a virtual environment with a complex and precise multi-body vehicle model allows rapid prototyping of the controller without the need for real world tests. This can save a large amount of time and resources. In this work, the vehicle model from IPG CarMaker is used.

2.1. Coordinate System

For modelling the vehicle, the ISO 8855 [33] standard coordinate system is used. This is shown in Figure 2.1. The coordinate system originates from the vehicle's center of gravity as has the following axes:

- X-axis: Points towards the front of the vehicle; rotation along this axis is roll motion
- Y-axis: Points towards the left of the vehicle when facing the front; rotation along this axis is pitch motion
- Z-axis: Points up from the vehicle; rotation along this axis is yaw motion

The coordinate system follows the right-hand rule, meaning that counter-clockwise rotation in the direction of the axis is defined to be positive.

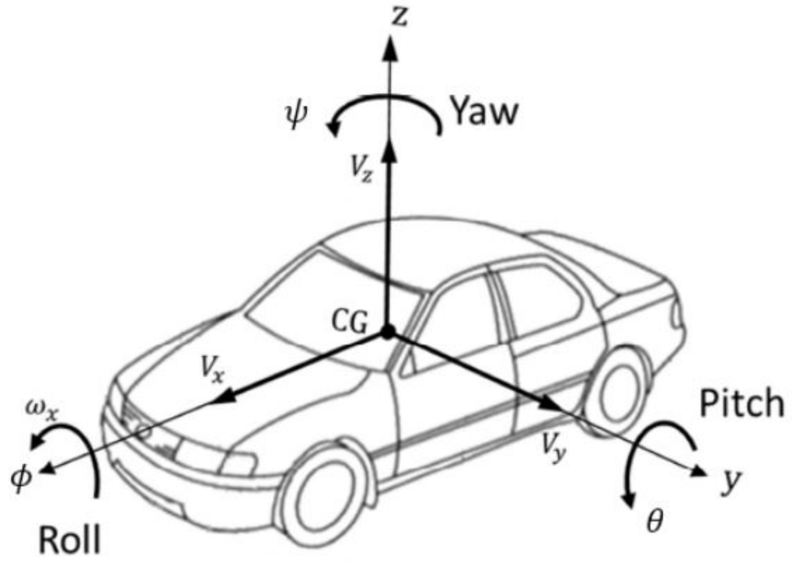


Figure 2.1: Vehicle Coordinate System [34]

2.2. Tire model

As the first point of contact between the vehicle and the ground, the tires are hugely influential in the dynamics of the vehicle. There are several ways with which tire forces can be modelled with varying degrees of accuracy and complexity. There are simple equation based models such as the Dugoff model [35], which can be used to calculate longitudinal tire forces, lateral tire forces, or combination tire forces. There are also more complex models such the Magic Formula by Pacejka [36] which is used in the vehicle model in IPG CarMaker. The benefit of such complexity is that the resulting tire forces from the Magic Formula are much more precise, especially in the nonlinear operating regions, and are therefore more suited for use in simulation software such IPG CarMaker. The Magic Formula is given as follows:

$$y(x) = Dsin(Carctan(Bx - E(arctan(Bx)))) \tag{2.1}$$

where the output $y(x)$ can be either the longitudinal tire force or the lateral tire force depending on if the input x is the longitudinal slip or the side slip angle. The rest of the variables are:

- D = Peak factor
- C = Shape factor
- B = Stiffness factor
- E = Curvature factor

The Magic Formula is an empirical model and is fitted to specific tire test data. The two additional variables, the horizontal shift S_h and the vertical shift S_v, are needed to calculate the final longitudinal force or lateral force. The equation is as follows:

$$Y(x) = y(x) + S_v \tag{2.2}$$

$$x = X + S_h \tag{2.3}$$

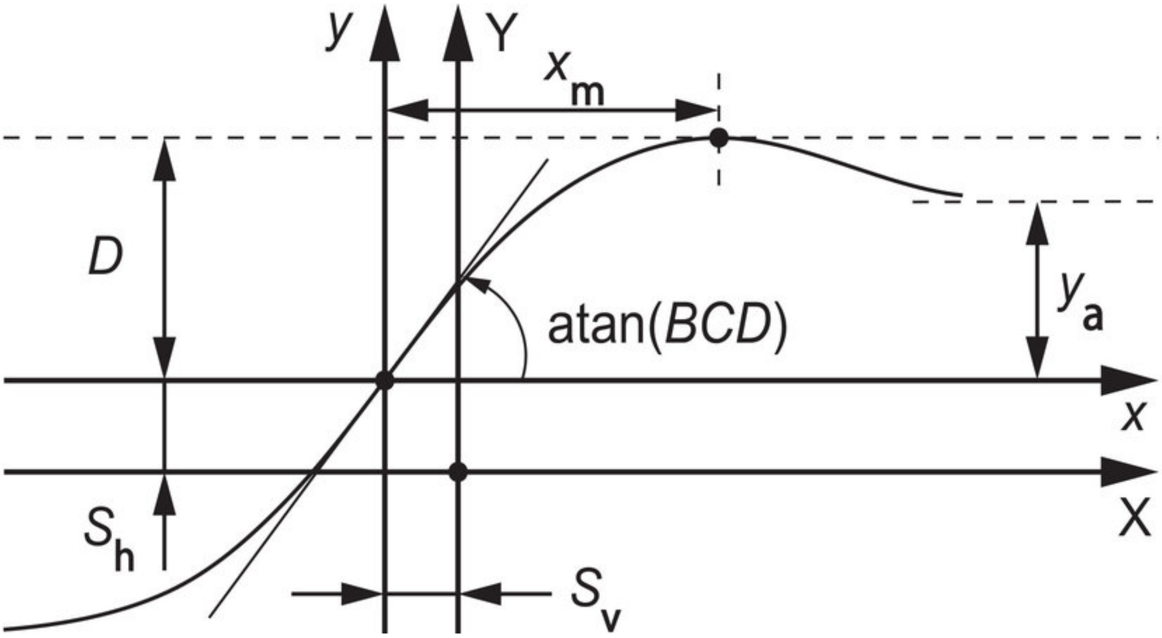


Figure 2.2: Magic Formula Curve

For the purposes of the MPC controller, the tire model used in this work is the linear tire model, where there is a linear relationship between the tire forces and longitudinal slip or slip angles. The bicycle model is being used and one of the assumptions is that there is no longitudinal acceleration. Due to that it is not necessary to consider longitudinal tire force and longitudinal slip. Equation 2.4 shows the linear tire model equation for lateral tire forces and side slip angles:

$$F_{yi} = C_{\alpha_i} \alpha_i \tag{2.4}$$

The side slip angle α_i for a given tire is:

$$\alpha_i = \delta_i - \frac{v + a_i r}{v_x} \tag{2.5}$$

$$a_i = \begin{cases} l_f & \text{Front tires} \\ -l_r & \text{Rear tires} \end{cases} \tag{2.6}$$

where:

- F_{yi}: Lateral tire force
- C_{α_i}: Tire cornering stiffness
- α_i: Side slip angle
- δ_i: Steering angle
- v: Lateral velocity
- r: Yaw rate
- v_x: Longitudinal velocity

- l_f: Length from front axle to vehicle center of gravity
- l_r: Length from rear axle to vehicle center of gravity

The linear tire model is suitable for this application due to its simplicity, since this keeps the computational load light in order to achieve real-time performance. Furthermore, the rollover indicators used in the controller were developed using a linear tire model as well.

2.3. Bicycle model

For the MPC controller, a set of differential equations for the vehicle dynamics is needed. To obtain these equations, the linear bicycle model is used as a basis. The model is shown in Figure 2.3. This two degree-of-freedom linear bicycle model has a few assumptions, namely:

- Longitudinal velocity is constant
- Roll, pitch, and vertical motion is neglected
- Ideal steering dynamics
- No suspension
- Small steering angle such that $sin\delta = \delta$

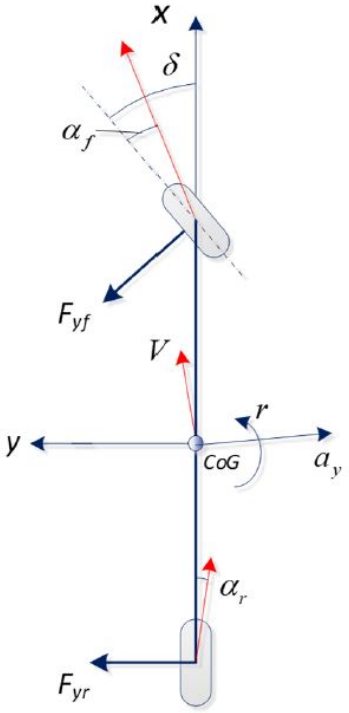


Figure 2.3: Linear Bicycle Model

Using this, the Newton-Euler equations for vehicle motion are given by:

$$ma_x = 0 \tag{2.7a}$$
$$ma_y = m(\dot{v} + v_x r) = \Sigma F_y \tag{2.7b}$$
$$I_{zz}\dot{r} = \Sigma M_z \tag{2.7c}$$

For small steering angle δ and ignoring tire self-aligning moments, we get:

$$\Sigma F_y \approx F_{yf} + F_{yr} \tag{2.8a}$$
$$\Sigma M_z = l_f F_{yf} - l_r F_{yr} \tag{2.8b}$$

By using the linear tire model and thus equations 2.4 and 2.5, we substitute the lateral forces in the equation to get:

$$m(\dot{v} + v_x r) = F_{yf} + F_{yr} = C_{\alpha f}(\delta_f - \frac{v + l_f r}{v_x}) + C_{\alpha r}(-\frac{v - l_r r}{v_x}) \tag{2.9a}$$
$$I_z \dot{r} = l_f F_{yf} - l_r F_{yr} = l_f C_{\alpha f}(\delta_f - \frac{v + l_f r}{v_x}) - l_r C_{\alpha r}(-\frac{v - l_r r}{v_x}) \tag{2.9b}$$

Rewriting the equations, we can put the bicycle model into state-space form:

$$\dot{x} = Ax + B_1\delta \tag{2.10}$$

$$x = \begin{bmatrix} v \\ r \end{bmatrix} \tag{2.11}$$

$$A = \begin{bmatrix} \frac{-C_{\alpha f} + C_{\alpha r}}{m v_x} & \frac{C_{\alpha r} l_r - C_{\alpha f} l_f}{m v_x} - v_x \\ \frac{C_{\alpha r} l_r - C_{\alpha f} l_f}{I_{zz} v_x} & \frac{l_r^2 C_{\alpha r} - l_f^2 C_{\alpha f}}{I_{zz} v_x} \end{bmatrix} \tag{2.12}$$

$$B_1 = \begin{bmatrix} \frac{C_{\alpha f}}{m} \\ \frac{l_f C_{\alpha f}}{I_{zz}} \end{bmatrix} \tag{2.13}$$

One more addition is needed for the MPC controller. Differential braking is used for stabilization, so total braking force u which represents the difference in braking force between the right and left sides is added to the model. The differential braking force acts on the system as a moment around the z-axis of the vehicle, thereby influencing the yaw motion. This is shown in Figure 2.4. With this addition, the system looks like this:

$$\dot{x} = Ax + B_1\delta + B_2 u \tag{2.14}$$

$$B_2 = \begin{bmatrix} 0 \\ \frac{-T}{2I_{zz}} \end{bmatrix} \tag{2.15}$$

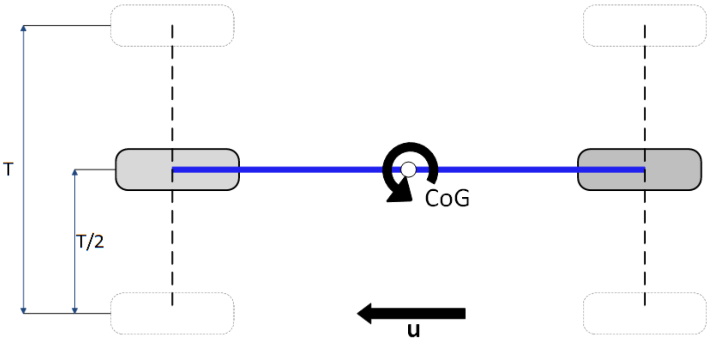


Figure 2.4: Differential Braking Force as control input

2.4. Summary

The modelling of the vehicle model used in the controller is described in this chapter. Various tire models are shown such as the the Dugoff model and the Magic Formula. Among the different tire models, the linear tire model is used for the controller vehicle model. Subsequently, the bicycle model is introduced and the vehicle equations of motion are derived from this. Finally, differential braking is added as a control input to the equations of motion since it is being used in the controller. This set of differential equations can then be used in the proposed controller which is explained in the next chapter.

