Lateral Control of Heavy Vehicles

Aravind Jawahar
Lokesh Palla

Master of Science in Engineering
Master program in Vehicle Engineering
KTH Royal Institute of Technology

Supervisors at Volvo GTT: Markus Gerdin, David Öst, Erik Börve
Supervisor at KTH: Mikael Nybacka
Examiner at KTH: Mikael Nybacka

*Date of Presentation: 04/09/2023

*TRITA-SCI-GRU 2023:289

KTH Royal Institute of Technology
School of Engineering Sciences
KTH SCI SE-100 44 Stockholm, Sweden
Website: [School of Engineering Sciences](http://www.kth.se)
Authors
Aravind Jawahar <jawahar@kth.se>
Lokesh Palla <lpalla@kth.se>
Department of Vehicle Engineering
KTH Royal Institute of Technology

Place for Project
Stockholm, Sweden

Examiner
Mikael Nybacka
Department of Vehicle Engineering
KTH Royal Institute of Technology
Stockholm, Sweden

Supervisors
Markus Gerdin, David Öst, Erik Börve
Volvo Group Truck Technology
Gothenburg, Sweden
Abstract

The automotive industry has been involved in making vehicles autonomous to various levels in the past decade. Particularly in the commercial vehicle market, there is a significant need to equip trucks with a certain level of automation to reduce dependence on human efforts for driving. This could help reduce accidents caused by human error. Interestingly, there are several challenges and solutions in achieving and implementing autonomous driving for trucks. First, a comparison of different control architectures that can enable a truck to drive autonomously is explored. The selected controllers (Pure Pursuit, Stanley, Linear Quadratic Regulator, Sliding Mode Control, and Model Predictive Control) vary in simplicity of implementation and versatility in handling various vehicle parameters and constraints. A thorough comparison of these path-tracking controllers is conducted using several metrics. Second, a collision avoidance system based on cubic polynomials, inspired by rapidly exploring random trees (RRT), is presented. Some of the path-tracking controllers have limitations, necessitating a standalone collision avoidance system to ensure safe maneuvering. Simulations are performed for different test cases with and without obstacles. These simulations help compare the safety margin and driving comfort of each path-tracking controller integrated with the collision avoidance system. Third, various performance metrics such as changes in acceleration input, changes in steering input, path tracking errors, deviations from the base frame of the track file, and lateral and longitudinal margins between the ego vehicle and the target vehicle are presented. In conclusion, a set of suitable controllers for heavy articulated vehicles is developed and benchmarked.

Keywords

Path Tracking, Collision Avoidance, Pure Pursuit, Stanley, Linear Quadratic Regulator, Sliding Mode Control, Model Predictive Control
Sammanfattning


Nyckelord

Path Tracking, Collision Avoiding, Pure Pursuit, Stanley, Linear Quadratic Regulator, Sliding Mode Control, Model Predictive Control
Acknowledgement

We take this opportunity to express our heartfelt gratitude to all those who have contributed to the successful completion of our master's thesis. This endeavor would not have been possible without the unwavering support, guidance, and encouragement from various individuals and institutions.

First and foremost, we extend our sincere appreciation to the esteemed professors at KTH Royal Institute of Technology. Their dedicated teaching and mentorship provided us with the necessary knowledge and skills that formed the bedrock of our thesis work. We are indebted to them for imparting invaluable insights that have significantly shaped our academic journey.

Our deepest gratitude goes to our supervisor at KTH Royal Institute of Technology, Mikael Nybacka. His consistent guidance, insightful feedback, and unwavering belief in our capabilities propelled us forward, instilling in us the confidence to tackle challenges and explore new horizons. His constant support has been the driving force behind the success of this project.

We would also like to extend our thanks to our supervisors at Volvo GTT, Markus Gerdin, David Öst and Erik Börve. Their trust in our abilities and their expert guidance have been instrumental in steering our project in the right direction. Their collective knowledge and professional insight enriched our work, enabling us to deliver results of the highest caliber.

We acknowledge with deep appreciation the invaluable resources, facilities, and freedom granted to us by both KTH Royal Institute of Technology and Volvo GTT. These institutions provided us with an enabling environment that fostered creativity and innovation, allowing us to explore and experiment with our ideas freely.

Last but not least, we express our heartfelt gratitude to our families and friends. Their unwavering support, understanding, and encouragement served as our anchor during the ups and downs of this journey. Their belief in our abilities and their constant presence motivated us to overcome challenges and reach this milestone.

In conclusion, we extend our sincere gratitude to all those who played a part in shaping our master's thesis. Your contributions, whether big or small, have left an indelible mark on our academic and personal growth. Thank you for being an integral part of this journey.
List of Abbreviations

PPC  Pure Pursuit Controller
LQR  Linear Quadratic Regulator
SMC  Sliding Mode Control
MPC  Model Predictive Control
KBM  Kinematic Bicycle Model
DBM  Dynamic Bicycle Model
VTM  Volvo Transport Model
CAS  Collision Avoidance System
## Nomenclature

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X$</td>
<td>Global X coordinate</td>
</tr>
<tr>
<td>$Y$</td>
<td>Global Y coordinate</td>
</tr>
<tr>
<td>$V_G$</td>
<td>Longitudinal velocity in global frame at centre of gravity</td>
</tr>
<tr>
<td>$\dot{x}$</td>
<td>Longitudinal velocity in body frame at centre of gravity</td>
</tr>
<tr>
<td>$\dot{y}$</td>
<td>Lateral velocity in body frame at centre of gravity</td>
</tr>
<tr>
<td>$\psi$</td>
<td>Orientation or heading angle</td>
</tr>
<tr>
<td>$\dot{\psi}$</td>
<td>Rate of change of orientation or heading angle</td>
</tr>
<tr>
<td>$\delta$</td>
<td>Steering angle</td>
</tr>
<tr>
<td>$a$</td>
<td>Acceleration</td>
</tr>
<tr>
<td>$\Delta \delta$</td>
<td>Change in steering angle</td>
</tr>
<tr>
<td>$\Delta a$</td>
<td>Change in acceleration</td>
</tr>
<tr>
<td>$e$</td>
<td>Tracking error</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Slip angle on wheel</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Body side-slip angle</td>
</tr>
<tr>
<td>$k$</td>
<td>Curvature</td>
</tr>
<tr>
<td>$T_s$</td>
<td>Sampling time</td>
</tr>
<tr>
<td>$l_f$</td>
<td>Length between front axle and centre of gravity</td>
</tr>
<tr>
<td>$l_r$</td>
<td>Length between rear axle and centre of gravity</td>
</tr>
<tr>
<td>$L$</td>
<td>Length between front axle and rear axle</td>
</tr>
<tr>
<td>$F_{yf}$</td>
<td>Tire force in lateral direction at front axle</td>
</tr>
<tr>
<td>$F_{yr}$</td>
<td>Tire force in lateral direction at rear axle</td>
</tr>
<tr>
<td>$C_f$</td>
<td>Cornering stiffness of front axle</td>
</tr>
<tr>
<td>$C_r$</td>
<td>Cornering stiffness of rear axle</td>
</tr>
<tr>
<td>$m$</td>
<td>Mass of vehicle</td>
</tr>
<tr>
<td>$I_{zz}$</td>
<td>Moment of inertia in vertical axis of vehicle</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Density of air</td>
</tr>
<tr>
<td>$C_d$</td>
<td>Drag coefficient</td>
</tr>
<tr>
<td>$A_r$</td>
<td>Frontal area</td>
</tr>
<tr>
<td>$f_r$</td>
<td>Rolling resistance coefficient</td>
</tr>
<tr>
<td>$\theta$</td>
<td>Road gradient</td>
</tr>
<tr>
<td>$A \ B \ C$</td>
<td>Continuous state space matrices</td>
</tr>
<tr>
<td>$A_d \ B_d \ C_d$</td>
<td>Discrete state space matrices</td>
</tr>
<tr>
<td>$A \hat{B} \hat{C}$</td>
<td>Augmented state space matrices</td>
</tr>
<tr>
<td>$T$</td>
<td>Total simulation time</td>
</tr>
</tbody>
</table>
List of Figures

1.1 Control architecture implemented in the proposed work .................................. 4
4.1 Expected path tracking behavior with collision avoidance system .................. 20
4.2 Geometrical Representation of Vehicle Path Tracking Figures [59] .................. 22
4.3 Stanley Controller setup for a bicycle model [60] ......................................... 24
4.4 Kinematics of lateral vehicle motion [62] .................................................... 34
4.5 Dynamics of lateral vehicle motion [62] ....................................................... 36
5.1 Volvo Transport Model visualization interface .............................................. 43
6.1 PPC tracking performance at 25 km/h ....................................................... 48
6.2 PPC performance tracking metrics at 25 km/h ........................................... 48
6.3 Stanley controller tracking performance at 25 km/h .................................. 49
6.4 Stanley controller performance tracking metrics at 25 km/h ....................... 49
6.5 LQR tracking performance at 25 km/h ...................................................... 50
6.6 LQR performance tracking metrics at 25 km/h .......................................... 50
6.7 SMC tracking performance at 25 km/h ..................................................... 51
6.8 SMC performance tracking metrics at 25 km/h .......................................... 51
6.9 Pure Pursuit lateral control of a truck with collision avoidance system .......... 53
   at 35 km/h ......................................................................................... 53
6.10 Performance evaluation of a Pure Pursuit lateral control for a truck ............... 53
   with collision avoidance system at 35 km/h ............................................ 53
6.11 Stanley lateral control of a truck with collision avoidance system at .......... 54
   35 km/h ......................................................................................... 54
6.12 Performance evaluation of a Stanley lateral control for a truck with .......... 55
   collision avoidance system at 35 km/h .................................................. 55
6.13 LQR lateral control of a truck with collision avoidance system at .............. 56
   35 km/h ......................................................................................... 56
6.14 Performance evaluation of a LQR lateral control for a truck with .......... 56
   collision avoidance system at 35 km/h .................................................. 56
6.15 SMC lateral control of a truck with collision avoidance system at .............. 57
   35 km/h ......................................................................................... 57
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.16</td>
<td>Performance evaluation of a Sliding Mode lateral control for a truck with collision avoidance system at 35 $km/h$</td>
<td>58</td>
</tr>
<tr>
<td>6.17</td>
<td>Pure Pursuit lateral control of a truck with collision avoidance system at 90 $km/h$</td>
<td>59</td>
</tr>
<tr>
<td>6.18</td>
<td>Performance evaluation of a Pure Pursuit lateral control for a truck with collision avoidance system at 90 $km/h$</td>
<td>60</td>
</tr>
<tr>
<td>6.19</td>
<td>Stanley lateral control of a truck with collision avoidance system at 90 $km/h$</td>
<td>61</td>
</tr>
<tr>
<td>6.20</td>
<td>Performance evaluation of a Stanley lateral control for a truck with collision avoidance system at 90 $km/h$</td>
<td>61</td>
</tr>
<tr>
<td>6.21</td>
<td>LQR lateral control of a truck with collision avoidance system at 90 $km/h$</td>
<td>62</td>
</tr>
<tr>
<td>6.22</td>
<td>Performance evaluation of a LQR lateral control for a truck with collision avoidance system at 90 $km/h$</td>
<td>63</td>
</tr>
<tr>
<td>6.23</td>
<td>SMC lateral control of a truck with collision avoidance system at 90 $km/h$</td>
<td>64</td>
</tr>
<tr>
<td>6.24</td>
<td>Performance evaluation of a Sliding Mode lateral control for a truck with collision avoidance system at 90 $km/h$</td>
<td>64</td>
</tr>
<tr>
<td>6.25</td>
<td>Path tracking by MPC-Kinematic integrated with VTM</td>
<td>66</td>
</tr>
<tr>
<td>6.26</td>
<td>Error in path tracking by MPC-Kinematic integrated with VTM</td>
<td>67</td>
</tr>
<tr>
<td>6.27</td>
<td>Input by MPC-Kinematic integrated with VTM while path tracking</td>
<td>67</td>
</tr>
<tr>
<td>6.28</td>
<td>Path tracking by MPC-Dynamic integrated with VTM</td>
<td>68</td>
</tr>
<tr>
<td>6.29</td>
<td>Error in path tracking by MPC-Dynamic integrated with VTM</td>
<td>69</td>
</tr>
<tr>
<td>6.30</td>
<td>Input by MPC-Dynamic integrated with VTM while path tracking</td>
<td>69</td>
</tr>
<tr>
<td>6.31</td>
<td>Path tracking by MPC-Kinematic with kinematic bicycle model as plant</td>
<td>70</td>
</tr>
<tr>
<td>6.32</td>
<td>Error in path tracking by MPC-Kinematic with kinematic bicycle model as the plant</td>
<td>71</td>
</tr>
<tr>
<td>6.33</td>
<td>Input by MPC-Kinematic while path tracking with kinematic bicycle model as the plant</td>
<td>71</td>
</tr>
<tr>
<td>6.34</td>
<td>Path tracking by MPC-Dynamic with dynamic bicycle model as the plant</td>
<td>72</td>
</tr>
<tr>
<td>6.35</td>
<td>Error in path tracking by MPC-Dynamic with dynamic bicycle model as the plant</td>
<td>72</td>
</tr>
<tr>
<td>6.36</td>
<td>Input by MPC-Dynamic while path tracking with dynamic bicycle model as the plant</td>
<td>73</td>
</tr>
<tr>
<td>6.37</td>
<td>Path tracking by MPC-Kinematic integrated with CAS with kinematic bicycle model as the plant</td>
<td>74</td>
</tr>
<tr>
<td>6.38</td>
<td>Deviation in path tracking by MPC-Kinematic integrated with CAS with kinematic bicycle model as the plant</td>
<td>75</td>
</tr>
<tr>
<td>6.39</td>
<td>Input by MPC-Kinematic integrated with CAS while path tracking with kinematic bicycle model as the plant</td>
<td>75</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>6.40</td>
<td>Path tracking by MPC-Dynamic integrated with CAS with dynamic bicycle model as the plant.</td>
<td></td>
</tr>
<tr>
<td>6.41</td>
<td>Deviation in path tracking by MPC-Dynamic integrated with CAS with dynamic bicycle model as the plant.</td>
<td></td>
</tr>
<tr>
<td>6.42</td>
<td>Input by MPC-Dynamic integrated with CAS while path tracking with dynamic bicycle model as the plant.</td>
<td></td>
</tr>
</tbody>
</table>
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>Controllers performance metrics evaluation in short curved road at 25 km/h</td>
<td>52</td>
</tr>
<tr>
<td>6.2</td>
<td>Overall controller performance metrics evaluation in a curved road with CAS at 35 km/h with a static obstacle</td>
<td>59</td>
</tr>
<tr>
<td>6.3</td>
<td>Overall controllers performance metrics evaluation in a curved road with CAS at 90 km/h with a static obstacle</td>
<td>65</td>
</tr>
<tr>
<td>6.4</td>
<td>MPC Controllers performance metrics evaluation in short curved road at 25 km/h</td>
<td>78</td>
</tr>
<tr>
<td>6.5</td>
<td>MPC Controllers with CAS performance metrics evaluation in the short curved road at 25 km/h with a static obstacle at 30m from entry</td>
<td>79</td>
</tr>
<tr>
<td>7.1</td>
<td>Comparison of a list of the control strategies that were examined in the current thesis project</td>
<td>82</td>
</tr>
<tr>
<td>7.2</td>
<td>Summary of a list of the control strategies that were examined in the current thesis project</td>
<td>82</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