3

Controller Design

3.1. Overview

A simplified overview of the proposed rollover prevention system is shown in Figure 3.1. From the vehicle, the vehicle states are obtained. In this case, it comes directly from IPG CarMaker, but in practice needs to be measured or observed using accelerometers for example. The current vehicle states are used for the reference generator and the controller. From the reference generator we get a reference that needs to be tracked, which is also used for the controller. Using this information, the controller then gives a control input, which is converted to individual brake torques on each of the four wheels on the vehicle. This command is given to the plant, which in this case is the IPG CarMaker model. Finally, the plant gives as an output updated vehicle states, which then subsequently feed into the reference generator and controller again, closing the loop. The vehicle plant model has already been discussed in the previous chapter. In this chapter, the controller, reference generator, and control allocation is elaborated upon.

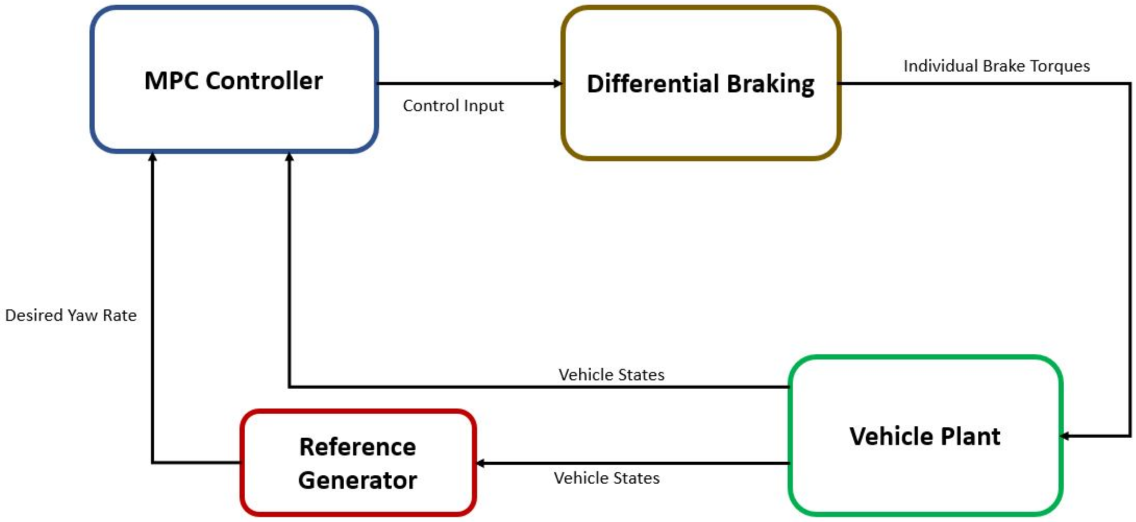


Figure 3.1: System Overview

3.2. Model Predictive Control

The controller is the key component for rollover prevention. Several different types of controller structures are available, such as PID controllers and LQR controllers. The chosen control structure in this work the model predictive controller. Each type of controller has its unique properties, traits, advantages, and disadvantages. The properties and advantages offered by MPC controllers are appealing for the set of circumstances and demands of a rollover prevention controller.

Using an MPC framework can generate an optimal control input sequence. One major advantage an MPC controller has over a controller such as a PID controller is the ability to easily incorporate state and actuator constraints into the problem formulation, thereby keeping the vehicle and actuator within realistic and feasible bounds. Additionally, using an MPC controller as opposed to a PID controller means the tuning process can be expedited. PID controllers need extensive tuning whereas the MPC controller has fewer tuning parameters.

This section will expand upon the theory behind model predictive control and its key components. Additionally, the MPC software implementation will be discussed as well.

3.2.1. MPC framework

Model predictive control is a model based method for system control. As mentioned in chapter 1, it has become a popular method for control in recent years. It has been in use for a longer time in the process industry for chemical plants, since the dynamics of those systems are much slower. Recent hardware advancements have made it possible to use for a real-time vehicular rollover prevention controller. The MPC has two primary components. The first is the prediction model, which can be either a linear or nonlinear dynamic model of the system. This model is used to predict the likely future evolution of the system [37]. The prediction is done over a finite time period, which is known as the prediction horizon.

The second part is the optimization algorithm since MPC is a form of optimal control. The system is optimized over the prediction horizon with a sequence of control inputs. The optimization algorithm is solved with a certain user-defined control law. The cost function and system constraints here determine the controller's behavior, which returns a control input for each timestep over the prediction horizon. Only the first control input in the sequence is applied on the system. The process is then repeated with updated states at the next time interval. The prediction horizon is constantly being shifted forward with each iteration and due to this MPC is also known as receding horizon control. A graphical representation of the MPC scheme is shown in Figure 3.2.

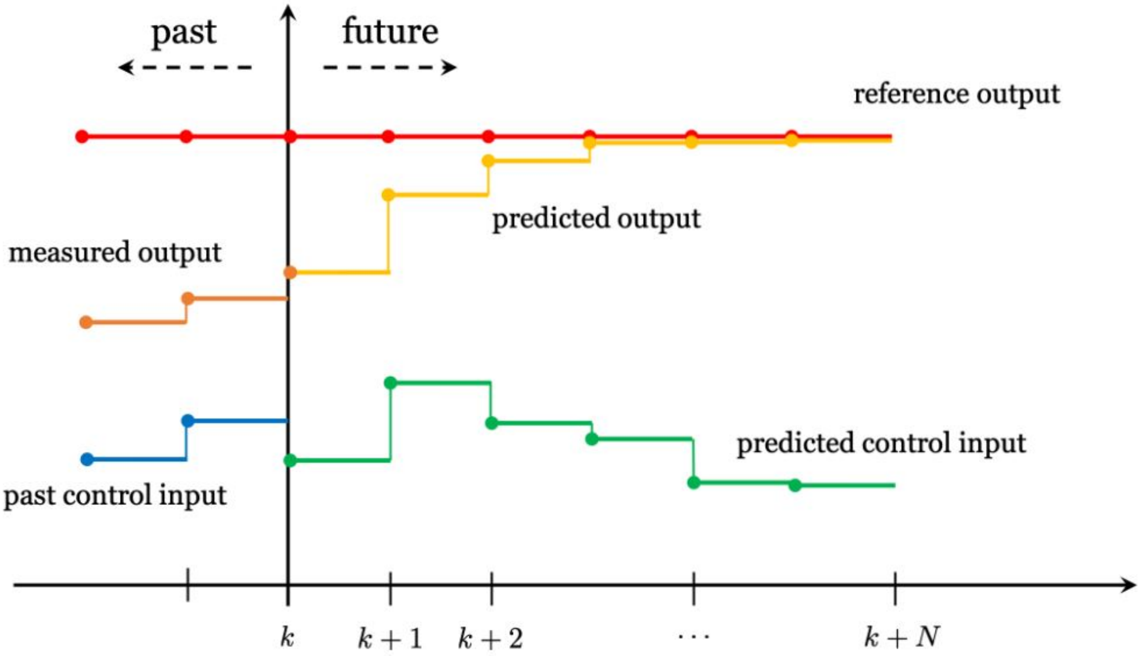


Figure 3.2: Model Predictive Control Receding Horizon Policy

Prediction Model

The prediction model is essential for the MPC to work well. The better the quality of the model, the better the performance of the MPC controller. The vehicle is a nonlinear system, especially at the limits of handling. However, a linear model is used in this work considering the goal is to have the controller work in real time. The literature also shows that a linear model in the MPC is a good approximation and performs adequately. The general linear time-varying model for MPC has the following form:

$$\frac{dx}{dt} = A(t)x + B(t)u \tag{3.1}$$

$$y = C(t)x + D(t)u \tag{3.2}$$

$$x(0) = x_0 \tag{3.3}$$

in which $A(t) \in \mathbb{R}^{nxn}$ is the state transition matrix, $B(t) \in \mathbb{R}^{nxm}$ is the input matrix, $C(t) \in \mathbb{R}^{pxn}$ is the output matrix, and $D(t) \in \mathbb{R}^{pxm}$. In practice $D = 0$ most of the time.

Optimal Control Problem

The optimal control problem (OCP) is necessary so that the MPC controller returns a control input that is optimal and keeps the system stable. The generalized OCP $\mathbb{P}$ is the following form:

$$
\begin{aligned}
\underset{x,u}{\text{minimize}} \sum_{k=0}^{N-1} \ell(x(k),u(k)) + V_f(x(N)) \\
\text{subject to} \\
x(0) = x_0 \\
x(k+1) = f(x(k),u(k)), \forall k \\
(x(k),u(k)) \in \mathbb{Z}, \forall k \\
x(N) \in \mathbb{X}_f
\end{aligned}
\tag{3.4}
$$

Here N is the length of the prediction horizon and k is the current timestep of the MPC problem. The cost function is minimized over the entire prediction horizon. This gives us the objective function with which the OCP can be solved. ℓ is the stage cost function, applying a cost to each timestep of the prediction horizon. Finally, V_f is the terminal cost function. This again demonstrates the receding horizon policy, as the MPC controller is optimizing over the entire prediction horizon and not just the next timestep. This allows the MPC controller to take more efficient action compared to a purely reactive controller like a PID controller. Both the stage cost and the terminal cost will often be a quadratic cost function of the deviation of the states and input from its reference point. Additionally, all the optimization must adhere to the set constraints for the states and dynamics.

Solving the OCP can be done in several ways depending on the nature of the OCP. One advantage of having a linear MPC instead of a nonlinear MPC is that the optimization problem is convex instead of nonconvex. In the case of a convex optimization problem, every local minimum is a global minimum. This means that finding the global optimal solution can be found in polynomial time. Nonconvex problems are much more difficult, as not every local minimum is a global minimum. It is also often not possible to verify if a solution is the global minimum. It is therefore desirable to have a convex optimization problem, which can then be solved with convex optimization algorithms such as the cutting-plane algorithm, ellipsoid algorithm, or interior point algorithm.

3.2.2. FORCESPRO

Once the mathematical description of the problem has been set, the next step is to translate that into something that can be deployed and interface with the vehicle plant, which in this case is IPG CarMaker. The software used to implement the MPC formulation is FORCESPRO [38] from Embotech in conjunction with Matlab and Simulink. FORCESPRO enables the generation of a numerical optimization solver based on a high-level mathematical description of the problem. This generated solver can be easily adjusted and suitable for real-time applications with varying data. Figure 3.3 shows an overview of FORCESPRO.

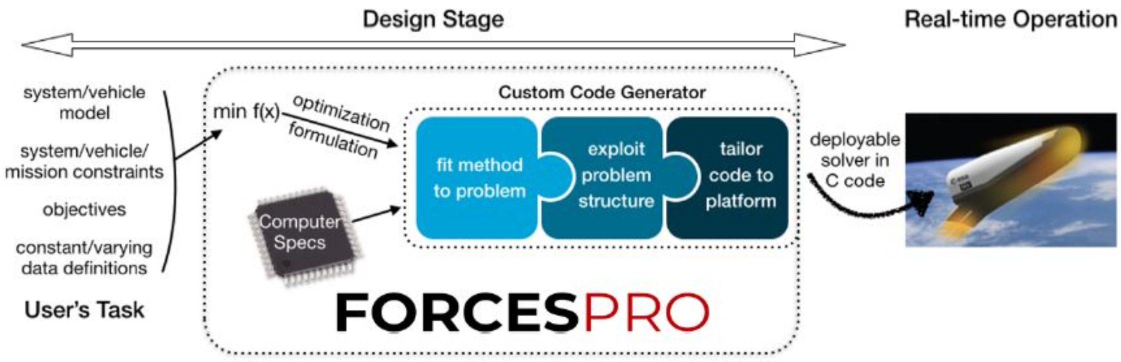


Figure 3.3: Overview of FORCESPRO

The generated code for the solver can only solve one optimization problem and is therefore much more efficient and smaller than general-purpose optimization solvers. On top of that, the generated code is libraryfree and uses no dynamic memory allocation, making it suitable for safe deployment on real-time systems. Using FORCESPRO and MatLab, a Simulink solver block based on the designed MPC controller can be generated. The parameters and settings for the MPC are simply defined in MatLab. First, the problem dimensions need to be defined. These are:

- Horizon length N
- Controller frequency T_s
- Number of variables including control
- Number of equality constraints
- Number of inequality constraint functions
- Number of parameters
- Number of differential states
- Number of control inputs

The horizon length N is the length of the prediction horizon and determines how far into the future prediction model and optimization problem of the MPC controller go. This is linked with the controller frequency, which determines the timestep size of the MPC controller. The horizon length multiplied with the controller frequency gives the prediction horizon in seconds. For the proposed controller, a controller frequency of 0.01 s is used. Based on numerical analysis and the literature [24] [32], the prediction horizon is set to 20 steps, resulting in a prediction horizon of 0.2 s.

The number of differential states, control inputs, and equality constraints are all dependent on the dimensions of the prediction model used in the MPC controller. The number parameters here indicate the number of online variables used in the MPC controller. For this, it is assumed that the following parameters are easily measured or estimated in a vehicle equipped with an ESP system:

- Longitudinal velocity
- Steering angle
- Reference yaw rate
- Lateral acceleration
- Steering speed
- Roll velocity
- Yaw acceleration

These signals are then taken directly from IPG CarMaker. After the problem dimension the dynamics of the system have to be defined. Here it is possible to input the dynamics in state-space form or differential equations. The equations used are the ones derived from the bicycle model in chapter 2. It is possible in FORCESPRO to input continuous dynamics, as the program will handle the discretization. Next, the constraints are defined. The boundary constraints on the state and input are simply given upper and lower bounds. The inequality function needs to be defined and then given upper and lower bounds as well. Finally, the solver options are defined. The options that can be changed include:

- Maximum number of iterations

- Integrator type
- Optimization solving method
- Sampling time of the integrator
- Number of integrator nodes
- Hessian approximation

The maximum number of iterations is, as the name suggests, the maximum number of iterations the solver will do to find the optimal solution. If it reaches the maximum, it will settle of the best solution found so far in those iterations. In practice, above a certain number of iterations the controller will not be able to run in real-time anymore. The integrator type is needed because we are dealing with numerical simulation of ordinary differential equations. The chosen method is the 4th-order explicit Runge-Kutta method (ERK4), which suffices for most applications. A table of the integrator options is shown below.

nlp.integrator.type	Type	Order
'ForwardEuler'	Explicit Euler Method	1
'ERK2'	Explicit Runge-Kutta	2
'ERK3'	Explicit Runge-Kutta	3
'ERK4'	Explicit Runge-Kutta	4
'BackwardEuler'	Implicit Euler Method	1
'IRK2'	Implicit Runge-Kutta	2
'IRK4'	Implicit Runge-Kutta	4

Table 3.1: Integrator Options

The optimization solving method is to specify which algorithm to use for the convex OCP. The options are:

- Nonlinear Primal-Dual Interior-Point method
- Sequential Quadratic Programming method
- Primal-Dual Interior-Point method
- Alternating Direction Methods of Multipliers
- Dual Fast Gradient method
- Fast Gradient method

The default option, which is the primal-dual interior-point method, is chosen as this is the most stable and robust method for most convex problems. The sampling time of the integrator is the same as the controller frequency. Next, the number of integrator nodes specifies the number of intermediate points between two timesteps during numerical integration of a continuous time model. More integrator nodes comes at the cost of computational time. Finally, we can specify the method used for the Hessian approximation. In FORCESPRO, there are two options for this: BFGS (Broyden-Fletcher-Goldfarb-Shanno) updates and Gauss-Newton approximation. With a least squares objective function, Gauss-Newton approximation can be used. This method leads to faster convergence and is more reliable when it can be applied. BFGS is a method for solving unconstrained nonlinear optimization problems [39] and should therefore only be used when Gauss-Newton approximation cannot be used. For the proposed controller, Gauss-Newton approximation is applicable and is therefore chosen.

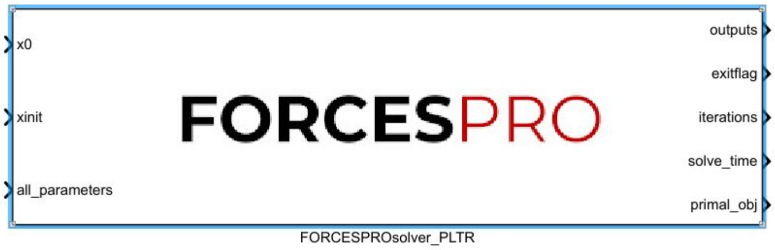


Figure 3.4: FORCESPRO MPC Block in Simulink

3.3. PLTR MPC

The controller to be designed is an MPC controller using the PLTR as the indicator for rollover. The PLTR, as discussed in chapter 1, is variation of the LTR that is able to give a time-advanced measure of the LTR and thereby the rollover propensity. The approximated and simplified equation for the PLTR is given by [22]:

$$
\begin{aligned}
PLTR(\Delta t) = \frac{2h_{cg}}{Tg}(1+kg)a_y + \frac{2h_{cg}}{Tg}[a_y + \frac{-C_0(a_y - rv_x) - C_1\dot{r}}{mv_x} + \ldots \\
\ldots \frac{2Cf}{m}\frac{1}{SR}\dot{\delta}_d + g\dot{\phi}] \cdot \Delta t
\end{aligned}
\tag{3.5}
$$

The k term in the equation is a constant used to approximate $sin(\phi)$ with the lateral acceleration and can be determined beforehand. This means that directly measuring and modelling vehicle roll behavior is not necessary for the PLTR, allowing the use of a much simpler model. This will help the computational efficiency of the controller. The bicycle model discussed in chapter 2 is used for this controller. The signals used to calculate the PLTR are the longitudinal velocity V_x, lateral acceleration a_y, yaw acceleration $\dot{r}$, roll acceleration $\dot{\phi}$, and driver steering angle δ. It is assumed that it is possible to either directly measure these signals or obtain them with a reasonable degree of accuracy with observers in a typical vehicle since these signals are also used for ESP systems.

With the differential equations derived in chapter 2, the MPC problem is set up as follows:

$$
\min_{x,u} \frac{1}{2}\sum(\bar{x}^T Q\bar{x} + u^T Ru) + \frac{1}{2}\bar{x}(N)^T P\bar{x}(N) \tag{3.6}
$$

Subject to:

$$
-u_{min} \leq u \leq u_{max} \tag{3.7}
$$

$$
-x_{min} \leq x \leq x_{max} \tag{3.8}
$$

$$
-LTR_{th} \leq PLTR \leq LTR_{th} \tag{3.9}
$$

Equation 3.6 is the objective function of the MPC controller. The goal is to minimize the value of this objective function. Here this means that the objective is to get the tracking error $\bar{x}$ as close to zero as possible while also keeping the control input as low as possible. The weights Q and R determine which part of the objective is prioritized. Equations 3.7 and 3.8 are the input constraints and state constraints, respectively.

These constraints are representative of the physical limitations of the system in question, which in this case is the vehicle. This ensures that the MPC controller does not attempt a braking force that is impossible to achieve and keeps the vehicle in safe operating conditions with respect to the states. Finally, equation 3.9 is a linear inequality constraint that integrates the PLTR into the MPC problem. With this the MPC controller will attempt to keep the calculated PLTR within the set bounds, thereby preventing rollover incidents.

The MPC controller has two objectives: first is to keep the vehicle operating within safe LTR boundaries to prevent rollover and second is to follow the driver's intentions as closely as possible. As mentioned previously, the tracking error $\bar{x}$ in the objective function serves this purpose and contains a term for tracking to desired yaw rate. The desired yaw rate is obtained from the reference generator. The higher priority lies with preventing rollover, as that is the more dangerous accident. However, as rollover accidents often occur during evasive maneuvers, it is still important that the driver's intentions are followed as closely as possible. The controller behavior can be modified with several parameters, with the most important ones being the weights in the cost function and the LTR threshold. The effects of these two parameters and their final chosen values will be examined further in chapter 4.

The output of the controller gives a desired braking force that acts on the left or right side of the vehicle as discussed in chapter 2. This desired brake force needs to be converted into a desired brake torque for which we use the following formula:

$$T_{br} = \frac{F_{br}}{R_w} \tag{3.10}$$

The desired brake torque from the controller is then divided evenly among the two wheels on the left or right side. That is to say, if the desired braking force on the right side is 1000 N, then the front right wheel would brake for 500 N and the rear right wheel would brake for 500 N. If the vehicle brakes on the left or right side is determined by the sign of the control input. With all this, it is then sent as a command to the vehicle plant to act on the system and prevent rollover.

Reference Generator

The reference generator gives as an output the desired yaw rate given a certain driver steering angle. The goal here is to take the driver's intentions into account and try to follow them as closely as possible. To do that, the desired steady-state steer angle input to yaw rate response is calculated based on the linear bicycle model with the following equation:

$$\frac{r}{\delta}|_{ss} = \frac{V_x}{L + \frac{K_{us} v_x^2}{g}} \delta \tag{3.11}$$

$$K_{us} = \frac{mg}{L}\left(\frac{l_r}{C_{\alpha f}} - \frac{l_f}{C_{\alpha r}}\right) \tag{3.12}$$

where K_{us} is the vehicle understeer gradient. With these equations it is possible to calculate the desired yaw rate of the vehicle given the vehicle's steering angle and longitudinal velocity. The desired yaw rate is then subsequently used by the controller as a reference to track.

3.4. Benchmark controller

In order to analyse the effectiveness of the proposed MPC controller an additional controller is used as a point of comparison. The same MPC controller using only the static LTR is utilized for this. As discussed in chapter

1, the static LTR is a simple but less accurate measure of the LTR. The PLTR is in essence an adjusted static LTR, as the first term in equation 3.5 is almost identical to the static LTR and adds another prediction term on top of that.

The equations governing the benchmark MPC controller are thus:

$$\min_{x,u}\frac{1}{2}\sum(\bar{x}^T Q\bar{x} + u^T Ru) + \frac{1}{2}\bar{x}(N)^T P\bar{x}(N) \tag{3.13}$$

Subject to:

$$-u_{min} \leq u \leq u_{max} \tag{3.14}$$

$$-x_{min} \leq x \leq x_{max} \tag{3.15}$$

$$-LTR_{th} \leq LTRs \leq LTR_{th} \tag{3.16}$$

The performance of this benchmark controller can then be compared against the proposed PLTR MPC controller. The reason other controller structures are not considered for the benchmark controller is because the advantages of MPC controllers for rollover prevention have been demonstrated in the literature such as by Yakub, Lee, and Mori [40]. By using the same MPC controller structure with the PLTR$_\text{s}$, we can evaluate the proposed PLTR MPC controller against the most simple implementation of an MPC rollover prevention controller.