In the contemporary era, there is a growing focus from both academia and industry on the topic of lateral control within autonomous vehicle technology. The historical development of control systems for driving has predominantly centered on the dynamics of movement on smooth and even road surfaces, resulting in the creation of various cutting-edge systems. However, with the emergence of autonomous vehicles, the terrain becomes more challenging, demanding a control system capable of handling highways while continuously tracking paths amidst closely spaced hazards. Furthermore, the vehicle must possess the capability to recover from significant disruptions autonomously. Snider et al. proposed several solutions to address these path-tracking challenges [1]. Their approach to tracking a vehicle’s trajectory focuses solely on the front wheel locations relative to the desired trajectory, omitting the vehicle’s body from the equation. This enables collocated system control, with globally asymptotically stable kinematic equations of motion or control laws. Subsequently, they enhanced their approach to control the dynamics of pneumatic tires and servo-actuated steering wheels. By employing appropriate steering actions to guide the vehicle along the path, it can effectively follow a globally defined geometric path. A path-tracking controller is designed to maintain stability while minimizing the lateral distance between the vehicle and the defined path, the difference in heading between the vehicle and the defined path, and the amount of steering input. Geometric path tracking techniques, widely utilized in robotics, exploit geometric relationships between the path and the vehicle, resulting in control law solutions for path tracking. These methods, ranging from straightforward calculations using circular arcs to more intricate calculations involving screw theory, often employ a look-ahead distance to assess errors ahead of the vehicle. This thesis delves into a comparative study of various path-tracking controllers and provides additional support to ensure the safe and comfortable operation of autonomous trucks.
1.1 Background

The development of Advanced Driver Assistance Systems (ADAS) has evolved over several decades, starting in the late 20th century. Significant progress was made between the 1970s and 2000s in ADAS. These included Anti-lock Braking Systems (ABS) and Electronic Stability Control (ESC), Traction control systems, Adaptive Cruise Control (ACC), Lane Departure Warning (LDW), Forward Collision Warning (FCW), and Blind Spot Detection (BSD), etc. in chronological order.

It was in the early 2000s when sensor and computation capability had grown to a large extent that it motivated researchers to explore control algorithms for autonomous driving. The DARPA Grand Challenge in 2004 proved to be an important occurrence in the history of the development of autonomous driving. This competition required participants to autonomously drive through an off-road track for 240 km. Following the success of this event, the DARPA Urban Challenge was conducted in 2007 where the participating autonomous vehicles had to complete an urban track, following traffic regulations and handling obstacle avoidance.

Since then, improvements in autonomous driving have gained pace. Though the research in this field of study has become extensive, at the industry level, still a PID controller is used for path tracking. Though it is computationally feasible to implement, it has several disadvantages like performance being susceptible to gain tuning, instability, oscillation while tracking, sharp steering input, error accumulation with disturbances or changes in external factors like varying behavior of trucks, changes in environment, etc. For several such disadvantages, it is necessary to explore several control architectures and evaluate their performance and feasibility. In this thesis, we present some of the most successful control solutions made so far in its foundational form for making a vehicle autonomously drive.

1.2 Problem Statement

In a truck, there are greater challenges to control its dynamics than passenger cars due to its physical characteristics. Lateral dynamics and roll dynamics pose the majority of the safety-critical concerns in a truck. With the addition of trailers, the vehicle dynamics of the truck become very complex.

The aim of the thesis is to develop lateral controllers for heavy vehicles that can provide accurate path tracking with safe and comfortable driving. This thesis tries to achieve the following objectives:

- Develop different controllers for normal driving and safe positioning of autonomous trucks
- Tackle the problem of slow and changing lateral dynamics of trucks with trailer combinations
- Perform benchmark among the chosen controllers and evaluate the suitable ones through simulations

1.3 Related Work

Several comparisons between different controllers for path tracking have been studied earlier. These are usually done for passenger cars with geometric controllers and model-based controllers using kinematic and dynamic bicycle models \[5\] \[6\] \[7\].

The book Vehicle Dynamics and Control provides a clear understanding of different vehicle models and control actions required for each of them \[8\]. The complexity of a truck dynamics manifolds with trailer combinations, hence making it essential for us to understand its impact while controlling. Higher-order models lead to excessive computation burden. A comparative study of reduced order lateral dynamic models of trailers is covered in \[9\].

An extensive work on modeling and control of articulated heavy vehicles for directional performance is covered in \[10\]. Path planning is a crucial step in ensuring the safety and stability of trucks while performing obstacle avoidance. Several factors like lateral stability goal, phase plane stability boundary, wheel anti-skid goal, and roll-over prevention goal were considered while performing path planning for collision avoidance in the work \[11\].

1.4 System Overview

As shown in Figure \[1.1\], the flow of events in the proposed work is as follows. First, a track file is chosen which represents the route between the start position and destination position. Then a collision avoidance system chooses a trajectory that is obstacle-free and optimal for path tracking. Along with the trajectory, a velocity profile is also generated. This is fed as input to the path-tracking controllers. Different path-tracking controllers take different inputs and are capable of giving different outputs based on their nature. The control inputs are implemented in a plant called VTM (Volvo Transport Model). The states are estimated and fed back into the path planning system and path tracking controllers to update their states. For every such time step, the process repeats ensuring that the truck progresses while maintaining a safe distance from obstacles and safe velocity for stability. Some controllers like Pure Pursuit and Stanley produce desired steering angles as control inputs, here for longitudinal control, a PID controller is used to achieve velocity tracking. Some controllers like Model Predictive Control produce integrated lateral...
and longitudinal control by giving acceleration and steering inputs. The Volvo Transport Model is a Simulink-based model of trucks developed by Volvo. It is a complex nonlinear vehicle model that represents truck dynamics to a large extent. Different truck combinations can be initiated and the model can assemble accordingly. A 6x4 rigid truck model, which describes a truck with six 'wheel ends' of which four are driven through rear axles was used to develop the complete system. Later articulated models were tested with the same control systems to check their effectiveness.

![Diagram](image.png)

Figure 1.1: Control architecture implemented in the proposed work.

The tracks chosen for the analysis are a curved road track of short distance at low velocity and a curved road track of long distance at high velocity. These tracks replicate sharp curvatures in semi-urban driving scenarios and graduate curvatures in highway driving. With an obstacle being detected and if only braking behind the obstacle is not a viable option, a lane change maneuver is performed. Once the obstacle is avoided, the truck tries to come back to its base frame (the rightmost lane), thus mimicking a double lane change.
Chapter 2

Literature Review

Striking a balance between performance and feasibility is the need of the hour in the autonomous vehicle industry. With growing interest in niche control strategies, it is often neglected to see how feasible the solution is for implementation in real-time systems. A study of different controllers suitable for autonomous trucks is discussed in the following sections. Based on several comparative studies, the following controllers are chosen:

- Pure Pursuit
- Stanley
- Linear Quadratic Regulator
- Sliding Mode Control
- Model Predictive Control

Some of the above-mentioned controllers are limited by their capacity. For example, the Pure Pursuit controller is a geometry-based reactive controller. The purpose of this controller is to follow global reference coordinates X and Y. The output given by the controller is the desired steering angle. Neither vehicle dynamics nor obstacles and constraints can be given consideration by this controller. To tackle such problems and still achieve safe driving and normal positioning, a collision avoidance system is developed that acts as a safe path planner for all the controllers.

Autonomous driving is not only about path tracking but also about path planning. It is of crucial importance that the path being planned considers three significant aspects of the environment. First, the path planned should ensure that the route specified is followed from end to end. Two, it should respect lane and road boundaries. Three, it should be able to negotiate with obstacles in all directions. Other factors like look ahead distance, velocity profile, etc. can be designed in accordance
with the application. There are several path planning methods that ensure a safe and comfortable reference path.

Just like the numerous options available for path-tracking controllers to choose from, there are numerous path-planning methods. Each method has its own advantages, disadvantages, features, and levels of complexity. Unlike a controller, path-planning methods cannot be compared with a few basic parameters because each path-planning method is created for its own purpose, satisfying its own application. There are a few relevant comparative works that have analyzed over 45 research articles and methods like the one shown in this[12]. The majority of the path-planning methods developed so far are based on Automated Guided Vehicles in urban driving or unknown terrain. Though there are several methods in path planning, they can be majorly grouped into two groups. One, that plans a long-term path that is collision-free based on graph search algorithms[13][14]. Two, that plans a collision-free path based on numerical optimization following discrete sampling[15][16].

The first type of path planning method is clearly not a suitable solution for the problem dealt with in this thesis. For autonomous trucks running in an environment-aware condition, it is not needed to choose a complex algorithm like this. Also, being computationally heavy, they are not feasible enough to handle dynamically changing environments like traffic. They are best used for an unknown environment that does not change. The other type of path planning methods seem promising for autonomous trucks as are sampled based on time. States are updated, the environment is updated, and the path is re-planned at a regular time interval. This ensures that any uncertainty in a known environment can be handled. Also, being computationally feasible, it provides room for developing complex path-tracking controllers.

The design of controllers must try to reduce the vehicle’s offset from the planned path and align the vehicle heading with the path heading in the path-following scenario. The geometry-based controllers are one of the two primary groups of lateral control design that are being employed extensively in autonomous vehicles. These controllers rely on the kinematic models of the vehicle as well as the geometry and coordinates of the desired path. The pure pursuit controller (PPC) uses a kinematic bicycle model, which combines the front left and right wheels into a single steerable wheel and the rear left and right wheels into a single driving wheel with an appropriate control-oriented model of a four-wheel vehicle. The difference between the path heading and the vehicle heading at the reference point along the path is the heading error. It is a key indicator of how well the vehicle is positioned relative to and traveling along the intended path. The kinematic bicycle model equations are used to calculate the rate of heading error $\dot{\psi}$, which aids in understanding how the heading error changes over time. Any of the three vehicle reference points may
also be used to describe the heading inaccuracy rate. The ideal heading rate of change for straight-line segments is zero, and it can be eliminated. This is due to the fact that the reference heading, which is now defined as the heading relative to the current path direction, is not time-varying for a straight line and is, in fact, equal to zero. The cross-track error is an offset error which is the other form of error. The distance between the vehicle’s reference point and the location that is closest to the intended path is known as the cross-track error. The distance between the vehicle’s current location and the target position along the course is primarily determined by this factor. The vehicle cannot effectively track the desired course until both heading error and cross-track error have decreased to zero. The path is perpendicular to the line connecting the vehicle reference point and the path reference point. By removing the lateral component from the forward velocity, it is possible to determine the rate of change of the cross-track error.