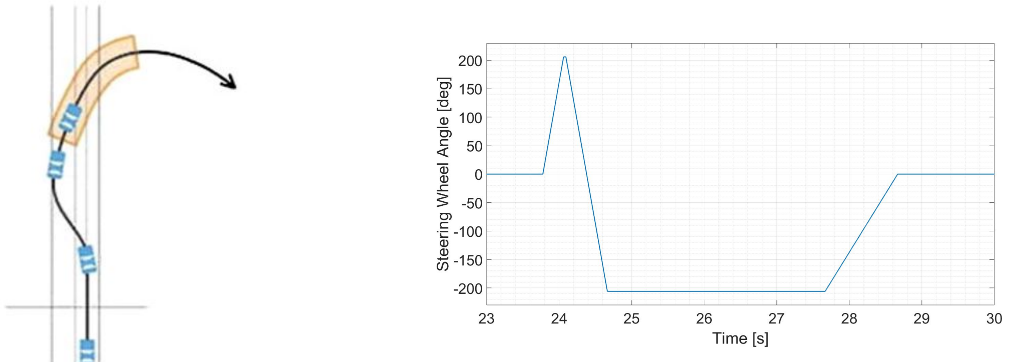
3.5. Summary

The design of the proposed rollover prevention controller is shown in this chapter. First, the general theory behind MPC controllers is presented, along with the software implementation of MPC controllers. The proposed MPC controller is designed with two goals: to prevent the onset of rollover and follow the driver's intentions as closely as possible. The desired yaw rate of the driver is obtained with a reference generator. The objective function of the MPC controller is set to drive the error between the reference yaw rate and actual yaw rate to 0 as a weighted least squares problem. In addition to that, the PLTR is used as an indicator for rollover. This metric is incorporated into the problem as a linear inequality constraint. The controller is thereby bound to keep the vehicle operating within these limits, thus preventing rollover. Finally, an additional controller is presented for the purpose of benchmarking the proposed MPC controller.

4 Simulation & Results

4.1. Driving Maneuver

The maneuver used to evaluate the controller's rollover prevention performance is the Fishhook maneuver. It is named this way because the trajectory of the vehicle during this maneuver resembles a fishhook, as shown in Figure 4.1a. Figure 4.1b shows the steering wheel angle to perform the maneuver. It is specially designed to induce rollover in vehicles and is therefore well-suited for this [41]. The maneuver is extremely aggressive and is capable of bringing the vehicle close to or over the critical rollover point. The maneuver has 3 phases. The first phase is the initial steer performed at an extremely high rate of 720 degrees per second until it reaches the maximum steering wheel angle which is determined beforehand. The second phase follows shortly after and is the counter steer in the opposite direction also performed at 720 degrees per second. The third phase comes after the counter steer and holds the steering wheel at the maximum steering wheel angle for 3 seconds after which it slowly returns to a neutral position.



(b) Fishhook Maneuver Steering Wheel Angle

(a) Fishhook Maneuver Trajectory [42]

Figure 4.1: Fishhook Maneuver

4.2. Plant Model

The vehicle plant model used for the test simulations is from IPG CarMaker. IPG CarMaker is a virtual vehicle test environment that models real-world test scenarios [43]. It includes a detailed vehicle model, intelligent driver model, and virtual roads. A visualisation of the virtual test environment is shown in Figure 4.2. The intelligent driver model helps with performing the fishhook maneuver to induce rollover. The method used by IPG CarMaker for vehicle modelling in shown in Figure 4.3. Figure 4.3a shows the internal and external forces that are considered in the model and Figure 4.3b shows the chain of calculation of the vehicle model. As mentioned in section 2.2, the tire model used in IPG CarMaker is the Magic Formula. When setting the model up a tire file needs to be specified. This file contains information on the tire such as its radius and the factors needed for the Magic Formula. The detailed vehicle model makes it suitable to be used as a vehicle plant model for testing the controller.

The results with the designed controller from IPG CarMaker are then a reasonable representation of a real-world test result. Finally, another advantage of using IPG CarMaker is that is provides vehicle state information that is typically difficult to obtain in a real-world scenario. For rollover, that means being able to access the tire forces directly without having to estimate them. The tire forces can be visualised such as in Figure 4.2 and can also be obtained numerically for further analysis.

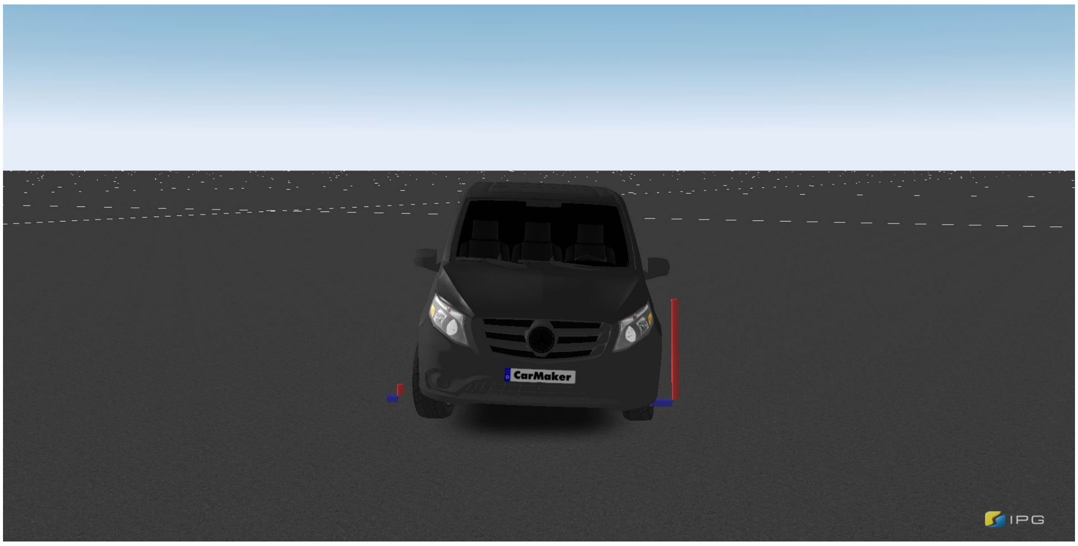
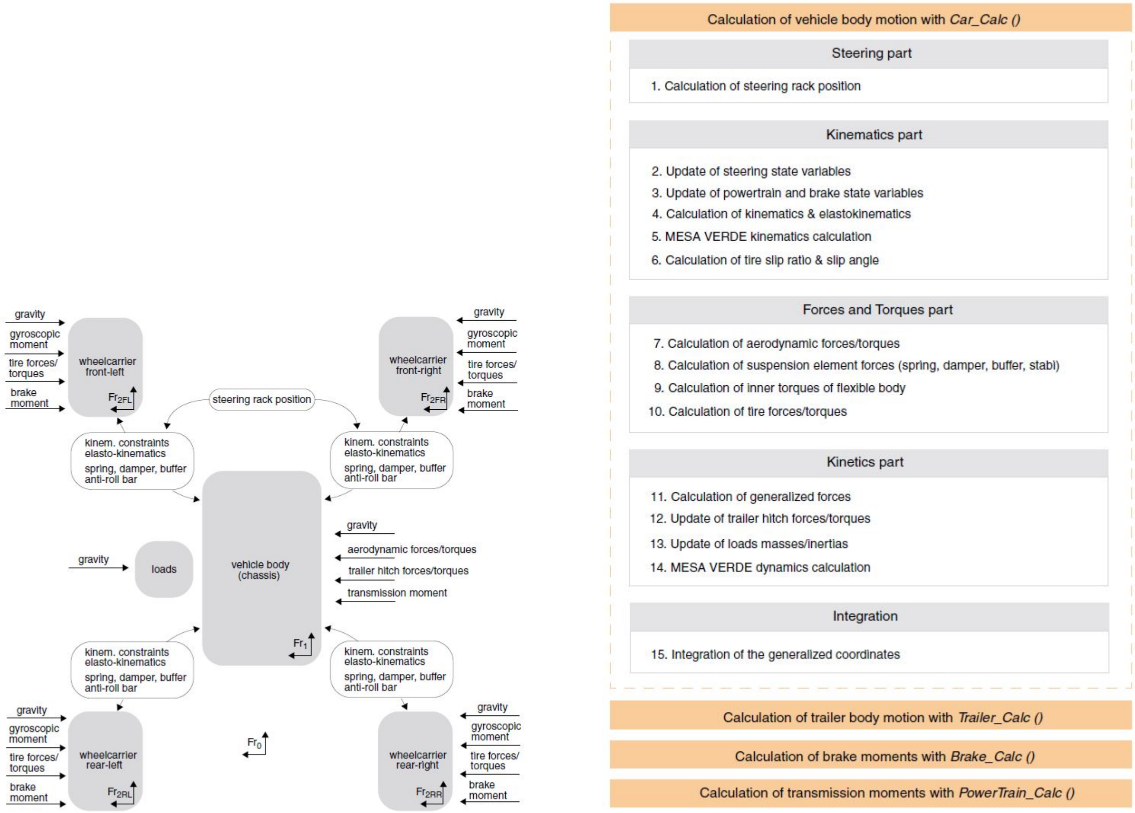


Figure 4.2: IPG CarMaker Virtual Test Environment



Calculation of vehicle body motion with *Car_Calc ()*

Steering part

1. Calculation of steering rack position

Kinematics part

2. Update of steering state variables
3. Update of powertrain and brake state variables
4. Calculation of kinematics & elastokinematics
5. MESA VERDE kinematics calculation
6. Calculation of tire slip ratio & slip angle

Forces and Torques part

7. Calculation of aerodynamic forces/torques
8. Calculation of suspension element forces (spring, damper, buffer, stabi)
9. Calculation of inner torques of flexible body
10. Calculation of tire forces/torques

Kinetics part

11. Calculation of generalized forces
12. Update of trailer hitch forces/torques
13. Update of loads masses/inertias
14. MESA VERDE dynamics calculation

Integration

15. Integration of the generalized coordinates

Calculation of trailer body motion with *Trailer_Calc ()*

Calculation of brake moments with *Brake_Calc ()*

Calculation of transmission moments with *PowerTrain_Calc ()*

(a) Architecture of Forces

(b) Chain of Calculation

Figure 4.3: Vehicle Body Modelling in IPG CarMaker [43]

4.3. Simulation

The controller is designed in MatLab with the help of the Embotech FORCESPRO software package. The generated solver block from FORCESPRO can then be used in Matlab Simulink and run in conjunction with IPG CarMaker as the plant model. The dimensions and parameters of the vehicle can be customized in IPG CarMaker. The test maneuver is also specified in IPG CarMaker. The maneuver is performed at a constant forward velocity of 55 km/h. While it is assumed that the velocity stays constant, the velocity does in fact reduce during the maneuver due to the braking. The simulation vehicle parameters can be seen in table 4.1.

Parameter	Description	Value
m	Vehicle mass	1800 kg
I_{zz}	Moment of inertia about z-axis	3500 kgm^2
L	Wheelbase	3.2 m
l_f	Length from front axle to CoG	1.6 m
l_r	Length from rear axle to CoG	1.6 m
h_{cg}	Vehicle center of gravity height	0.726 m
T	Vehicle track width	1.64 m
R_w	Tire radius	0.16 m
SR	Steering ratio	18.4
$C_{\alpha f}$	Front tire cornering stiffness	1.2E5 N/rad
$C_{\alpha r}$	Rear tire cornering stiffness	1.9E5 N/rad

Table 4.1: Simulation Vehicle Parameters

4.4. Results Analysis

This section will show and discuss the results of the simulations. There are several performance indicators (PI) to assess the controller, namely:

- Peak Yaw Rate - The highest absolute yaw rate recorded during the maneuver. Higher values for yaw rate can be unstable and dangerous for handling.

- Root Mean Square Error (RMSE) Yaw Rate - This describes the severity of the error between the actual yaw rate and the reference yaw rate. It is ideally as close to 0 as possible, meaning the reference is tracked perfectly.

- Peak LTR Initial Steer - The largest LTR recorded during the initial steer. Recall that if the value is 1, the vehicle is rolling over.

- Peak LTR Countersteer - The largest LTR recorded during the countersteer.

- Maximum Deviation LTR - The largest deviation of the LTR from the set LTR threshold.

- Deviation Time LTR - The amount of time the vehicle exceeds the LTR threshold.

- Peak Brake Torque - The largest brake torque applied during the maneuver.

- RMSE Brake Torque - This gives an indication of how much braking was applied during the maneuver. It is ideally as close to 0 as possible, so that as little braking is applied as possible.

4.4.1. PLTR controller settings

An MPC controller is easier to tune compared to a PID controller, but it still has parameters that can varied to affect the performance in various ways. As mentioned in chapter 3, the two most important ones are weights and the LTR threshold. This section is about these two parameters and how they affect the controller's performance.

LTR threshold

Figures 4.4 and 4.5 show that as the LTR threshold gets lower the yaw rate error increases. This is also reflected in the Peak Yaw Rate and the RMSE Yaw Rate, shown in table 4.2. This is to be expected since the linear inequality constraint is more restrictive, causing the tracking objective to suffer as a result. Figure 4.6 shows the LTR during the test maneuver. Interestingly, the Peak LTR during the initial steering maneuver was lower when the LTR threshold is set at 0.75 compared to when the LTR threshold is set at 0.60. As a result, the value for Maximum Deviation LTR and Deviation Time LTR is also significantly higher for the controller with the LTR threshold at 0.60. During the simulation we see that the MPC controller is unable to adhere to the set constraints due to the aggressive maneuver. This causes the MPC controller to output sub-optimal control input sequences. This may explain this behavior, especially because a lower LTR threshold will make it harder for the MPC controller to adhere to the set constraints. Methods such as implementing a soft constraint could help this issue and will be discussed further in chapter 5.

Finally, looking at Figure 4.7 and table 4.2 we see that a lower LTR threshold causes larger control inputs. Additionally, we can also see that the two goals of limiting the LTR and minimizing tracking error can be in opposition. In Figure 4.7a there is a large control input action when the LTR threshold is high whereas the other two are closer to zero. Since the LTR does not reach 0.9 the MPC controller is not hindered by the LTR constraint and outputs brake action to the detriment of the overall LTR. The LTR threshold should be set high enough so that it does not impact normal driving but not too high that it reacts too late to prevent rollover accidents. The figures and PIs show that setting the LTR threshold at 0.75 is a reasonable middle ground.

LTR_{th} PI	0.6	0.75	0.9
Peak Yaw Rate [rad/s]	6.86E-1	6.7E-1	6.2E-1
RMSE Yaw Rate [rad/s]	1.14E-1	6.2E-2	5.3E-2
Peak LTR Initial Steer	0.82	0.79	0.85
Peak LTR Countersteer	-0.83	-0.83	-0.83
Maximum Deviation LTR	0.23	0.08	None
Deviation Time LTR [sec]	0.279	0.114	0
Peak Left Brake Torque [Nm]	702	576	663
Peak Right Brake Torque [Nm]	800	1414	800
RMSE Left Brake Torque [Nm]	179	90	154
RMSE Right Brake Torque [Nm]	133	119	159

Table 4.2: PIs with Varying LTR Thresholds

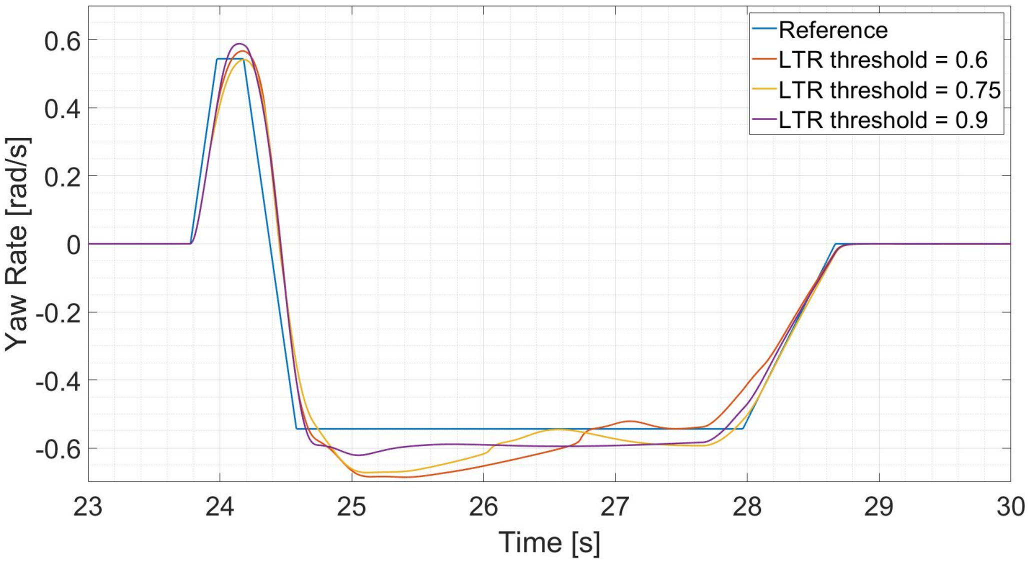


Figure 4.4: Yaw rate with varying LTR thresholds

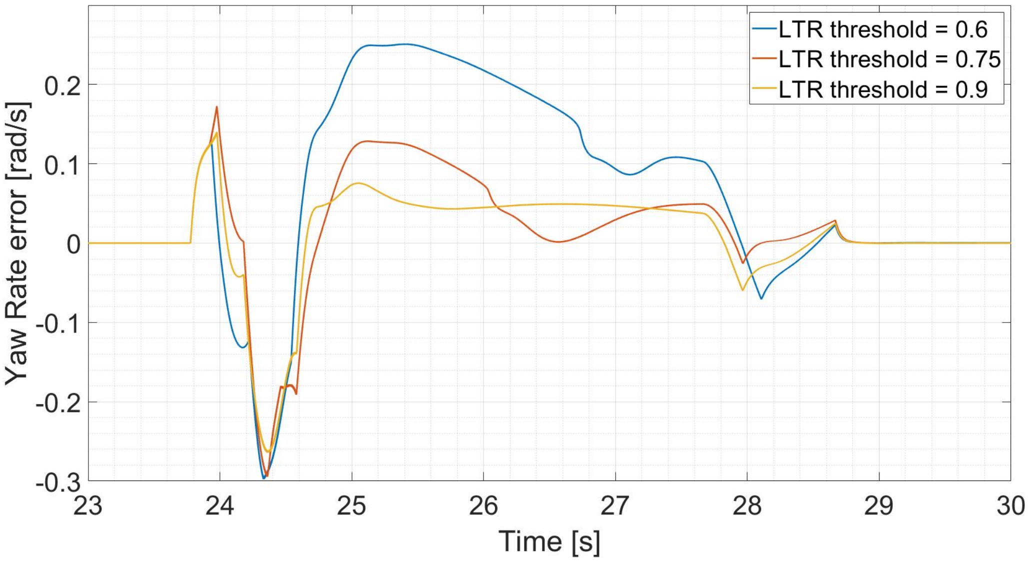


Figure 4.5: Yaw rate error with varying LTR thresholds

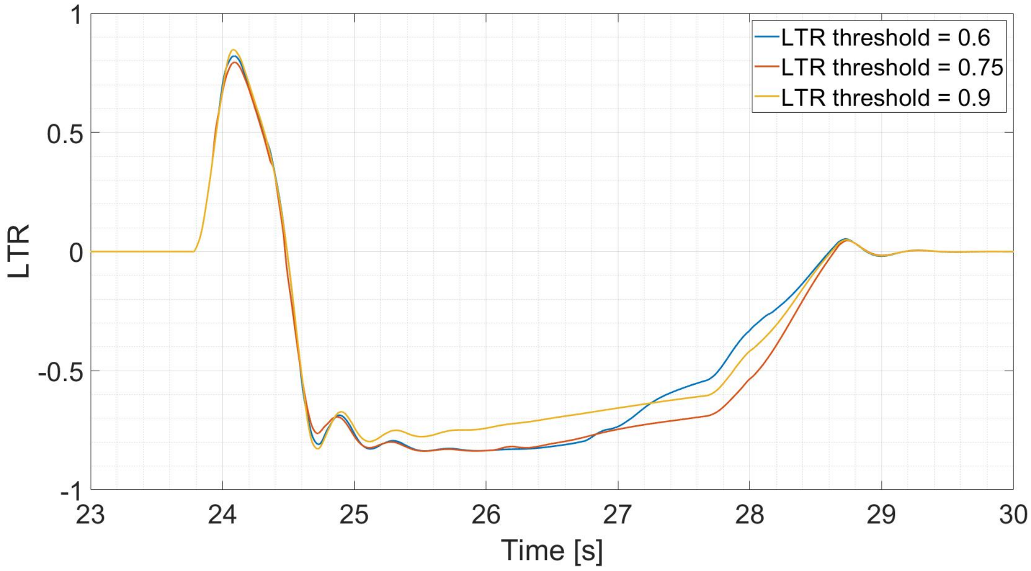
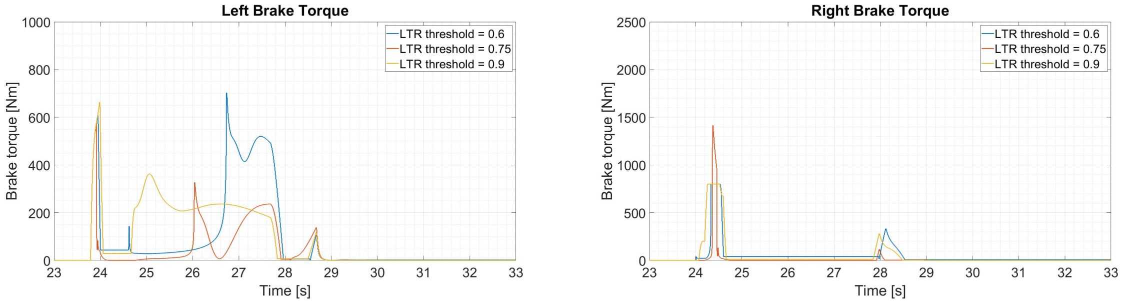


Figure 4.6: LTR with varying LTR thresholds



(a) Left brake torque

(b) Right brake torque

Figure 4.7: Brake torques with varying LTR thresholds

Weight Selection

The weight in the objective function of the MPC controller is adjusted here. Specifically, only the weight that acts on the yaw rate error is adjusted. Figures 4.8 and 4.9 show that as the weight of the objective function is increased the yaw rate error is also decreased, and vice versa. This is also reflected by the Peak Yaw Rate and RMSE Yaw Rate shown in table 4.3. This behavior is expected, since the objective function is attempting to minimize the yaw rate error. However, at higher weights the controller starts to jitter, which can most easily be seen in Figure 4.11. Not only is this behavior not physically possible, it is also undesirable behavior. The Peak Brake Torques also show that at higher weights the control input is constrained by the brake torque boundary conditions, reaching the maximum allowable value. Additionally, it is also clear from the RMSE Brake Torque that higher weights lead to more braking, which is to be expected.

The Peak Yaw Rate during the initial steer increases as the weight increases, also shown in Figure 4.10. The Deviation Time LTR also increases as the weight increases. Curiously, the Peak LTR during the countersteer is lower when then weight is increased. It is possible that in this instance following the reference yaw rate more aggressively is beneficial for the LTR as well. Overall, this demonstrates again that the objective function that aims to minimize yaw rate error can come into conflict with the linear inequality constraint that keeps the LTR under the threshold. These opposing objectives need to be resolved somehow. Possible solutions for this are discussed further in chapter 5. Finally, we choose 6E4 as the value for the weight for its balanced controller behavior.