Due to the ease of PPC’s implementation, researchers have used it extensively. Several DARPA Grand Challenge participants have also used it. Rankin et al. examined a PID and a PPC controller using a test vehicle and simulation and discovered that PPC is steady and fairly precise at low vehicle speeds and small lateral errors. The other writers additionally verified that when vehicle speed increases, PPC reliability decreases. Furthermore, it has been asserted that the look-ahead distance, which is defined as how far along the path the truck should look from the current location, is simpler to adjust compared to the PID controller’s variables. In addition to the PPC, Rajagopalan et al. have utilized a receding horizon optimum control strategy in order to account for the influence of slip angle at greater speeds. Coulter et al. concluded that the velocity of the vehicle has a significant part in the dependability and effectiveness of the controller together with the look-ahead distance. Snider et al. further pointed out that because PPC lacks vehicle dynamics, tracking inaccuracy rises with speed and larger look-ahead distances are necessary. Heredia et al. conducted an analysis of PPC stability and created a criterion to identify the broad range of look-ahead distances and velocity values for which the controller is stable. Serna et al. used adaptive methods to determine the optimum look-ahead distance range. Park et al. also used Fuzzy Logic Systems (FLS) to fine-tune the look-ahead distance. Castaño et al. employed a variety of applications, such as steering a large vehicle and creating an adaptive PPC, by determining the proper look-ahead distance based on the GPS data for the path’s curve, the vehicle’s speed, and lateral error. Finally, Wit et al. suggested a novel geometric path monitor method built around the screw theory to address the PPC issues, taking into account both the direction and location of the vehicle at the target.

Stanley is an autonomous car developed by Stanford University’s Racing Team in cooperation with the Volkswagen Electronics Research Laboratory (ERL). It came
to recognition after winning the 2005 DARPA Grand Challenge [28]. The controller used in the car became famous as the 'Stanley Controller', and has been since implemented for lateral and longitudinal control of autonomous vehicles. The basic idea behind the controller is to try and minimize cross-track ($e$) and heading error ($\psi$) of the vehicle with reference to the desired trajectory using kinematic relations. These states are represented in Figure 4.3. The kinematic relations have been proven to always converge given permissible values of tunable parameters [28]. Thanks to its performance in the DAPRA Challenge, the controller is one of the promising options while choosing a strategy for lateral control of autonomous vehicles. It also forms the basis on which other modifications or controllers have been built on. Although the controller ideology is kinematic, the controller as developed by the Stanford Racing Team had a dynamic version on top of the default kinematic one. Nevertheless, inputs to this version were cross-track error $e$ and heading error $\psi$, while the output was the front wheel steering angle. These basic formulations resulted in 0.3 m cross-track error in the standard tests conducted for evaluation of the controllers [28]. This sort of error magnitudes and steering oscillations were compatible with off-road vehicles without passengers, but for on-road applications, the margins are often smaller due to the presence of passengers in the vehicle and other vehicles on the road. Hence, several modifications to the original Stanley Controller have been researched to improve tracking accuracy and driving stability.

Roads on which heavy vehicles are supposed to go are very complex, with unexpected obstacles and pedestrians appearing often. These surrounding aspects mean that any delay in the steering actuation of the vehicle may lead to large cross-track errors or also accidents in the worst case. Therefore, Lei et al. designed a path-tracking control strategy for the automatic driving of open-pit mine transportation vehicles based on a model predictive algorithm [29]. They conducted a model-based comparison study on pure pursuit control, Stanley control, and model predictive control (MPC) path tracking methods and ensured their solution to delay compensation can be implemented to all these path tracking methods. Where this modification makes the most difference is when we are dealing with high speeds and uncertainties, which is beyond the scope of our project. Another aspect to consider is that the road friction might not be enough to assume negligible slip, and hence, purely kinematic relations will not be enough. Specifically, in the case of low-friction roads, the wheels need to be turned more to achieve the same steering action as compared to the case with high-road friction. This makes kinematic path tracking very difficult on low-friction roads. Thus, Jeong et al. converted path tracking control to the yaw rate tracking one to cope with problems caused by low friction roads [30]. To generate a reference yaw rate for path tracking, they investigated several methods using a driver model and a reference trajectory. Better results were obtained using this version of the Stanley controller when compared to the original Stanley controller on low friction conditions. This version is not strictly necessary in the
ongoing project because the road is assumed to be dry, with enough friction for slip to be negligible.

The above modifications were directed toward accounting for various external factors and environments. Amer et al. hence proposed particle swarm optimization (PSO) to deduce optimal controller parameters depending on the vehicle under consideration \[31\]. In this research, a default Stanley controller for a particular vehicle is designed and then PSO is implemented to find the optimal controller parameters for that vehicle at any given time. It was found that the optimized controller decreased the overall lateral error to a large extent. It concluded that the optimization does not need to be done in real time, rather can choose to retune the controller, thus saving computational burden. This feature of re-tunability of controller parameters is not mandatory for Volvo Trucks as they have set the vehicle parameters. The controller, by default, can be designed for manual tuning. Considering environmental and vehicle-related aspects and uncertainties separately has yielded positive benefits over the default Stanley controller as described above. Combining both the external and vehicle aspects to make the controller better AbdElmoniem et al. focused on the predictive Stanley controller \[32\]. The original Stanley controller restricts the vehicle’s ability to handle sudden changes in the trajectory heading angle because the basic design of the controller relies just on the closest point on the trajectory at each instant. Hence proposed a new approach that mimics human behavior while driving. This method is based on a discrete prediction model that anticipates the future states of the vehicle, allowing the use of the control algorithm in future predicted states augmented with the current controller output. Results obtained from the proposed control approach show advantages and performance improvement in reducing lateral error and ensuring yaw stability as compared to the original Stanley controller. This method, although very intriguing, is essentially model predictive control, which is tackled separately in our project.

Mouri et al. have demonstrated that the linear quadratic regulator (LQR) controller performs significantly better than the proportional differential (PD) controller when traversing a curved path \[33\]. Both Tabatabaei Oreh et al. and Zong et al. used the concept of employing a LQR controller \[34\] \[35\]. Khalaji et al. created a robust adaptive dynamic controller that can produce good tracking results by combining a feedback linearization-based dynamic controller with a Lyapunov-based kinematic control rule \[36\]. Comparing the LQR to a Dominate Second-Order Pole based controller, Salerno et al. found that the LQR demonstrated more resilient performance in the presence of the system’s unmodeled dynamics and equivalent robustness in the presence of parametric uncertainty \[37\]. Also, when navigating highly curved pathways, the LQR unfavorably produces steady-state errors \[1\]. Kim et al. was suggested employing an additional feedforward controller to enhance tracking performance for pathways with dynamic curvatures \[38\]. Sorniotti et al. infer that
the feedback controller takes the cross-track and yaw error from the state feedback, while the feedforward controller uses information about the road’s curvature \[39\]. Zhang et al. developed an ideal preview using the feedforward LQR controller to address the system’s overshoot response \[40\]. This work used lateral and yaw error to create the feedforward controller, which was built on the trajectory information for a preview horizon. The usage of Iterative LQR (ILQR) has also been recommended by H.j. to account for potential system disturbances \[41\]. Lee et al. presented an observer-based LQR to increase tracking accuracy in the event of parametric uncertainty \[42\]. Peng et al. additionally recommended Frequency-Shaped Linear Control (FSLQ) to boost robustness \[43\]. Robust LQR (RLQR) has also been used by Hu et al. to address the issue of model uncertainty and external disruptions \[44\]. In this study, a second robust controller was added to reduce the impact of parametric and nonlinear uncertainties.

Model Predictive Control (MPC) stands as a sophisticated control method employed across industries, robotics, and autonomous systems. It optimizes control inputs based on future predictions, accounting for system dynamics and constraints to guide systems effectively. The accurate tracking of desired trajectories is a critical aspect of controlling autonomous mobile systems. The Model Predictive Control (MPC) strategy has emerged as a promising solution by utilizing predictive models of system behavior to optimize control actions over a finite prediction horizon. This approach enables the integration of complex dynamics and constraints into the control design. MPC is a receding horizon control strategy that repeatedly optimizes control inputs over a prediction horizon while implementing only the initial optimal input. This process is reiterated at each time step, accounting for updated system states and disturbances. The formulation involves a cost function that considers the desired trajectory, system states, and control inputs, subject to system dynamics and constraints. MPC offers several distinct advantages for path tracking applications like handling nonlinear dynamics - MPC can effectively manage intricate nonlinear system dynamics, ensuring accurate trajectory tracking even in challenging conditions; incorporating constraints - the inherent capability of MPC to accommodate various constraints, such as input limitations, obstacle avoidance, and safety margins, enhances system stability and safety; and adaptability - MPC’s predictive nature allows it to adapt to evolving conditions and disturbances, making it well-suited for real-world scenarios.

Recent research endeavors aim to enhance the performance and applicability of MPC-based path-tracking controllers through the integration of deep learning - some studies explore the fusion of MPC with deep learning techniques to enhance predictive models and controller robustness; adaptive MPC - researchers are investigating adaptive MPC strategies to address varying system dynamics and uncertainties; and real-time implementation - efforts are underway to alleviate the
computational demands of MPC algorithms, allowing for real-time deployment on resource-constrained platforms. Several challenges persist in the field of MPC-based path tracking controllers like computational complexity, modeling precision, and parameter tuning. The real-time execution of MPC can be computationally intensive, necessitating efficient algorithms and hardware resources. Developing accurate predictive models is pivotal, and uncertainties in the model can lead to suboptimal controller performance. Appropriate selection of prediction horizon, control weights, and other parameters significantly influences controller efficacy.

MPC can be briefly classified into two types, namely linear MPC and nonlinear MPC. A linear MPC uses a linear model of the system to capture system dynamics. It has linear constraints and solves linear or quadratic open-loop performance objectives for control optimization [45]. On the contrary, nonlinear MPC uses a nonlinear model of the system to capture system dynamics. It can have nonlinear constraints and solve nonlinear optimization problems. Though linear MPC is computationally less burdening than nonlinear MPC, it is only suitable for applications with linear and moderately nonlinear system dynamics. Linear MPC can be further divided into linear time-invariant MPC (LTI-MPC), linear time-varying MPC (LTV-MPC), and linear parameter-varying MPC (LPV-MPC). Several previous works have been conducted with these types of controllers for path-tracking applications. It can be concluded that nonlinear MPC outperforms linear MPC in path tracking by substantial margins [46]. Particularly for articulated vehicles, it proves to be a better solution than linear MPC for path tracking. A detailed comparison of computation time taken by linear and nonlinear MPC for the same path-tracking application was presented in the article [47]. Linear MPC is much easier for the CPU to handle as compared to nonlinear MPC. Though nonlinear MPC proves to be better with performance, it takes a great amount of effort to develop, tune, and run nonlinear MPC in real-time applications. It can be understood that linear MPC is capable of providing good performance at lowered computation cost [48]. It is clearly stated that “linear MPC comes with a disadvantage that the linearized model is only valid for points near the reference trajectory”. Furthermore, Bai et al. provide an extensive comparison both with its literature review and their own work that the above-discussed characteristics of linear and nonlinear MPC hold good [49]. Work conducted by Kong presents a clear distinction between kinematic and dynamic bicycle models, which are used as the prediction model for developing MPC controllers [50]. A novel method of developing a controller based on forecast error is also discussed. An integrated kinematic and dynamic vehicle model is presented, and asymptotic stability is proven in [51]. Performance is tracked for varying velocities and varying friction coefficients. MPC is not just limited to one form of MPC or the different models discussed above, it also varies with different forms like robust MPC, hybrid MPC, stochastic MPC, data-driven MPC, etc.
One of the control design strategies to regulate the uncertainties and disturbances occurring in the systems is the sliding mode control (SMC) method, which was first introduced in the early 1950s [52]. The controller introduces states (error) to a sliding surface, and from that point on, the states are obligated to stay on the sliding surface. The sliding phase and the reaching phase are the two parts of the sliding mode control method. Separate control laws can be constructed for the equivalent control and the switching control, which correspond to two phases. In this methodology, an initially selected sliding surface is employed, followed by the design of a controller to guide the system states toward that surface within a finite time while adhering to constraints. One notable benefit of the SMC approach is the reduction in the system’s dynamic order when the state resides on the sliding surface, simplifying the control process significantly. As the chosen sliding surface is based on vehicle non-linear dynamics, many strategies can be utilized to create the sliding surface and accompanying control rule. In emergency maneuvers, the lateral control system must improve the vehicle’s stability and handling. In this instance, the vehicle’s response to the driver’s input is mostly nonlinear. Additionally, there are a number of variables that affect the dynamics of the vehicle, including the inertia parameters, the friction coefficient, and the cornering stiffness of the tires. In order to manage the complicated and non-linear vehicle dynamics, an adequate control system is required.

It appears that there is no consistent pattern in the choice of the control variables and the establishment of their reference values when it comes to large trucks [53]. Yang et al. implemented a feedback linearization-based continuous-time SMC technique, making certain assumptions that the initial location of the reference trajectory coincides with the origin and that there exists a non-perpendicular orientation between the coordinate angle and the heading angle [54]. A polar coordinate system was used by Chwa et al. as part of an enhancement effort to get rid of the necessary geometric restrictions for the controller’s effective operation [55]. When the system’s state is close to its beginning and the physical constraints of the controllers are not taken into account, these proposed controllers do not produce adequate results, but they function well when there are uncertainties and outside disruptions. The ‘chattering’ problem is one of the basic issues with SMC [52]. It demands a lot of control activity is not ideal for regular control applications, and is caused by discontinuous shifting and the time delay of the control activities. Additionally, it could result in undesirable oscillations that excite unmodeled high-frequency dynamics.
Chapter 3

System Modelling

The various physical models of the tractor-trailer system’s lateral dynamics are introduced in this chapter, along with the methods for identifying the system and estimating the model’s parameters.

This work examines a variety of models, which can be divided into two groups: kinematic models and dynamic models. Kinematic constraints—i.e., the fact that the wheels can only move in the direction they point—are used to develop kinematic models. Dynamic models, on the other hand, originated from Newton’s principles regarding the system’s force and torque equilibrium. All types are single-track bicycles, featuring one wheel in place of the left and right wheels of an axle. Here is a kinematic model that only represents the tractor used for a simple control. Additionally, two dynamic models with escalating complexity are presented with tractor and tractor-trailer combinations.

In its most basic form, a heavy vehicle consists of two units: a tractor and a semi-trailer. The fifth wheel, a mechanical coupling point, connects the trailer unit, which is used to haul large freight, to the tractor unit, which is operated by the driver. In general, the driver should be able to control the tractor unit with ease, and its behavior should be predictable. The possibility of an incorrect driver reaction can be prevented by adopting an appropriate control technique to modify the tractor unit’s dynamic behavior. Therefore, there have been a number of research initiatives to look at the simultaneous active lateral control of the tractor unit and the trailer unit.
3.1 Geometric based Lateral Controllers

When choosing steering commands to follow a reference path, a geometric path tracking controller relies on the kinematic vehicle model. Geometry-based controllers are any controllers that track a reference path solely based on the geometry of the vehicle’s kinematics and the reference path. It is a kind of lateral controller that takes the no-slip condition at the wheels into account and disregards dynamic forces acting on the vehicles. However, geometric route tracking controllers can be very effective when the vehicle is running in the linear tire region and the tire is not saturated. The primary objective of lateral control is to track changes in the direction of the path as they emerge and choose the steering angle necessary to rectify any errors that build up. The error between the vehicle location and the exact desired path coordinates must be taken into account while designing the lateral controller, and it must be driven to zero while still respecting steering angle constraints. By keeping in mind, the dynamic restrictions of the vehicle and desired ride qualities, such as the maximum lateral acceleration and the least amount of jerk. When compensating for tracking mistakes, the control command must be aware of the available tire forces and not go beyond what the vehicle is capable of. There are several approaches to define the reference path, which serves as a key interface between the planning system in the lateral controller. In this project, predefined track files with global X and Y position coordinates have been fed into the controllers as road geometry input files.

3.2 Model-based Lateral Controllers

Model-based controllers are predictive in nature. A model of the system being controlled is used to predict the behavior of the system and estimate the required control input accordingly. Model-based controllers like Linear Quadratic Regulator, Sliding Mode Control, and Model Predictive Control use a vehicle model to estimate the required control input for reference tracking. The vehicle model can be adapted to be a point mass model, kinematic bicycle model, dynamic bicycle model, or a full car model. Depending on the complexity of the model chosen, it provides varying degrees of control. Unlike geometric controllers, model-based controllers can handle limitations like no-slip assumptions and small angle approximations. They can also adopt nonlinear vehicle models for nonlinear control strategies. Model-based controllers also offer flexibility in modeling state space equations to target specific objectives.
3.3 Variable Look ahead distance method

Appropriate look-ahead distance adjustment is one of the key prerequisites for effective geometric controllers. When the path curvature is high for a given look ahead value, it can result in the effect of cutting the corner. On the contrary, a look-ahead distance that is too close, despite offering superior accuracy, may cause the vehicle to behave in an oscillating manner. Similarly, model-based controllers like Model Predictive Control have a prediction horizon. The prediction horizon is the number of time steps in the future that the optimization problem is going to make a prediction. It predicts the future states and compares against a set of reference values that lies ahead of the current state, which is the look-ahead distance. Thus, look ahead distance plays a role in Model Predictive Control as a part of reference generation and thus the optimization problem. The velocity of the vehicle has a significant part in the dependability and effectiveness of the controller together with the look-ahead distance. Along with this, the geometrical controllers can be tuned to get better performance and lower errors by keeping a balance between the tracking and the stability of the truck. Another way to improve the tracking could be to reduce the velocity at the turns and add a longitudinal controller to track the reference velocity. Hence velocity profiling is used.
Chapter 4

Lateral Control Strategies

4.1 Collision Avoidance System

4.1.1 Design

The primary goal of designing a collision avoidance system in this thesis is to assist all the selected path-tracking controllers in achieving the objectives of safe and comfortable driving. For path tracking with safe and comfortable driving, path planning and velocity profiling are necessary. Without path planning, smooth trajectories may not be followed and collisions cannot be avoided. Velocity profiling is needed to ensure smooth acceleration and deceleration provided the vehicle is stable.

Collision avoidance systems can be implemented using different methods as discussed before in the literature review. A collision avoidance method based on geometric relationships and cubic polynomials was chosen as the solution for our case. The reason behind implementing this method is that its computation burden is low, it updates for every time step, can handle unexpected obstacles, and corrects vehicle position and heading angle at the terminal states irrespective of initial states. This work is an addition to what has been implemented in [56]. Here, only the kinematic feasibility of the vehicle is considered. Also, the geometric relationship and thus the constraints that ensure collision avoidance are not addressed. The kinematic feasibility of the vehicle is ensured by regulating the velocity at which the vehicle can travel in the trajectory decided in [56]. Roll dynamics was not a deciding factor in velocity profiling constraints, which is being implemented in this work. The reason behind these additions to the previous work is that we are dealing with heavy vehicles. Hence it is extremely important to consider vehicle rollover and instability while path planning. The details and flow of work in the collision avoidance system are discussed in the following subsections and an illustration is provided in Figure 4.1.
Variable look ahead distance
Trucks nowadays are equipped with multiple sensors like cameras, RADAR, LiDAR, etc. for perception. Data from different sensors are fused to visualize and localize the truck and environment. Beyond a certain distance, there is a compromise on the accuracy with which the vehicle can perceive its environment. Judging based on experience, the maximum look ahead distance was set to 100 meters for detecting and locking obstacles and creating alternate trajectories. The look-ahead distance varies based on the velocity with which the vehicle travels. A linear relationship between minimum and maximum look ahead distance is set for this purpose.

Alternate trajectories
In the specified route, the vehicle’s current state shall be called as the initial state. The interception of look-ahead distance on the road ahead of the vehicle forms the terminal state. Since the aim is to create alternate trajectories, the terminal state consists of five different points across three different lanes. This can be altered as per the number of lanes available in a road. In this thesis, a three-lane highway road where the rightmost lane is assumed to be the base frame or the route. For every time step, the position of the vehicle in global X and Y coordinates along with the heading angle is observed. Five different positions across the different lanes at the end of the look-ahead distance and with the heading angle of the road is set as the terminal states. A cubic polynomial is mapped out from the initial state to the terminal states. The cubic polynomial forms the five different alternate trajectories for every time step the vehicle runs.

First obstacle
With a certain look-ahead distance for every time step, the truck can sense obstacles around it. In this thesis, only forward collision is considered. This can easily be extended to side and rearward collisions with similar algorithms. Once an obstacle is detected within the look-ahead distance and across terminal states laterally, the relative velocity and distance between the target and ego vehicle are estimated. If multiple obstacles fall in the detection range, the first obstacle ahead of the ego vehicle is locked for consideration in the remaining steps of the algorithm. A safe velocity is also calculated at every time step for limiting the velocity profile. The velocity with which it is possible to bring the ego vehicle to a complete halt with a certain safe distance between the ego and the target vehicle is defined as safe velocity.

Collision test
The ego vehicle after detecting the obstacle slows down to safe velocity. The information about relative velocity and distance is updated for every time step. It is
important to check if there is a possibility of collision. If a scenario arises where it is not possible to halt the vehicle to a standstill, a collision test is conducted where the five different alternate trajectories are checked if they intersect with the obstacle lying ahead. The obstacle is modeled as a moving ellipse with a safety factor. The alternate trajectories are also given two offset curves along the two ends of the track width of the ego vehicle. If and only if each pair of offset curves does not intersect with the ellipse-modeled obstacle, that particular alternate trajectory is labeled as a free trajectory.

**Velocity profiling**

A velocity profile is generated for the free trajectories based on the speed limit of the road, curvature of each trajectory, roll dynamics, safe velocity, and longitudinal and lateral acceleration limit. These limits are used to create an upper bound and lower bound of the velocity across each alternate trajectory. For ensuring comfortable driving, the velocity profile between initial state and terminal states is also equated as a cubic polynomial.

\[
\dot{x}_{\text{baseframe}} = \sqrt{a_{\text{ymax}} \times k_{\text{baseframe}}} \tag{4.1}
\]

\[
V_{\text{limit1}} = \sqrt{a_{\text{ymax}} \times (k_{\text{traj}})_{\text{max}}} \tag{4.2}
\]

\[
V_{\text{limit2}} = (\dot{x}_{\text{baseframe}})_{\text{current state}} \tag{4.3}
\]

\[
V_{\text{limit3}} = \sqrt{2 \times a_{x_{\text{max}}} \times (\text{dist})_{\text{egototarget}}} \tag{4.4}
\]

\[
V_{\text{limit4}} = \sqrt{a_x \times (d_{\text{safe}} - (\text{dist})_{\text{egototarget}} + (v_{\text{init}})^2 + v_{\text{fin}})^2} \tag{4.5}
\]

\(\dot{x}_{\text{baseframe}}\) is the speed limit of the road. \(a_{\text{ymax}}\), maximum lateral acceleration is the minimum of \(a_{\text{yslip}}, a_{\text{yroll}}\) and \(a_{\text{ylegislation}}\). \(a_{\text{yslip}}\) is equal to the product of road friction and acceleration due to gravity. \(a_{\text{yroll}}\) is calculated using a wheel lift-off condition and \(a_{\text{ylegislation}}\) is the legal restriction on lateral acceleration. \(a_{x_{\text{max}}}\) is the maximum longitudinal acceleration. Work done in [57] is similar to the \(a_{\text{yroll}}\) implementation performed. \(k_{\text{baseframe}}\) is the curvature of the track or base frame. \(k_{\text{traj}}\) is the curvature of the trajectory. Several limits like \(V_{\text{limit1}}\) to \(V_{\text{limit4}}\) are used to find the least allowed velocity through different velocities. Once the minimum velocity is found, a velocity profile from the current state to the terminal state is modeled as a cubic polynomial with limiting values. \(\text{dist}\) is the distance between the ego and the target vehicle. \(d_{\text{safe}}\) is a predefined constant suggesting a safe amount of distance between two vehicles for braking smoothly. \(v_{\text{init}}\) is the current velocity. \(v_{\text{fin}}\) is the terminal velocity calculated based on current velocity, available longitudinal acceleration, and length of the trajectory.
Chapter 4 – Lateral Control Strategies

Optimal trajectory
At the end of the above-mentioned process, information about free trajectories, that is their global X and Y coordinates, heading angles, curvatures and velocity profiles are available. A minimization function based on four different factors decides the optimal trajectory amongst the free trajectories. The deciding factors are as follows:

- **Length of trajectory**
  \[ J_p = \frac{S_{max} - S_{traj}}{S_{max}} \]  
  (4.6)

- **Smoothness criteria**
  \[ J_s = \frac{1}{S} \int_0^s \frac{|k_{traj}(s)|}{k_{max}} ds \]  
  (4.7)

- **Deviation from the base frame**
  \[ J_d = \frac{1}{S} \int_0^s \frac{|D_{traj}(s)|}{D_{max}} ds \]  
  (4.8)

- **Path inconsistency between every time step**
  \[ J_c = \frac{|l_{traj} - l_{p_{traj}}|}{l_{max}} \]  
  (4.9)

Different alternate trajectories are of different lengths, the trajectory being furthest away from the base frame being the longest of all while the vehicle is on the base frame. It is desirable to choose longer trajectories for a smoother transition of the vehicle. It is also important to be conservative in choosing a trajectory in terms of required lateral movement and deviation from the base frame. The trajectory which is just sufficient to avoid collision and stays closer to the base frame is given priority. Lastly, the trajectory chosen should gradually transition over a time period through several time steps. For example, it would be undesirable to choose the opposite extremum trajectories between consecutive time steps, rather a gradual change from one extremum to the other over several time steps as the situation requires is desirable. The minimization problem considering the four factors mentioned above with their own weights \( w_p, w_s, w_d, \) and \( w_c \) is as follows.

\[ N = \min(w_p \ast J_p + w_s \ast J_s + w_d \ast J_d + w_c \ast J_c) \]  
(4.10)

The minimum value \( N \) translates to the trajectory number, making it the chosen trajectory or optimal trajectory as a reference for the path-tracking controllers. \( S \) represents the length of each trajectory and \( S_{max} \) is the maximum among all \( S \). \( s \) is the smallest unit of arc length. \( k_{traj} \) is the curvature of trajectories and \( k_{max} \) is the maximum among them. Similarly, \( D_{traj} \) and \( D_{max} \) represent the lateral offset between different trajectories and base frame and its maximum, respectively. \( l_{traj} \) is the lateral offset of each trajectory, \( l_{p} \) is the lateral offset of previous trajectory and \( l_{max} \) is the maximum of all lateral offset.
4.1.2 Implementation

An expected behavior of the ego vehicle with any of the six chosen controllers, integrated with the collision avoidance system can be seen in Figure 4.1. For a reference path with a static obstacle and ego vehicle starting with zero velocity, it can be observed that the alternate trajectory creation is an iterative process. For every time step, new alternate trajectories are created as the vehicle progresses. With varying velocity, the look-ahead distance varies and so does the length and curvature of the alternate trajectories. For every time step, an optimal trajectory is chosen.