Weight PI	6E3	6E4	6E5
Peak Yaw Rate [rad/s]	6.79E-1	6.7E-1	6.6E-1
RMSE Yaw Rate [rad/s]	7.1E-2	6.2E-2	4.5E-2
Peak LTR Initial Steer	0.77	0.79	0.84
Peak LTR Countersteer	-0.84	-0.83	-0.80
Maximum Deviation LTR	0.09	0.08	0.09
Deviation Time LTR [sec]	0.081	0.114	0.153
Peak Left Brake Torque [Nm]	85	576	2400
Peak Right Brake Torque [Nm]	147	1414	2400
RMSE Left Brake Torque [Nm]	12	90	311
RMSE Right Brake Torque [Nm]	15	119	424

Table 4.3: PIs with Varying Weights

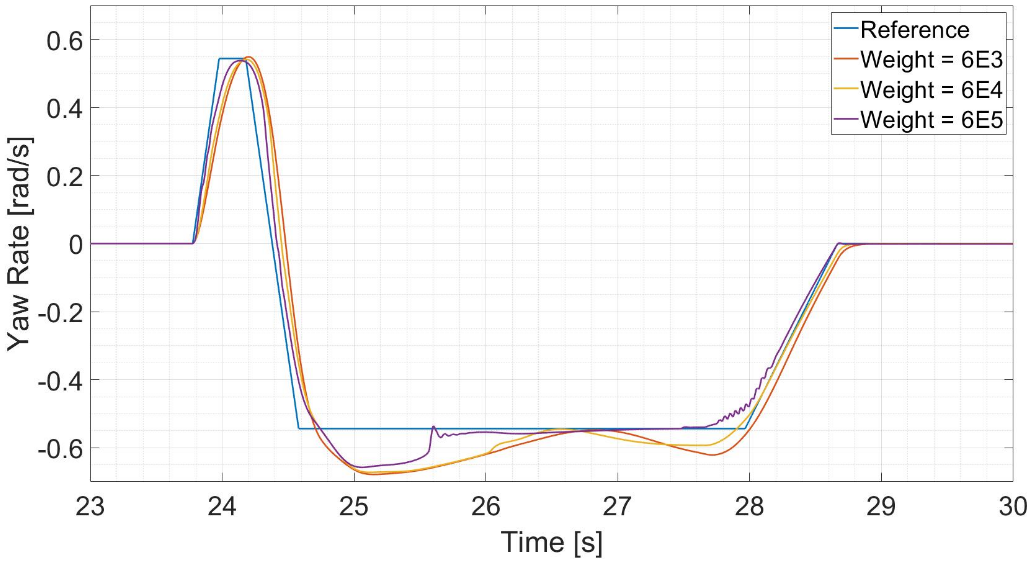


Figure 4.8: Yaw rate with varying weights

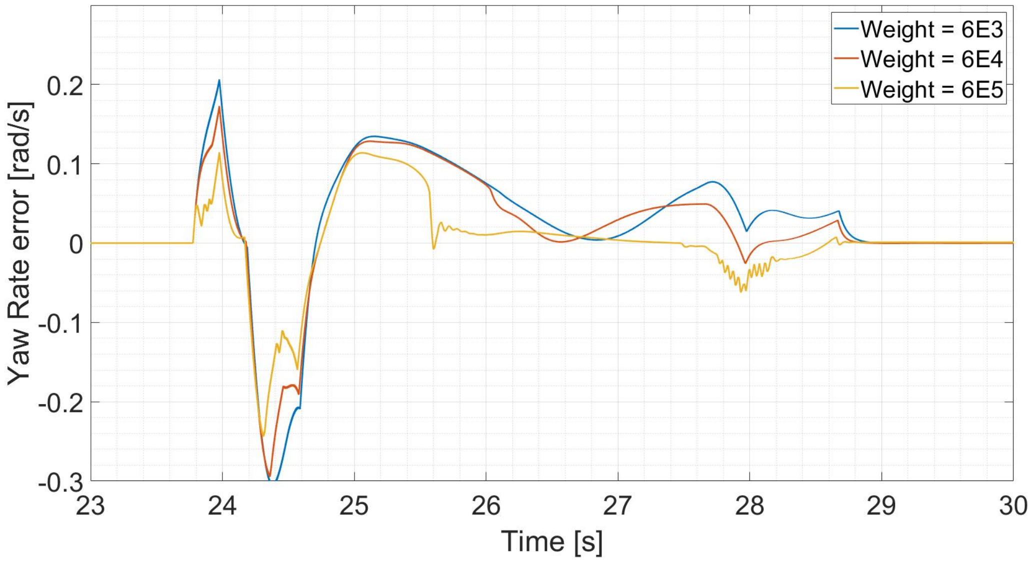


Figure 4.9: Yaw rate error with varying weights

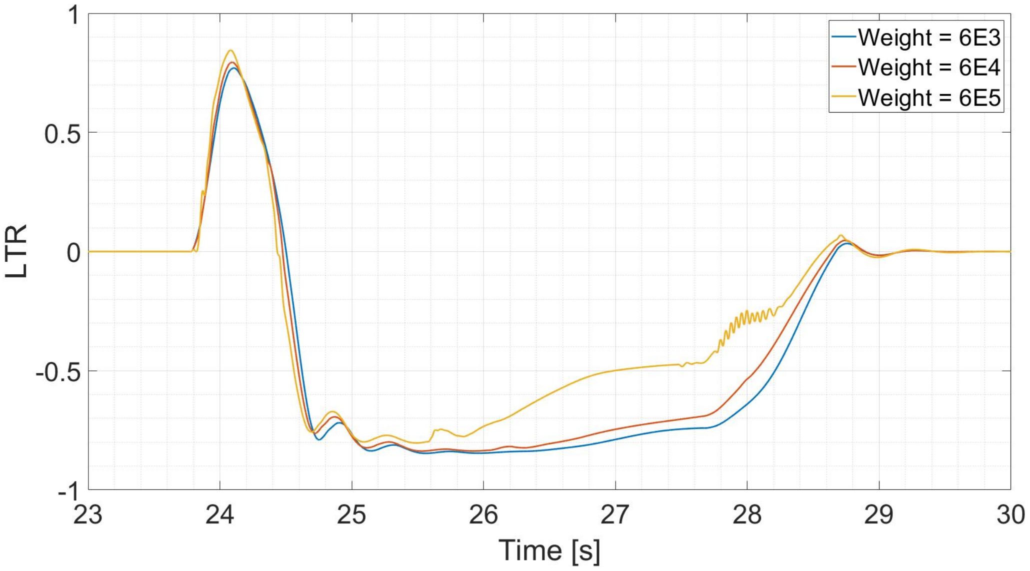
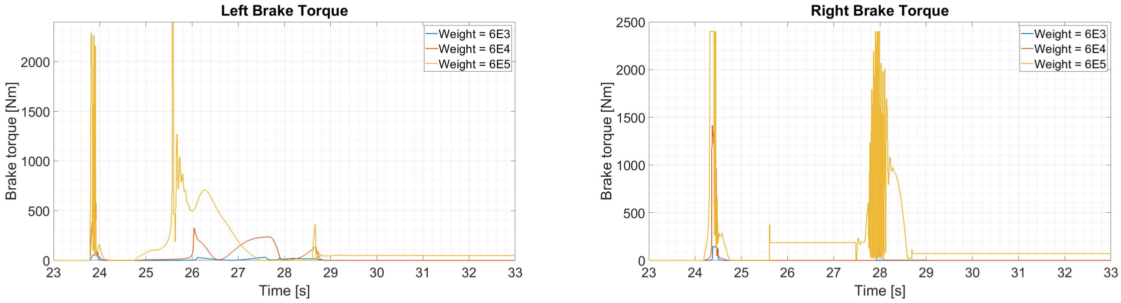


Figure 4.10: LTR with varying weights



(a) Left brake torque

(b) Right brake torque

Figure 4.11: Brake torques with varying weights

4.4.2. PLTR controller vs. LTRs controller

From the Peak Yaw Rate and RMSE Yaw Rate we see that in terms of yaw rate and reference tracking, the performance of the proposed PLTR controller is worse than the benchmark LTRs controller. This is not the main objective of the controller, but is still something that needs to be kept in mind. Considering the objective functions of the two controllers are identical, the difference is caused by the change from PLTR to LTRs. One reason that the benchmark controller performs better in this regard could be due to the fact that the LTRs is much more strongly linked with the yaw rate than the PLTR is, since it uses the lateral acceleration as its sole dynamic variable.

The PLTR controller performs better during the first phase of the maneuver, as shown by the Peak LTR during the initial steer. However, the Peak LTR during the countersteer is larger for the PLTR controller. This behavior is similar to one demonstrated by varying the weights as shown in Figure 4.10. Again, it could be that during this specific instance, following the reference yaw rate more closely is beneficial. Overall, the PLTR controller is better at keeping the vehicle within safe rollover bounds as shown by the Maximum Deviation LTR and the Deviation Time LTR metrics. As mentioned in the previous section, the reference tracking objective can be in opposition to the rollover prevention objective. We see in this case as well, where the PLTR controller is better at rollover prevention but is worse at the yaw rate reference tracking. Finally, the RMSE Brake Torques show that the PLTR controller is more efficient than the LTR controller at the rollover prevention task. By incorporating the PLTR, which has a predictive element, it gives the controller more time to act, thereby making it more efficient.

All in all, the usage of the PLTR in a real-time rollover prevention MPC controller has been shown to be beneficial compared to the simplest LTRs. The PLTR is not much more complex than the LTRs and can therefore by run in real-time. A controller for rollover prevention using PLTR is therefore an enticing option for production vehicles. However, this iteration of the controller could use a few improvements, which will be elaborated further upon in chapter 5.

Controller PI	PLTR	Benchmark	Relative Performance
Peak Yaw Rate [rad/s]	6.7E-1	6.3E-1	-6.9%
RMSE Yaw Rate [rad/s]	6.2E-2	5.3E-2	-18%
Peak LTR Initial Steer	0.79	0.85	6.7%
Peak LTR Countersteer	-0.83	-0.80	-5.4%
Maximum Deviation LTR	0.08	0.10	12%
Deviation Time LTR [sec]	0.114	0.154	35%
Peak Left Brake Torque [Nm]	576	663	15%
Peak Right Brake Torque [Nm]	1414	1187	-19%
RMSE Left Brake Torque [Nm]	90	162	81%
RMSE Right Brake Torque [Nm]	119	192	61%