![Figure 4.1: Expected path tracking behavior with collision avoidance system.](image)

The optimal trajectory’s information about global X and Y coordinates, heading angle, Ackermann steering angle, curvature, velocity profile, etc. are given as output from the collision avoidance system as a vector. Here, Ackermann’s steering angle is defined as a specific angle set for the wheels on an axle to ensure that when a truck makes a turn, the inner and outer wheels follow paths of different radii. In other words, it’s a way to ensure that all wheels travel along concentric circles when
Chapter 4 – Lateral Control Strategies

the vehicle is turning, which results in smoother and more efficient cornering. The vector consists of information needed to travel from the initial state to the terminal state, modeled as a cubic polynomial. For reactive controllers like Pure Pursuit and Stanley controllers, only one of the elements of the vector can be fed as a reference. So this becomes a tuning factor for such controllers. For controllers that have a horizon, a set of elements from the optimal trajectory vector can be fed.

4.1.3 Inference

Since the trajectories are created using cubic polynomials, the slope is almost zero at the beginning and end of the curve. The change in lateral direction occurs majorly in the middle of the curve. Though cubic polynomials ensure a smooth transition and maximum straight line stretch, the aggressiveness of turn can only be understood by controllers that have a horizon. For controllers that are single input single output (SISO), it becomes important to tune the index of the input vector. Also, the smaller the time step of the whole system, the smoother the transition of the vehicle. In a situation where all the alternate trajectories are colliding with the obstacle, there are no feasible trajectories for the minimization function to work with. Then, the immediate course of action that the collision avoidance system generates is to provide zero velocity and current states as input over the whole horizon or input vector. This replicates an emergency braking scenario, which is what can be done to avoid or alleviate the impact of a collision.

4.2 Pure Pursuit Controller

PPC builds a steering command rule to achieve path tracking using a kinematic bicycle model and error monitoring. In this control approach, a look-ahead distance—a measurement of the distance in front of the vehicle—is used to select a target point on the reference path. By fitting an arc between the rear axle and the point of the target, it is possible to determine the relationship between the exact location of the vehicle and its trajectory of the path. It is incredibly popular and helpful in autonomous driving due to its simplicity. Geometric path tracking controllers depend on a point of reference along the desired path, which could be the same reference point used to calculate heading and cross-track errors or it can be a look-ahead point a distance in front of the vehicle along the path, as shown as a target point in red dot color in Figure 4.2.

The look-ahead point on the reference path is used by the pure pursuit controller that is going to be derived. The fundamental concept behind the pure pursuit approach is that a reference point may be set up on the path ahead of the vehicle at a given distance, and the steering instructions required to intersect with this point can be calculated using a constant steering angle. The point keeps moving forward, reducing the steering angle, and gently pulling the vehicle towards the path as it
turns towards the route to follow this curve. The look-ahead distance, which is represented by the dashed line in Figure 4.2(a), is a fixed distance that connects the center of the rear axle to the target reference point on the vehicle, which is utilized as the reference point in this approach. The angle between the look-ahead line and the vehicle's body heading is $\alpha$.

![Diagram of Vehicle Path Tracking](image)

Figure 4.2: Geometrical Representation of Vehicle Path Tracking Figures [59].

The idea of the instantaneous center of rotation is factored in when building the pure pursuit controller. A triangle with two sides of length $R$ and one of length $l_d$ is formed by the target point on the trajectory, the center of the rear axle, and the instantaneous center of rotation. The path's target point from the vehicle reference point, as shown in Figure 4.2(b) above. This arc represents the region of the ICR circle that encompasses the $2\alpha$. It is possible to derive the angle $2\alpha$ using common trigonometric identities. By using the law of sines and trigonometric identities, the following equations are derived:

\[ \frac{l_d}{\sin 2\alpha} = \frac{R}{\sin\left(\frac{\pi}{2} - \alpha\right)} \tag{4.11} \]

\[ \frac{l_d}{2 \sin \alpha \cos \alpha} = \frac{R}{\cos \alpha} \tag{4.12} \]

\[ \frac{l_d}{\sin \alpha} = 2R \tag{4.13} \]
The path curvature $k$, is the inverse of the arc radius $R$. A kinematic bicycle model that represents the truck is used here to determine the steering angle required to follow this arc. To be remembered that the steering angle determines the arc radius and produces the relationship $\tan \delta$ equals the truck length $L$, over the arc radius $R$. Now the steering angle to be defined is required to follow the arc in terms of clearly calculable quantities by combining this formula with the expression for $R$ derived earlier.

$$k = \frac{1}{R} = \frac{2 \sin \alpha}{l_d} \quad (4.14)$$

$$\delta = \tan^{-1}(kL) \quad (4.15)$$

$$\delta = \tan^{-1}\left(\frac{2L \sin \alpha}{l_d}\right) \quad (4.16)$$

The distance between the heading vector and the target point is what one may use to define the cross-track error for the pure pursuit controller. Combining these formulae demonstrates that the cross-track error at the look-ahead reference point is inversely correlated with the curvature of the path generated by the pure pursuit controller. The vehicle is forcefully brought back to the path as the error grows along with the curvature. This equation explains how the pure pursuit controller uses route curvature as the controller’s output to rectify cross-track error in a way similar to proportional control. This controller design will demonstrate how the effects of parameters are affected, and how the algorithm will be given in a way that reduces the complexity of performance tuning.
4.3 Stanley Controller

The schematic Figure 4.3 represents the simple control technique of the Stanley controller.

![Figure 4.3: Stanley Controller setup for a bicycle model](image)

In the most primitive form, the Stanley controller outputs the following steering command for the front wheels:

\[
\delta(t) = \delta_{\text{min}} < \psi(t) + \arctan \left( \frac{ke(t)}{v(t)} \right) < \delta_{\text{max}} \tag{4.17}
\]

where \( k \) is a tunable gain to make the cross-track negation more aggressive. The dynamic version considered here aspects like the yaw rate and rate of change of steering input, to account for the comfort and practical aspects of implementation. This version is shown below:

\[
\delta(i) = (\psi(i) - \psi_{\text{SS}}(i)) + \arctan \left( \frac{ke(i)}{k_{\text{soft}} + v(i)} \right) + k_{d,yaw}(r_{\text{meas}} - r_{\text{ traj}}) + k_{d,steer}(\delta_{\text{meas}}(i)\delta_{\text{meas}}(i - 1)) \tag{4.18}
\]

where subscript \( \text{SS} \) means steady state, \( k_{\text{soft}} \) ensures stability at low \( v \), \( k_{d,yaw} \) dictates rate of change of yaw rate \( r \) and \( k_{d,steer} \) dictates rate of change of steering.
command $\delta$. All these parameters are tuned as per the application and desired tracking and comfort outcomes.

Crucial parameters for a functioning front wheel steering Stanley controller are cross-track error $e$ and heading error $\psi$. $e$ is then the distance from the vehicle COG to the closest point on the reference trajectory. $\psi$ is the angle between the vehicle direction and trajectory direction at the closest point. Determining $e$ and $\psi$ in real time is the main challenge in Stanley Controller. This is typically done numerically at each instant because the reference trajectory and position are both known as points in space. Quadratic interpolation gives a continuous path depending on the discrete reference points. This path serves as the 'constraint' for minimization. Distance between the vehicle’s current position and reference path in space is the 'cost function' to be minimized. Finding the point closest to the vehicle’s center of gravity on the path is done numerically as a minimization problem given the cost function and constraint. In this project, a standard block from MATLAB-Simulink along with the formulation using mathematical function blocks is used for this purpose. Upon determining $e$ and $\psi$, generating the steering commands from these inputs involves just kinematic calculations as shown below:

$$
\delta_f = -\frac{k_\psi \psi}{v} - \arctan \left( \frac{k_e}{v} \right) + \frac{1}{v} \arcsin (f \kappa \cos(\psi))
$$

(4.19)

Where $k_\psi$, $k$ are tunable parameters that make the respective terms more aggressive, while $\kappa$ is the curvature of the reference path at any instant.

### 4.4 Linear Quadratic Regulator

The primary goal in control techniques is to identify a sequence of control inputs that effectively align the system’s output with predetermined reference trajectories. When the system’s output is expected to consistently remain at zero, this scenario is referred to as a regulation problem, with zero as the reference value. The linear quadratic regulator (LQR) technique for vehicle dynamics has been studied and proposed to enhance passenger comfort and road handling for heavy vehicle applications. Lyapunov candidate is chosen with the help of the solution of the algebraic riccati equation and thus obtain stability through LQR, that attracted attention after a thorough analysis of control methods. The LQR methodology, which is based on optimal theory and achieves the optimal control law with the help of state linear feedback, is proven to be straightforward and simple to use. It uses low-cost methods to improve the performance of the original system. Surprisingly, LQR control is overly reliant on explicit dynamic models, and when it is used in practice, numerous constraints cannot be implemented. To enhance the tractor-semi trailer’s lateral stability at high speeds, LQR is envisioned in this project work. A good way
Chapter 4 – Lateral Control Strategies

to balance system response and control effort is to use LQR. The process of LQR design involves optimizing the performance index while adhering to specific constraints, ultimately obtaining the optimal feedback controller through the solution of the associated algebraic Riccati equation. The idea of optimum control is focused on running a dynamic system efficiently. Here, design parameters for penalizing the state variables and control signals can be the weighting parameters Q and R.

In the LQR control strategy, for an articulated semi-trailer and truck body combination, the truck’s lateral velocity \( V_y \), yaw angle \( \psi_t \) and articulation angle \( \tau \) represents state variables, whereas steering angle \( \delta \) is a control variable. The control system’s objective is to control the tractor’s lateral velocity, yaw rate, and articulation angle with the semitrailer unit. The suggested control system directs the tractor and trailer’s wheels in order to get the desired results. Here, finite horizon, continuous time LQR is formulated which determines the optimal feedback gain by using a non-linear model of the vehicle.

\[
M^* \dot{x}_1 = A^* x_1 + C^* u_1
\]

(4.20)
The above equation can be represented in state space by multiplying with \( M^{-1} \) and the matrices \( M^*, A^* \) and \( C^* \) are described in the Appendix. where,

\[
x_1 = \begin{bmatrix} V_y & \psi_t & \tau \end{bmatrix}^T \text { and } u_1 = \delta
\]

(4.21)
The equations in state space form are represented as,

\[
\dot{x}_1 = A_1 x_1 + B_1 u_1
\]

(4.22)
For the purpose of designing the LQR controller, the tracking error can be described as

\[
e_1 = \begin{bmatrix} V_y \\ \psi_t \\ \tau \end{bmatrix} - \begin{bmatrix} V_{yr} \\ \psi_r \\ \tau_r \end{bmatrix} = x_1 - x_r
\]

(4.23)
The differential equation governing the tracking error can be stated as follows by deriving it from the above relationship:

\[
\dot{e}_1 = \dot{x}_1 - \dot{x}_r = A_1 x_1 + B_1 u_1 - \dot{x}_r - A_1 x_r + A_1 x_r
\]

\[
= A_1 (x_1 - x_r) + B_1 u_1 - \dot{x}_r + A_1 x_r
\]

(4.24)
As LQR provides steady-state errors while negotiating the dynamic high curvature paths and to improve the performance of tracking, an additional feed-forward controller has been designed. The control input can now be divided into two categories: feedback and feedforward.

\[
u_1 = u_{fb} + u_{ff}
\]

(4.25)
From the above equations, the error rate can be calculated as,

\[ \dot{e}_1 = A_1 e_1 + B_1 u_{fb} + B_1 u_{ff} - \dot{x}_r + A_1 x_r \]  

(4.26)

As follows is the definition of the feedforward control input,

\[ u_{ff} = B_1^{-1}(\dot{x}_r - A_1 x_r) \]  

(4.27)

The feedforward’s goal is to follow the reference signal, whereas the feedback’s goal is to correct for modeling inaccuracies and other erroneous disturbances. The error equation is now represented using the standard form by combining the aforementioned relationships as below,

\[ \dot{e}_1 = A_1 e_1 + B_1 u_{fb} \]  

(4.28)

The optimal control rule is used to define the feedback control. If the following functional index is minimized, the ideal feedback control input can be obtained.

\[ J = \int_{0}^{\infty} (e_1^T Q_1 e_1 + u_{fb}^T R_1 u_{fb}) \, dt \]  

(4.29)

Lastly, the definition of the control input is as follows:

\[ u_{fb} = -K_1 e_1 \]  

\[ K_1 = R_1^{-1} B_1^{-1} P_1 \]  

(4.30)

By solving the following algebraic Riccati equation, the response of \( P_1 \) is calculated.

\[ A_1^T P_1 + P_1 A_1 - P_1 B_1 R_1^{-1} B_1^T P_1 + Q_1 = 0 \]  

(4.31)

So, the final control input can be written as,

\[ \delta = u_1 = u_{fb} + u_{ff} \]  

(4.32)

Additionally, when a minor parameter variation results in quickly unstable conditions, the selection of the weighting matrices becomes crucial to the system’s stability.
4.5 Sliding Mode Controller

Based on basic SMC Law, it is necessary to construct a control law for the sliding surface that drives the error asymptotically to zero \((s_1 = 0, s_2 = 0)\). The design of a sliding mode controller can be explained by considering the nonlinear model of the vehicle system and its homogeneous linear time-invariant approximation. For designing lateral and trajectory control, the traditional first-order SMC was chosen.

\[
\dot{x}_1 = x_2 \quad \text{and} \quad x_2 = u + f(x_1, x_2, t) \tag{4.33}
\]

\[
\dot{x}_1 + cx_1 = 0 \tag{4.34}
\]

Where \(u\) is the control force, and \(f(x_1, x_2, t)\) is the nonlinear disturbance. Since \(x_2(t) = \dot{x}_1(t)\) a general solution of the above equation and its derivative is given by,

\[
x_1(t) = x_1(0)e^{-ct} \quad \text{and} \quad x_2(t) = \dot{x}_1(t) = -cx_1(0)e^{-ct} \tag{4.35}
\]

Both \(\dot{x}_1\) and \(\dot{x}_2\) converge to zero asymptotically. Note, no effect of the disturbance \(f(x_1, x_2, t)\) on the state compensated dynamics is observed. To achieve these state-compensated dynamics, a new variable \((\sigma)\) is introduced in the state space of the system in Equation 4.33,

\[
\sigma = \sigma(x_1, x_2) = x_2 + cx_1, \quad c > 0 \tag{4.36}
\]

Here, \(\sigma\) which is called as a sliding variable. In order to achieve asymptotic convergence of the state variables \(x_1\) and \(x_2\), with a given convergence rate as in Equation 4.35 in the presence of the bounded disturbance \(f(x_1, x_2, t)\), need to drive the variable \(\sigma\) in the Equation 4.36 to zero in finite time by means of the control \(u\). This task can be achieved by applying Lyapunov function techniques to the \(\sigma\)-dynamics that are derived from Equation 4.33 and 4.36.

\[
\dot{\sigma} = cx_2 + u + f(x_1, x_2, t), \quad \sigma(0) = \sigma_0 \tag{4.37}
\]

For the \(\sigma\)-dynamics above, a candidate Lyapunov function is introduced taking the form,

\[
V = \frac{1}{2}\sigma^2 \tag{4.38}
\]

In order to provide the asymptotic stability of the Equation 4.37 about the equilibrium point \(\sigma=0\), the following conditions must be satisfied.

a) \(\dot{V} < 0\) for \(\neq 0\)

b) \(\lim_{|\sigma| \to \infty} V = \infty\)

Condition (b) is obviously satisfied by \(V\) in the Equation 4.38. In order to achieve finite-time convergence (global finite-time stability), condition (a) can be modified to be,

\[
\dot{V} = -\alpha V^{\frac{1}{2}}, \quad \alpha > 0 \tag{4.39}
\]
Indeed, separating variables and integrating inequality in the above to get,

\[
V^{\frac{1}{2}}(t) = -\frac{1}{2} \alpha t + V^{\frac{1}{2}}(0)
\]  
(4.40)

Consequently, \(V(t)\) reaches zero in a finite time \(t_r\) that is bound by the following,

\[
t_r \leq \frac{2V^{\frac{1}{2}}(0)}{\alpha}
\]  
(4.41)

Therefore, a control \(u\) that is computed to satisfy the Equation 4.39 will drive the variable \(\sigma\) to zero in finite time and will keep it at zero thereafter. The derivative of \(V\) is computed as,

\[
\dot{V} = \sigma \dot{\sigma} = \sigma(cx_2 + u + f(x_1, x_2, t)) 
\]  
(4.42)

Assuming the control law \(u = cx_2 + v\) and substituting it into the above equation, thus obtain

\[
\dot{V} = \sigma(v + f(x_1, x_2, t)) = \sigma(v + f(x_1, x_2, t)) \leq |\sigma|L + \sigma v
\]  
(4.43)

By selecting \(v = \rho \text{sign}(\sigma)\), where

\[
\text{sign}(x) = \begin{cases} 
1, & \text{if } x > 0 \\
-1, & \text{if } x < 0 
\end{cases} \quad \& \quad \text{sign}(x) \in [-1, 1] 
\]  
(4.44)

With \(\sigma > 0\) the Equation 4.43 becomes,

\[
\dot{V} \leq |\sigma|L - \sigma \rho = -|\sigma|(\rho - L)
\]  
(4.45)

Now, the Equation 4.39 can be rewritten as,

\[
\dot{V} = -\alpha V^{\frac{1}{2}} = -\frac{\alpha}{\sqrt{2}} |\sigma|, \quad \alpha > 0
\]  
(4.46)

Combining both the above equations results in,

\[
\dot{V} \leq -|\sigma|(\rho - L) = -\frac{\alpha}{\sqrt{2}} |\sigma|, \quad \alpha > 0
\]  
(4.47)

Finally, the control gain \(\rho\) is computed as,

\[
\rho = L + \frac{\alpha}{\sqrt{2}}
\]  
(4.48)

Consequently, a control law \(u\) that drives \(\sigma\) to zero in finite time in the Equation 4.41 is derived as,

\[
u = -cx_2 - \rho \text{sign}(\sigma)
\]  
(4.49)
Now, the sliding variable can be rewritten in a form that corresponds to a straight line in the state space of the chosen system and is referred to as a sliding surface.

\[ \sigma = x_2 + cx_1 = 0, \quad c > 0 \quad (4.50) \]

Condition from the Equation 4.39 is equivalent to,

\[ \sigma \dot{\sigma} \leq -\frac{\alpha}{\sqrt{2}} \quad (4.51) \]

and is often termed the reachability condition. Meeting the reachability or existence condition in the above means that the trajectory of the system in Equation 4.33 is driven towards the sliding surface Equation 4.50 and remains on it thereafter. The control \( u = u(x_1, x_2) \) in the Equation 4.49 that drives the state variables to the sliding surface Equation 4.48 in finite time \( t_r \) and keeps them on the surface thereafter in the presence of the bounded disturbance \( f(x_1, x_2, t) \) is called a SMC.

Here the nonlinear 3DOF model from the above has been used as the foundation for the SMC’s design in order to combat the nonlinear behavior and uncertainties of the real truck with a semi-trailer. The tractor’s yaw rate, lateral velocity, and articulation angle are three control variables that need to be controlled for the reference changes. In this section, control input as front axle steering wheels are used to get an appropriate response from the truck.

The 3DOF model’s motion equations can be expressed as a matrix as,

\[ M \ddot{q} + D \dot{q} \dot{u} = U \quad (4.52) \]

where \( q \) is the control output vector, which should track the reference value, and \( U \) is the control input vector. The sliding surface that will cause the outputs to follow the desired values can be defined as,

\[ s_1 = V_y + \lambda_0 \int_0^t V_y \, dt \quad (4.53) \]

\[ s_2 = \dot{\psi}_t - \dot{\psi}_r + \lambda_1 \int_0^t (\dot{\psi}_t - \dot{\psi}_r) \, dt \quad (4.54) \]

\[ s_3 = \dot{\tau} - \dot{\tau}_r + \lambda_2 (\tau - \tau_r) + \lambda_3 \int_0^t (\tau - \tau_r) \, dt \quad (4.55) \]

The \( M \) matrix exhibits symmetry. The Sylvester theorem makes it easy to demonstrate that the matrix is positive definite. The steady-state errors are reduced by using the integral terms. The positive constants \( \lambda_0, \lambda_1, \lambda_2, \) and \( \lambda_3 \) are used. As a
result, all state variables will follow the reference values until the sliding surface selected reaches 0.

In a matrix representation, the sliding surface may be rewritten as,

\[ S = q - q_r \]  

(4.56)

where,

\[ S = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} \]

and

\[ q_r = \begin{bmatrix} -\lambda_0 \int_0^t V_y \, dt \\ \dot{\psi}_r - \lambda_1 \int_0^t (\psi_t - \dot{\psi}_r) \, dt \\ \tau_r - \lambda_2 (\tau - \tau_r) - \lambda_3 \int_0^t (\tau - \tau_r) \, dt \end{bmatrix} \]

The Lyapunov function can be added and defined as follows to assess and enhance the system’s stability,

\[ V_t = \frac{1}{2} S^T M S \]  

(4.57)

Maintaining scalar values \( s_1, s_2, \) and \( s_3 \) at zero is the control target. The above equation’s time derivative can be calculated using,

\[ \dot{V}_t = S^T M \dot{S} = S^T M (\ddot{q} - \dot{q}_r) = S^T (U - M \ddot{q} + D \dot{\psi}_t u_t) \]

(4.58)

A suitable control law can thus be described as,

\[ U = \hat{U} - K \text{sgn}(S) \]  

(4.59)

Where \( K = \text{diag}[s_1, s_2, s_3], \text{sgn}(S) = [\text{sgn}(s_1), \text{sgn}(s_1), \text{sgn}(s_1)]^T, \) here assuming the dynamics were precisely known, and \( U \) is the control input vector that would cause \( \dot{V} \) to equal zero,

\[ \hat{U} = \hat{M} \ddot{q}_r + \hat{D} \dot{\psi}_t u_t \]  

(4.60)

where \( \hat{D} \) and \( \hat{M} \) are the nominal-value vectors of \( D \) and \( M. \) Additionally, the modeling error bounds are provided as,

\[ \hat{M} \geq |\tilde{M} - M| \quad \text{and} \quad \hat{D} \geq |\tilde{D} - D| \]

(4.61)

Consequently, the Lyapunov function’s time derivative can be expressed as,
\[ \dot{V}_t \leq S^T (\dot{D} \dot{\psi}_t u_t + \dot{\tilde{M}} \dot{q}_r) - \sum_{i=1}^{3} k_i |s_i| \]  

(4.62)

As a result, the \( k_i \) components are selected as,

\[ k_i \geq |[\dot{D} \dot{\psi}_t u_t + \dot{\tilde{M}} \dot{q}_r]| + \eta_i \]  

(4.63)

When the parameters \( \eta_i \) are positive, resulting in the satisfaction of the sliding condition,

\[ \dot{V}_t \leq - \sum_{i=1}^{3} k_i |s_i| \]  

(4.64)

The surface \( S = 0 \) is reached in a finite amount of time because of the aforementioned sliding condition. Unwanted chattering could result from the control law’s discontinuity when it crosses the sliding surface \( s_i = 0 \), which is defined in equation (39). This can be avoided by substituting an approximation, such as a saturation function, for the \( \text{sgn}(.) \) function.

\[
\text{sat}(\frac{s_i}{\phi_i}) = \begin{cases} 
\frac{s_i}{\phi_i}, & \text{if } |s_i| < \phi_i \\
\text{sgn}\frac{s_i}{\phi_i}, & \text{if } |s_i| \geq \phi_i 
\end{cases} \quad \text{with } i = 1,2,3 \]  

(4.65)

Here, the thicknesses of the boundary layers are \( \phi_1, \phi_2, \) and \( \phi_3 \). The control steering angle of each wheel should be measured as the last step. Using equation (33), the association between the lateral force vector \( F_y \) and the control input vector can be expressed as

\[ U = \Lambda F_y \]  

(4.66)

The lateral force vector can be derived as follows using the invertible matrix \( L \), \( F_y = \Lambda^{-1} U \). The lateral force from the axle is therefore known. The VTM model uses the magic formula model of the tire for normal load calculations which is used to determine the front axle side-slip angle, \( \alpha_f \). The normal tire load, wheel slip ratio, and side-slide angle are model inputs for this [61]. The model’s standard form is,

\[ Y(x) = D \sin(C \arctan\{Bx - E[Bx - \arctan(Bx)]\}) \]  

(4.67)

where \( x \) is the slip ratio or the side-slip angle \( \alpha_f \), \( Y(x) \) stands for the lateral force, the longitudinal force, or the self-aligning moment and \( B,C,D,E \) represents stiffness factor, shape factor, peak value, curvature factor.