Table 4.4: PIs with PLTR and Benchmark Controllers

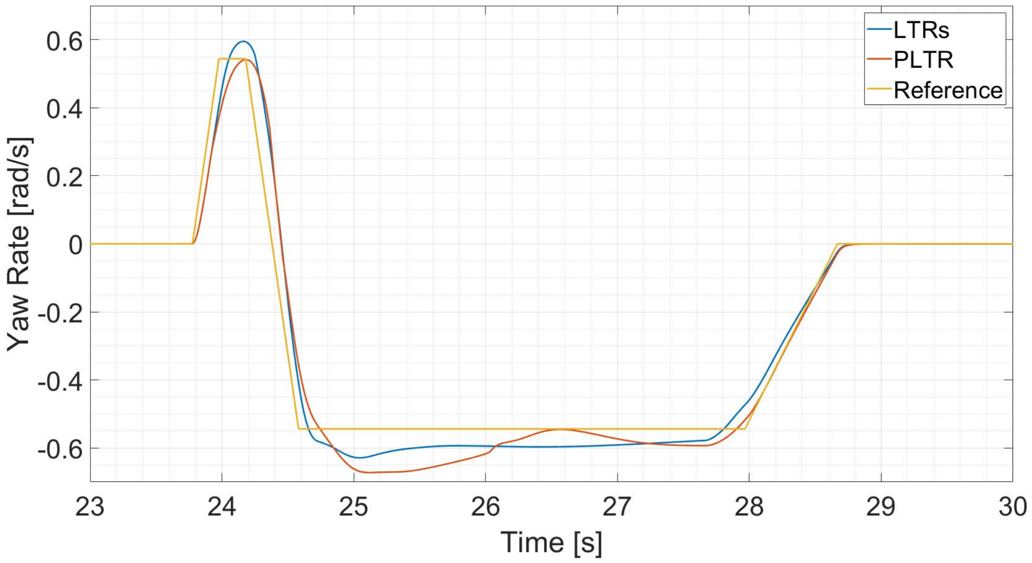


Figure 4.12: Yaw rate with PLTR and LTRs MPC controllers

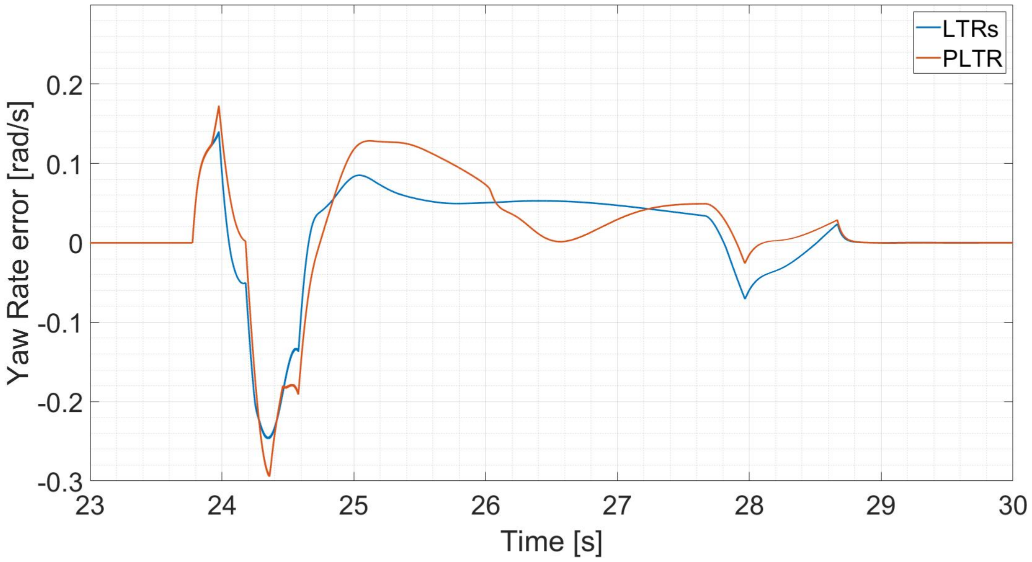


Figure 4.13: Yaw rate error with PLTR and LTRs MPC controllers

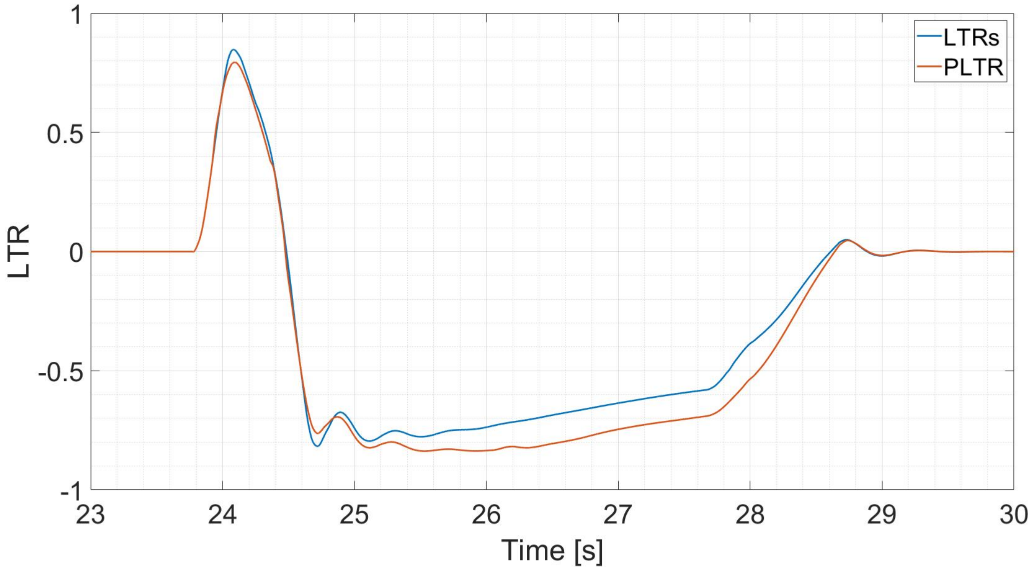
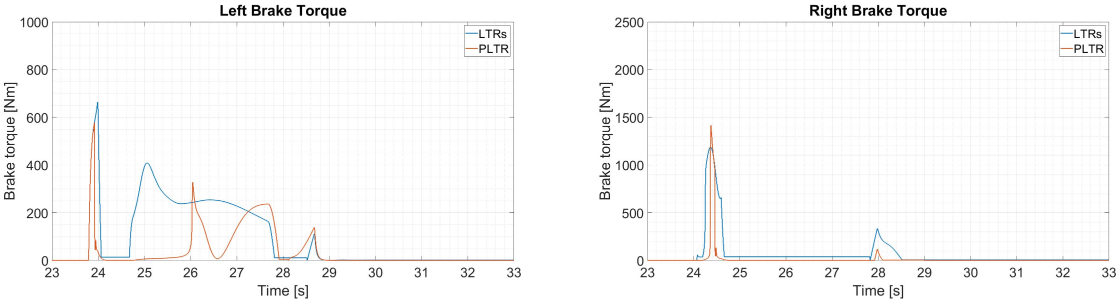


Figure 4.14: LTR with PLTR and LTRs MPC controllers



(a) Left brake torque

(b) Right brake torque

Figure 4.15: Brake torques with PLTR and LTRs MPC controllers

4.5. Summary

In this chapter the testing method and simulation used in this thesis are explained. The Fishhook maneuver, which is designed to induce rollover in vehicles is used to assess the controller's ability to prevent rollover. For the simulation, Simulink and IPG CarMaker are used in a closed loop. The designed controller is implemented in Simulink with the help of FORCESPRO and then is able to interface with IPG CarMaker as the vehicle plant. The obtained results are also shown. We see that the controller is able to help prevent rollover. Compared to the benchmark controller, it is better at the task of rollover prevention, but performs worse at the reference tracking task. Overall, the proposed controller using PLTR is adept at preventing rollover in real-time.

5
Conclusions and Recommendations

Recently, vehicles with a higher center of gravity such as SUVs are becoming more popular. These types of vehicles with a higher center of gravity are more prone to rollover accidents than other passenger cars. Rollover accidents are relatively infrequent compared to most other accidents, but are extremely dangerous and deadly. The prominence of these vehicles highlights a need for a rollover prevention system to prevent these rollover accidents and preserve the passenger's safety.

A review of existing methods for rollover detection and prevention shows that it is a difficult problem to solve. The most common way to detect rollover involves the load transfer ratio (LTR). The LTR is based on the normal forces on the wheels and can therefore not be easily measured and used for controllers. Various ways to estimate and approximate the LTR exist. Other methods for rollover detection are also available, but among all the methods either the lateral acceleration or roll angle is the most important part. Roll angle based methods are more accurate, but the issue lies in obtaining an accurate measure of the roll angle of the vehicle in real-time. On the other hand, lateral acceleration based methods are less accurate but the lateral acceleration is easily obtained. Methods to predict rollover ahead of time also exist, giving a controller more time to act and prevent rollover. One method that stands out is the predictive load transfer ratio (PLTR), which is a simple detection method that is also able to detect rollover up to 100 ms in advance.

Various rollover prevention controllers have been found in the literature. To prevent rollover, one could use differential braking, active steering, active suspensions, or a combination of the three. One commonly used controller structure is the model predictive controller, which offers many advantages compared to other controller structures. From this review, there came an idea to combine a simple yet predictive rollover detection method with an MPC controller for rollover prevention.

In order to design the controller and run tests, the system model is needed. A linear bicycle model for the vehicle is derived and is used. Furthermore, an IPG CarMaker vehicle model is used as the vehicle plant. This allows us to run tests in a simulation with a high degree of accuracy. With the linear bicycle model, an MPC controller is designed to prevent rollover. It follows the driver's desired yaw rate while trying to keep the LTR in a safe operating region based on the PLTR. The controller is implemented and deployed using FORCESPRO. The maneuver used to test the controller efficacy is the Fishhook maneuver designed to induce rollover in vehicles. The results demonstrate that the controller is able to respond effectively and prevent rollover. Compared to a benchmark controller using static LTR instead of the PLTR, it can be observed that the proposed PLTR controller performs better at rollover prevention as the cost of yaw rate reference tracking performance. In conclusion, such a system could be used in production vehicles to increase the safety of vehicles, especially ones with a higher center of gravity.

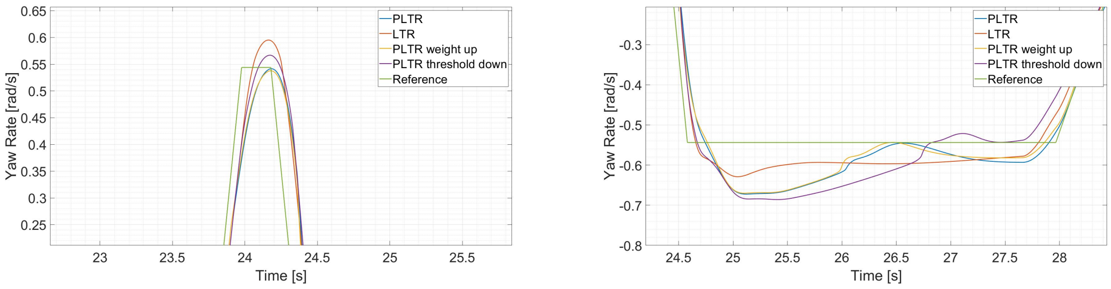
5.1. Recommendations and Future Work

The rollover prevention controller designed in this thesis can be improved upon in several ways:

- The controller will in certain cases be unable to find a feasible solution to be able to satisfy the constraints set on the maximum allowable load transfer ratio due to the aggressive nature of the maneuver. Changing the constraint into a soft constraint could help alleviate this problem.
- The controller has an issue with two opposing objectives. With an aggressive steering maneuver, it will have sacrifice one for the other. If it chooses to try and follow the reference yaw rate based on the driver's steering, it can lead to rollover stability being compromised. This can be fixed by utilizing a controller with stages or dynamic weights, where it will start to prioritize rollover stability as it gets closer to tipping over. In this case, instead of a linear inequality constraint, the LTR should be an additional tracking objective that only activates above a certain LTR threshold.
- The controller currently does not account for actuator dynamics. Introducing a delay in the braking force would make the results more realistic.
- The controller only utilizes differential braking. Combining differential braking with active steering and/or an active suspension could make it operate more efficiently.

A

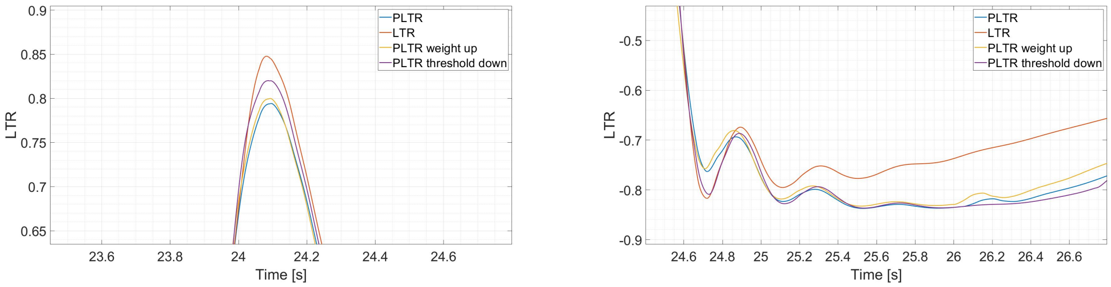
Results with different MPC settings



(a) Initial steer

(b) Countersteer

Figure A.1: Yaw rate



(a) Initial steer

(b) Countersteer

Figure A.2: LTR

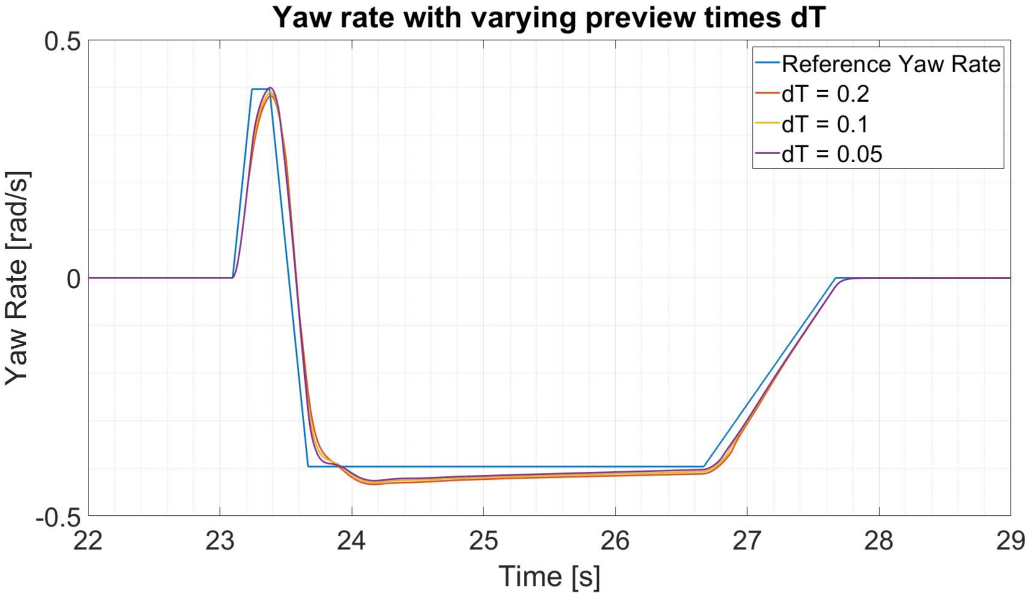
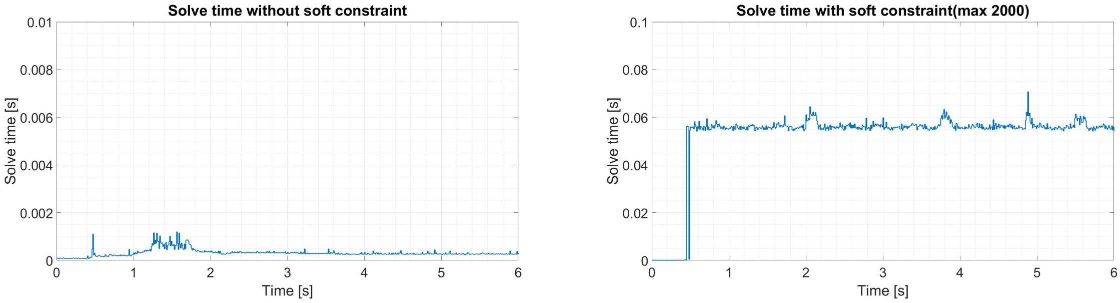