Finally, it is possible to determine the steering angle of the front axle using Equation 4.68,

\[ \delta = \arctan\left(\frac{\dot{y} + \dot{\psi}_t f}{\dot{x}}\right) + \alpha \]  

(4.68)
Chapter 4 – Lateral Control Strategies

4.6 Model Predictive Controller

Capturing system dynamics plays a key role in formulating MPC. The type of MPC, the complexity, the parameters it can handle, and several such factors are all determined by the prediction model used. The first step involved in formulating MPC lies in developing a prediction model. The targeted plant in this control system is a 4*2 tractor with a 3-axle semitrailer. The geometric model (point mass model), kinematic bicycle model, dynamic bicycle model, and full car model are four models that can be adopted as a prediction model. The full car model captures system dynamics to the maximum extent compared to other models, however, it costs excessively to solve an optimization problem. The dynamic bicycle model is a simplified version of the full car model, which still considers the vehicle dynamics such as forces and moments into account. The kinematic bicycle model simply considers only the kinematics of the vehicle such as position, velocity, and yaw angle. The geometric model represents a point mass model, and it considers only the geometric relationship between vehicle and reference, like time, position, and velocity. Both kinematic and dynamic bicycle models replicating a 4*2 rigid truck, were chosen as the two prediction models for their overall good performance at a low computation cost.

4.6.1 Kinematic bicycle model

Shown below in Figure 4.4 is an illustration of the kinematic bicycle model. In a bicycle model, the wheels to the left and right side of an axle are represented by a single wheel, here at points $F$ and $R$. The steering angles at the front wheels are represented by $\delta$. The center of gravity is about the point $G$. The distance from point $F$ and $G$ is denoted by $l_f$ and the distance from point $R$ and $G$ is denoted by $l_r$. The sum of the two distances amounts to the length of the vehicle, represented by $L$. The instantaneous center of rotation is about the point $O$. The distance from point $O$ to $F$, $G$ and $R$ are denoted by $r_F$, $r_G$ and $r_R$. The vehicle motion is in the global $X$ and $Y$ coordinates, represented by $X$ and $Y$. The orientation or yaw angle of the vehicle is represented by $\psi$. The longitudinal velocity of the vehicle at the center of gravity is denoted by $V_G$ and the angle it makes with the axis of the vehicle is called body slip angle, denoted by $\beta$. The $\angle ROG$ is equal to $\beta$ and the $\angle ROF$ is equal to $\delta$.

The major assumption made in a kinematic bicycle model is that the vehicle does not slip. In other words, the assumption is that the longitudinal velocity makes an angle of $\delta$ in the front wheel with the axis of the vehicle. In reality, the steering input will not be fully followed, hence there will be a slight decrease in the angle made by longitudinal velocity with the axis of the vehicle.

One of the main objectives of the thesis is to develop controllers for comfortable driving while achieving other objectives. Comfortable driving in terms of vehicle
control translates to a smooth transition in control inputs over a period of time. This is best achieved by modeling the change in control inputs in the state space equations rather than by introducing appropriate constraints or cost functions. Through the kinematic bicycle model, we can control the longitudinal velocity and steering angle of the vehicle. Hence, it is ideal to have \( X, \dot{X} \) (velocity), \( Y, \psi \) and \( \delta \) as the vehicle states, and \( a \) (longitudinal acceleration) and \( \Delta \delta \) as control inputs to the vehicle. From the geometric relationship that can be observed in Figure 4.4, the following equations that represent a nonlinear continuous kinematic bicycle model can be derived:

\[
\begin{align*}
\dot{X} &= V_G \cos(\beta + \psi) \quad (4.69) \\
\dot{Y} &= V_G \sin(\beta + \psi) \quad (4.70) \\
\dot{\psi} &= V_G / R = V_G / S \cos(\beta) = V_G / L / \tan(\delta) \cos(\beta) = V_G \tan(\delta) \cos(\beta) / L \quad (4.71) \\
\dot{V}_G &= a \quad (4.72) \\
\Delta \delta &= \Delta \delta \quad (4.73) \\
\beta &= \arctan(l_r / S) = \arctan(l_r / (L / \tan(\delta))) = \arctan(l_r \ast \tan(\delta) / L) \quad (4.74)
\end{align*}
\]

By discretizing the above equations for a time step of \( T_s \), we obtain the following equations:

\[
X(k + 1) = X(k) + V_G(k) \cos(\beta(k) + \psi(k)) T_s \quad (4.75)
\]
Chapter 4 – Lateral Control Strategies

\[
Y(k + 1) = Y(k) + V_G(k) \sin(\beta(k) + \psi(k))T_s \tag{4.76}
\]

\[
\psi(k + 1) = \psi(k) + (V_G(k)/l_r) \sin(\beta(k))T_s \tag{4.77}
\]

\[
V_G(k + 1) = V_G(k) + a(k)T_s \tag{4.78}
\]

\[
\delta(k + 1) = \delta(k) + \Delta\delta(k)T_s \tag{4.79}
\]

\[
\beta = \arctan(l_r \cdot \tan(\delta)/L) \tag{4.80}
\]

Writing the above equations in a matrix, we get:

\[
\begin{bmatrix}
X(k + 1) \\
V_G(k + 1) \\
Y(k + 1) \\
\psi(k + 1) \\
\delta(k + 1)
\end{bmatrix}
= \begin{bmatrix}
1 & \cos(\beta + \psi)T_s & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & \sin(\beta + \psi)T_s & 1 & 0 & 0 \\
0 & \cos(\beta)\tan(\delta)T_s/L & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
X(k) \\
V_G(k) \\
Y(k) \\
\psi(k) \\
\delta(k)
\end{bmatrix}
+ \begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
a(k) \\
\Delta\delta(k)
\end{bmatrix}
\tag{4.81}
\]

where \( \beta = \arctan(l_r \cdot \tan(\delta)/L) \)

which is of the discrete form,

\[
\mathbf{x}_{k+1} = A_d \cdot \mathbf{x}_k + B_d \cdot \mathbf{u}_k
\]

Reference tracking needs a comparison between input from reference and output from the system. In this model, all five states are taken as output and compared with the reference input, thus the state space matrix \( C \) is a diagonal matrix of size 5*5, which is of the discrete form,

\[
\mathbf{y}_k = C_d \cdot \mathbf{x}_k \tag{4.82}
\]

The state and input constraints used in the kinematic bicycle model-based MPC are as follows:

\[
\begin{bmatrix}
X_{\text{min}} \\
V_{G\text{min}} \\
Y_{\text{min}} \\
\psi_{\text{min}} \\
\delta_{\text{min}}
\end{bmatrix}
\leq
\begin{bmatrix}
X(k) \\
V_G(k) \\
Y(k) \\
\psi(k) \\
\delta(k)
\end{bmatrix}
\leq
\begin{bmatrix}
X_{\text{max}} \\
V_{G\text{max}} \\
Y_{\text{max}} \\
\psi_{\text{max}} \\
\delta_{\text{max}}
\end{bmatrix}
\tag{4.83}
\]

\[
\begin{bmatrix}
a_{\text{min}} \\
\Delta\delta_{\text{min}}
\end{bmatrix}
\leq
\begin{bmatrix}
a(k) \\
\Delta\delta(k)
\end{bmatrix}
\leq
\begin{bmatrix}
a_{\text{max}} \\
\Delta\delta_{\text{max}}
\end{bmatrix}
\tag{4.84}
\]

which are of the form,

\[
lbX \leq x \leq ubX
\]

\[
lbU \leq u \leq ubU
\]
4.6.2 Dynamic bicycle model

Assuming no slip works well at low velocity, however at higher velocities, it is crucial to consider slip. The dynamic bicycle model takes into account the dynamic forces and thus the slip at each wheel of the model. Figure 4.5 is an illustration of the dynamic bicycle model. Similar to the kinematic bicycle model, \( G, F, \) and \( R \) represent the center of gravity, front wheel, and rear wheel respectively. The steering angles at the front wheels are represented by \( \delta \). The distance from point \( F \) and \( G \) is denoted by \( l_f \) and the distance from point \( R \) and \( G \) is denoted by \( l_r \). The sum of the two distances amounts to the length of the vehicle, represented by \( L \). The vehicle motion is in the global \( X \) and \( Y \) coordinates, represented by \( X \) and \( Y \). The orientation or yaw angle of the vehicle is represented by \( \psi \). The longitudinal velocity of the vehicle at the center of gravity is denoted by \( V_G \) and the angle it makes with the axis of the vehicle is called the body slip angle, denoted by \( \beta \). The slip angle at each wheel is the difference between the steering input at each wheel and the longitudinal velocity at each wheel. \( F_{yf} \) and \( F_{yr} \) represent the lateral tire forces at the front and rear wheels respectively. Resistive forces like aerodynamic force and roll gradient force and tire forces can be modeled with coefficients and constants like \( m \) (mass of vehicle), \( C_f \) (front axle stiffness), \( C_r \) (rear axle stiffness), \( \rho \) (density of air), \( C_d \) (drag coefficient), \( A_r \) (frontal area of vehicle), \( f_r \) (rolling resistance coefficient) and \( \theta \) (gradient of road).

Figure 4.5: Dynamics of lateral vehicle motion [62].
Chapter 4 – Lateral Control Strategies

The dynamic bicycle model offers more flexibility in modeling state space equations compared to the kinematic bicycle equations. To control steering input and longitudinal velocity, $\delta$ and $a$ (longitudinal acceleration) are the control inputs. The states are $\dot{x}$ (longitudinal velocity in body frame), $\dot{y}$ (lateral velocity in body frame), $\psi$, $\dot{\psi}$, $X$ and $Y$. After solving equations of motion based on Newton’s force equilibrium conditions, we obtain the following equations:

\[
m(\ddot{x} - \dot{\psi}\dot{y}) = ma - m g f_r \cos(\theta) - \frac{1}{2} \rho C_d A_r \dot{x}^2 - F_{yf} \sin(\delta)
\]  

(4.85)

\[
m(\ddot{y} + \dot{\psi}\dot{x}) = F_{yf} \cos(\delta) + F_{yr}
\]  

(4.86)

\[
I_{zz} \ddot{\psi} = F_{yf} \cos(\delta) l_f - F_{yr} l_r
\]  

(4.87)

where $F_{yf} = C_f (\delta - \frac{\dot{y}}{\dot{x}} - \frac{\dot{\psi} l_f}{\dot{x}})$ and $F_{yr} = C_r (-\frac{\dot{y}}{\dot{x}} + \frac{\dot{\psi} l_r}{\dot{x}})$

After separating the double derivatives in the above equations to one side and writing them in a matrix form, we obtain the following:

\[
\begin{bmatrix}
\dot{x} \\
\dot{y} \\
\dot{\psi} \\
\dot{\psi} \\
\dot{X} \\
\dot{Y}
\end{bmatrix} =
\begin{bmatrix}
A_{11} & A_{12} & 0 & A_{14} & 0 & 0 \\
0 & A_{22} & 0 & A_{24} & 0 & 0 \\
0 & 0 & 0 & A_{34} & 0 & 0 \\
0 & A_{42} & 0 & A_{44} & 0 & 0 \\
A_{51} & A_{52} & 0 & 0 & 0 & 0 \\
A_{61} & A_{62} & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\dot{x} \\
\dot{y} \\
\dot{\psi} \\
\dot{\psi} \\
\dot{X} \\
\dot{Y}
\end{bmatrix} +
\begin{bmatrix}
0 \\
0 \\
\psi \\
\dot{\psi} \\
X \\
Y
\end{bmatrix} +
\begin{bmatrix}
B_{11} & B_{12} \\
B_{21} & 0 \\
0 & 0 \\
B_{41} & 0 \\
0 & 0 \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
\delta \\
a
\end{bmatrix}
\]  