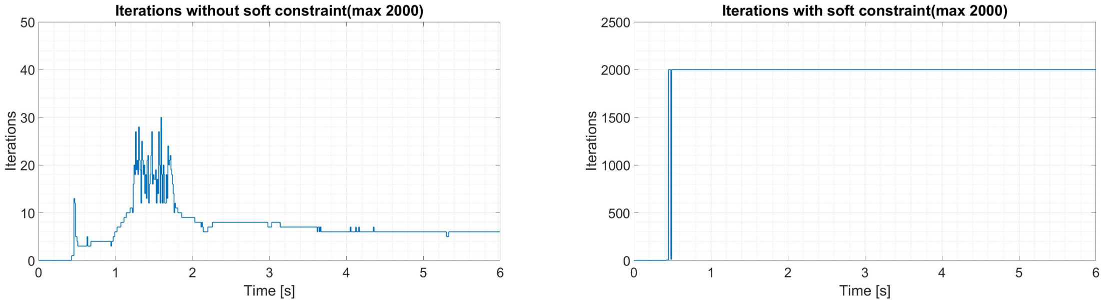
Figure A.3: Varying dT



(a) Without soft constraint

(b) With soft constraint

Figure A.4: Solve time with and without soft constraint



(a) Without soft constraint

(b) With soft constraint

Figure A.5: Iterations with and without soft constraint

Bibliography

[1] A. Strashny, “An analysis of motor vehicle rollover crashes and injury outcomes,” NHTSA, Tech. Rep., 2007.

[2] *Common injuries from rollover crashes: Law offices of jacob emrani*, Aug. 2020. [Online]. Available: `https://www.calljacob.com/common-injuries-from-rollover-accidents-and-crashes/`.

[3] D. Odenthal, T. Bünte, and J. Ackermann, “Nonlinear steering and braking control for vehicle rollover avoidance,” in *1999 European Control Conference (ECC)*, IEEE, 1999, pp. 598–603.

[4] S. Solmaz, M. Corless, and R. Shorten, “A methodology for the design of robust rollover prevention controllers for automotive vehicles: Part 2-active steering,” in *2007 American Control Conference*, IEEE, 2007, pp. 1606–1611.

[5] H. Dahmani, O. Pagès, A. El Hajjaji, and N. Daraoui, “Observer-based robust control of vehicle dynamics for rollover mitigation in critical situations,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, pp. 274–284, 2013.

[6] B.-C. Chen and H. Peng, “Differential-braking-based rollover prevention for sport utility vehicles with human-in-the-loop evaluations,” *Vehicle System Dynamics*, vol. 36, no. 4-5, pp. 359–389, 2001.

[7] H. Yu, L. Güvenc, and Ü. Özgüner, “Heavy duty vehicle rollover detection and active roll control,” *Vehicle system dynamics*, vol. 46, no. 6, pp. 451–470, 2008.

[8] G. Phanomchoeng and R. Rajamani, “New rollover index for the detection of tripped and untripped rollovers,” *IEEE Transactions on Industrial Electronics*, vol. 60, no. 10, pp. 4726–4736, 2012.

[9] S. B. Choi, “Practical vehicle rollover avoidance control using energy method,” *Vehicle System Dynamics*, vol. 46, no. 4, pp. 323–337, 2008.

[10] B. Johansson and M. Gafvert, “Untripped suv rollover detection and prevention,” in *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, IEEE, vol. 5, 2004, pp. 5461–5466.

[11] S.-K. Chen, N. Moshchuk, F. Nardi, and J. Ryu, “Vehicle rollover avoidance,” *IEEE Control Systems Magazine*, vol. 30, no. 4, pp. 70–85, 2010.

[12] J. Yoon and K. Yi, “A rollover mitigation control scheme based on rollover index,” in *2006 American control conference*, IEEE, 2006, 6–pp.

[13] A. Hac, T. Brown, and J. Martens, “Detection of vehicle rollover,” SAE Technical Paper, Tech. Rep., 2004.

[14] C. R. Carlson and J. C. Gerdes, “Optimal rollover prevention with steer by wire and differential braking,” in *ASME 2003 international mechanical engineering congress and exposition*, American Society of Mechanical Engineers Digital Collection, 2003, pp. 345–354.

[15] V. Cherian, R. Shenoy, A. Stothert, J. Shriver, J. Ghidella, and T. D. Gillespie, “Model-based design of a suv anti-rollover control system,” SAE Technical Paper, Tech. Rep., 2008.

[16] V. Trent and M. Greene, “A genetic algorithm predictor for vehicular rollover,” in *IEEE 2002 28th Annual Conference of the Industrial Electronics Society. IECON 02*, IEEE, vol. 3, 2002, pp. 1752–1756.

[17] J. Cao, L. Jing, K. Guo, and F. Yu, “Study on integrated control of vehicle yaw and rollover stability using nonlinear prediction model,” *Mathematical Problems in Engineering*, vol. 2013, 2013.

[18] H. Dahmani, M. Chadli, A. Rabhi, and A. El Hajjaji, “Vehicle dynamic estimation with road bank angle consideration for rollover detection: Theoretical and experimental studies,” *Vehicle System Dynamics*, vol. 51, no. 12, pp. 1853–1871, 2013.

[19] B. Zhu, Q. Piao, J. Zhao, and L. Guo, “Integrated chassis control for vehicle rollover prevention with neural network time-to-rollover warning metrics,” *Advances in Mechanical Engineering*, vol. 8, no. 2, p. 1 687 814 016 632 679, 2016.

[20] E. N. Sanchez, L. J. Ricalde, R. Langari, and D. Shahmirzadi, “Rollover prediction and control in heavy vehicles via recurrent neural networks,” in *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, IEEE, vol. 5, 2004, pp. 5210–5215.

[21] L.-k. Chen and J.-m. Huang, “Predictive rollover index based on frontal road geometry and on-line driver information,” in *2013 IEEE International Conference on Systems, Man, and Cybernetics*, IEEE, 2013, pp. 4128–4133.

[22] C. Larish, D. Piyabongkarn, V. Tsourapas, and R. Rajamani, “A new predictive lateral load transfer ratio for rollover prevention systems,” *IEEE Transactions on Vehicular Technology*, vol. 62, no. 7, pp. 2928–2936, 2013.

[23] S. Solmaz, M. Corless, and R. Shorten, “A methodology for the design of robust rollover prevention controllers for automotive vehicles with active steering,” *International Journal of Control*, vol. 80, no. 11, pp. 1763–1779, 2007.

[24] M. Ataei, A. Khajepour, and S. Jeon, “A general rollover index for tripped and un-tripped rollovers on flat and sloped roads,” *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of automobile engineering*, vol. 233, no. 2, pp. 304–316, 2019.

[25] S. Yim, Y. Park, and K. Yi, “Design of active suspension and electronic stability program for rollover prevention,” *International journal of automotive technology*, vol. 11, no. 2, pp. 147–153, 2010.

[26] K. Shao, J. Zheng, and K. Huang, “Robust active steering control for vehicle rollover prevention,” *International Journal of Modelling, Identification and Control*, vol. 32, no. 1, pp. 70–84, 2019.

[27] J. Ackermann, T. Bünte, and D. Odenthal, “Advantages of active steering for vehicle dynamics control,” 1999.

[28] S. Yim, “Design of a robust controller for rollover prevention with active suspension and differential braking,” *Journal of mechanical science and technology*, vol. 26, no. 1, pp. 213–222, 2012.

[29] A. Riofrio, S. Sanz, M. J. L. Boada, and B. L. Boada, “A lqr-based controller with estimation of road bank for improving vehicle lateral and rollover stability via active suspension,” *Sensors*, vol. 17, no. 10, p. 2318, 2017.

[30] J.-S. Lee, H.-J. Kwon, and C.-Y. Oh, “A study on the effects of active suspension upon vehicle handling,” *Transactions of the Korean Society of Mechanical Engineers A*, vol. 22, no. 3, pp. 603–610, 1998.

[31] V. T. Vu, O. Sename, L. Dugard, and P. Gáspár, “Enhancing roll stability of heavy vehicle by lqr active anti-roll bar control using electronic servo-valve hydraulic actuators,” *Vehicle system dynamics*, vol. 55, no. 9, pp. 1405–1429, 2017.

[32] L. Li, Y. Lu, R. Wang, and J. Chen, “A three-dimensional dynamics control framework of vehicle lateral stability and rollover prevention via active braking with mpc,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 4, pp. 3389–3401, 2016.

[33] *Road vehicles — vehicle dynamics and road-holding ability — vocabulary.* [Online]. Available: `https://www.iso.org/obp/ui/#iso:std:iso:8855:ed-2:v1:en`.

[34] M. Kissai, B. Monsuez, X. Mouton, D. Martinez, and A. Tapus, “Adaptive robust vehicle motion control for future over-actuated vehicles,” *Machines*, vol. 7, no. 2, p. 26, 2019.

[35] H. Dugoff, P. S. Fancher, and L. Segel, “Tire performance characteristics affecting vehicle response to steering and braking control inputs,” Tech. Rep., 1969.

[36] H. B. Pacejka and E. Bakker, “The magic formula tyre model,” *Vehicle system dynamics*, vol. 21, no. S1, pp. 1–18, 1992.

[37] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: theory, computation, and design.* Nob Hill Publishing Madison, WI, 2017, vol. 2.

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

[39] R. Fletcher, *Practical methods of optimization.* John Wiley & Sons, 2013.

[40] F. Yakub, S. Lee, and Y. Mori, “Comparative study of mpc and lqc with disturbance rejection control for heavy vehicle rollover prevention in an inclement environment,” *Journal of Mechanical Science and Technology*, vol. 30, no. 8, pp. 3835–3845, 2016.

[41] G. J. Forkenbrock, B. C. O'Harra, and D. Elsasser, "A demonstration of the dynamic tests developed for nhtsa's ncap rollover rating system - phase viii of nhtsa's light vehicle rollover research program," NHTSA, Tech. Rep., 2004.

[42] J. Yoon, W. Cho, J. Kang, B. Koo, and K. Yi, "Design and evaluation of a unified chassis control system for rollover prevention and vehicle stability improvement on a virtual test track," *Control Engineering Practice*, vol. 18, no. 6, pp. 585–597, 2010.

[43] *Carmaker reference manual*. [Online]. Available: `https://ipg-automotive.com/en/products-solutions/software/carmaker/`.