(4.88)

where,

- $A_{11} = -\frac{f_r g \cos(\theta)}{x} \frac{\rho C_d A_r \dot{x}}{2m}$
- $A_{12} = \frac{C_f \sin(\delta)}{m_x}$
- $A_{14} = \frac{C_f \sin(\delta) l_f}{m x} + \dot{y}$
- $B_{11} = -\frac{C_f \sin(\delta)}{m}$
- $B_{12} = 1$
- $A_{22} = \frac{C_f \cos(\delta) + C_r}{m_x}$
- $A_{24} = \frac{C_r l_f - C_f l_f \cos(\delta)}{m x} - \dot{x}$
- $B_{21} = \frac{C_f \cos(\delta)}{m}$
- $A_{34} = 1$
- $A_{42} = -\frac{C_f \cos(\delta) l_f + C_r l_r}{I_{zz} \dot{x}}$
- $A_{44} = -\frac{C_f \cos(\delta) l_f^2 + C_r l_r^2}{I_{zz} \dot{x}}$
Chapter 4 – Lateral Control Strategies

\[ B_{41} = \frac{C_1 \cos(\delta) I_f}{I_{xx}} \]
\[ A_{51} = \cos(\psi) \]
\[ A_{52} = -\sin(\psi) \]
\[ A_{61} = \sin(\psi) \]
\[ A_{62} = \cos(\psi) \]

To control this bicycle model, it is enough to compare the following states with the reference input. Thus the chosen output from the system shall be \( \dot{x}, \psi, X, \) and \( Y. \)

\[
y = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
\begin{bmatrix}
\dot{x} \\
\dot{y} \\
\psi \\
\dot{\psi} \\
X \\
Y
\end{bmatrix} = \begin{bmatrix}
\dot{x} \\
\psi \\
X \\
Y
\end{bmatrix}
\]

which is of the continuous form,

\[ \bar{y} = C \ast \bar{x} \] (4.89)

Using the zero-order hold method, the state space matrices in continuous form are discretized.

\[ A_d = e^{Ah} \] (4.90)
\[ B_d = \int_0^h e^{As} ds \] (4.91)
\[ C_d = C \] (4.92)

\[ \bar{x}_{k+1} = A_d \ast \bar{x}_k + B_d \ast \bar{u}_k \] (4.93)
\[ \bar{y}_k = C_d \ast \bar{y}_k \] (4.94)

To achieve a smooth transition of steering angle and longitudinal acceleration input, the above state space equations need to accommodate the change in steering input (\( \Delta \delta \)) and change in longitudinal acceleration input (\( \Delta a \)). This is done by augmenting the state space matrices. Equation 4.93 can be rewritten as,

\[ \bar{x}_{k+1} = A_d \ast \bar{x}_k + B_d \ast \bar{u}_{k-1} + B_d \ast \Delta u_k \] (4.95)
\[ \bar{u}_k = \bar{u}_{k-1} + \Delta u_k \] (4.96)

Writing the above equations in matrix form, we get,

\[
\begin{bmatrix}
\bar{x}_{k+1} \\
\bar{u}_k
\end{bmatrix} = \begin{bmatrix}
A_d & B_d \\
0 & I
\end{bmatrix} \ast \begin{bmatrix}
\bar{x}_k \\
\bar{u}_{k-1}
\end{bmatrix} + \begin{bmatrix}
B_d \\
I
\end{bmatrix} \ast \Delta u_k
\]

38
\[
[y_k] = [C_d \ 0] \cdot \left[ \begin{array}{c}
\bar{x}_k \\
\bar{u}_{k-1}
\end{array} \right]
\]

which is of the augmented form,
\[
\bar{x}_{k+1} = \bar{A}\bar{x}_k + \bar{B}\Delta u_k \tag{4.97}
\]
\[
\bar{y}_k = \bar{C}\bar{x}_k \tag{4.98}
\]

Therefore, \(\bar{x}_k\), \(\Delta u_k\), \(\bar{y}_k\), \(\bar{A}\), \(\bar{B}\) and \(\bar{C}\) represents the states, control input, output and state space matrices of the dynamic bicycle model. The state and input constraints used in the dynamic bicycle model-based MPC are as follows:
\[
\begin{bmatrix}
\dot{x}_{\min} \\
\dot{y}_{\min} \\
\psi_{\min} \\
\psi_{\min} \\
X_{\min} \\
Y_{\min} \\
\delta_{\min} \\
a_{\min}
\end{bmatrix}
\leq
\begin{bmatrix}
\dot{x}(k) \\
\dot{y}(k) \\
\psi(k) \\
\psi(k) \\
X(k) \\
Y(k) \\
\delta(k) \\
a(k)
\end{bmatrix}
\leq
\begin{bmatrix}
\dot{x}_{\max} \\
\dot{y}_{\max} \\
\psi_{\max} \\
\psi_{\max} \\
X_{\max} \\
Y_{\max} \\
\delta_{\max} \\
a_{\max}
\end{bmatrix}
\tag{4.99}
\]
\[
\begin{bmatrix}
\Delta\delta_{\min} \\
\Delta a_{\min}
\end{bmatrix}
\leq
\begin{bmatrix}
\Delta\delta(k) \\
\Delta a(k)
\end{bmatrix}
\leq
\begin{bmatrix}
\Delta\delta_{\max} \\
\Delta a_{\max}
\end{bmatrix}
\tag{4.100}
\]

which are of the form,
\[
lbX \leq x \leq ubX
\]
\[
lbU \leq u \leq ubU
\]

### 4.6.3 Solver selection

In order to solve the optimization problem, there are several solvers available to choose from. With the aim of formulating an LTV (linear time-varying) and LPV (linear parameter varying) MPC, a quadratic cost function subjected to constraints needs to be solved\[63\]. For their ease of use, robustness, and quickness, the following three solvers were narrowed down to choose from.

**CasADi**

CasADi serves as an open-source instrument for nonlinear optimization and algorithmic differentiation. It expedites the swift and effective deployment of diverse approaches for achieving optimal numerical control. This capability applies in scenarios encompassing offline contexts as well as linear and nonlinear model predictive control. The integration of solvers such as qpOASES and OSQP is also attainable within the framework of CasADi. CasADi uses algorithmic differentiation, unlike most frameworks. Algorithmic differentiation is faster and more accurate than numerical and symbolic differentiation.
qpOASES

qpOASES is a structure-exploiting active-set QP solver. Usually, qpOASES expects QPs to be formulated in the following standard form with a positive (semi-)definite Hessian matrix $H$ and lower and upper bounds $lb$ and $ub$:

$$
\min_u \frac{1}{2} u^T H u + u^T g(x) \quad (4.101)
$$

$$
lb A(x) \leq Au \leq ub A(x)
$$

$$
lb(x) \leq u \leq ub(x)
$$

Although the online active set strategy was originally designed for QP sequences with fixed Hessian and constraint matrices, it can be extended to the case where these matrices vary from one QP to the next.

OSQP

The OSQP (Operator Splitting Quadratic Program) solver is a numerical optimization package for solving convex quadratic programs in the form:

$$
\min_x \frac{1}{2} x^T P x + x^T q(x) \quad (4.102)
$$

$$
lb A \leq Ax \leq ub A
$$

where $x$ is the optimization variable and $P$ is a positive semidefinite matrix. It is similar to qpOASES but the constraints are formulated a bit differently.

Both qpOASES and OSQP can be applied to the bicycle model scenario. Though the system considered in this work is relatively simple, the overall performance and robustness of OSQP outperform qpOASES. The decision was made to initially adopt the OSQP solver for the MPC problem. The integration of the CasADi framework (alongside the aforementioned solvers) will only be pursued if the solver’s performance is deemed sluggish. Additionally, this step might be considered if the formulation of the MPC optimization problem, which evolves based on the selected state variables, cannot be expressed within the solver’s equation framework.

4.6.4 MPC formulation

As discussed earlier, MPC solves a cost function or optimization problem using a prediction model that may have certain state and input constraints to provide an optimal control sequence. The objective of MPC in this case is reference tracking provided safety and driving comfort are ensured. The cost function is of the
Chapter 4 – Lateral Control Strategies

form:

\[ J = \min \left[ \frac{1}{2} \tilde{e}_N^T S \tilde{e}_N + \frac{1}{2} \sum_{i=0}^{N-1} \left[ \tilde{e}_{k+i}^T Q \tilde{e}_{k+i} + \Delta \tilde{u}_{k+i}^T R \Delta \tilde{u}_{k+i} \right] \right] \quad (4.103) \]

where,

- \( J \) is the cost function
- \( N \) is the prediction horizon
- \( \tilde{e} \) is the difference between reference and output called error
- \( Q \) is the weight matrix on error
- \( R \) is the weight matrix on control input
- \( S \) is the weight matrix on an error in the terminal set

The cost function aims at reducing the error between states and references and reducing the magnitude of control inputs. The state constraints and input constraints represent the physical, legal, and comfort constraints of the vehicle. The constraints create supersets \( X, X_N, \) and \( U \) within which the states and control inputs respectively are allowed to vary. The constraints are of the form:

\[
\begin{align*}
\text{The optimization problem can be written as,} \\
\min_{\tilde{e}, \tilde{u}, \Delta \tilde{u}} \left[ \tilde{e}_N^T S \tilde{e}_N + \frac{1}{2} \sum_{i=0}^{N-1} \left[ \tilde{e}_{k+i}^T Q \tilde{e}_{k+i} + \Delta \tilde{u}_{k+i}^T R \Delta \tilde{u}_{k+i} \right] \right] \\
\text{s.t.} \quad \tilde{x}_{k+1} = \tilde{A} \tilde{x}_k + \tilde{B} \Delta \tilde{u}_k \\
\tilde{x}_k \in X, \quad k = 0, 1, \ldots, N-1 \\
\tilde{u}_k \in U, \quad k = 0, 1, \ldots, N-1 \\
\tilde{x}_N \in X_N, \quad k = 0, 1, \ldots, N-1
\end{align*}
\]

(4.104)
Chapter 5

Implementation and Integration

The designed control law formulations are implemented in the MATLAB-Simulink environment. The reference trajectory is loaded from the pre-defined track files in a MATLAB script. The Volvo Transport Model (VTM) was used as the plant model, and it runs on Simulink. It is a complex nonlinear model of trucks, capturing a vast amount of real truck dynamics. Given the reference track file input as discrete points of $X$ and $Y$ coordinates of the path in the global coordinate system, the pure pursuit block in MATLAB-Simulink has a function that generates the heading angle with respect to the target look ahead point. The control variable, steering wheel angle has been formulated using mathematical function blocks to generate appropriate responses based on the controller design. Later, the control strategy of PPC was integrated with the collision avoidance system by placing an obstacle on the predefined path. The Stanley model consists of kinematic and dynamic blocks. The blocks have inputs referencing the trajectory and current position of the truck along with the velocity and direction of the truck. This determines the closest point to the vehicle on the reference trajectory. Additional input ports are enabled by changing the Dynamic bicycle model as a vehicle model parameter. The curvature of the path at the reference point can be represented as a scalar in the Curvature port. The current yaw rate in degrees per second is inputted as a scalar into the CurrYawRate port. The current steering angle in degrees is represented as a scalar that can be passed to the CurrSteer port. This information is then used to calculate cross-track error $e$ and heading error $\psi$ and then the front steering wheel angle command is generated. Visualization of the truck behavior can be seen in the VTM2VR tool which was developed by Volvo for internal verification purposes in Figure 5.1. The basic formulation of the Stanley controller was made more robust and tuned for better response in terms of the cross-track, heading error.
metrics, oscillatory behavior of the truck, and driver comfort. Tuning and robustness were directed towards the simulation of the VTM vehicle model in Simulink along with the VTM2VR visualization interface. $k_\psi$ and $k$ from the Equation ?? are tuned based on simulation results, to improve robustness, level of driver comfort, and tracking performance. Whereas in LQR, given state variables as the input reference and feedback of the vehicle states, the control variable, steering wheel angle has been calculated using mathematical function blocks to generate an appropriate steering response based on the controller design. Later, the control strategy of LQR was integrated with the collision avoidance system by placing an obstacle on the predefined path.

In SMC design, chattering refers to the rapid switching between control actions near the sliding surface. It can cause high-frequency oscillations in the system, leading to undesirable effects such as increased wear and tear on actuators and disturbances in the controlled variable. Here a few techniques are adopted to reduce the chattering effect in SMC design. The super-twisting algorithm (STA) is designed to govern systems with a relative degree of 1 and to maintain robust stability while lowering chattering. By adding an analogous command $\delta_{eq}$ that is generated by solving the $\dot{s} = 0$ equation in order to avoid significant peaks in transient phases. This serves as a feedforward to the system’s approach to the sliding surface, and it is provided by,

$$\delta_{eq} = \frac{m}{C_f} \phi(t, s) \quad \text{and} \quad \phi(t, s) = -\frac{C_f + C_r}{m V_x} V_y - \frac{L_f C_f - L_r C_r}{m V_x} \psi + \lambda_0 V_y \quad (5.1)$$

As a result, the steering angle used to describe the system’s control input is defined.
as follows,

$$\delta_f = \delta + \delta_{eq} \quad (5.2)$$

By introducing a boundary layer around the sliding surface where the control action is gradually switched rather than abruptly. Thus, widening the boundary layer results in reducing the frequency and magnitude of chattering. This has been achieved by modifying the sliding surface and replacing the saturation function with a hyperbolic tangent function to smooth out the control signal. Also, applied a low-pass filter to the control signal to attenuate high-frequency components. This filtering technique helped in reducing chattering without significantly affecting the overall system response.

MPC involves solving an optimization problem and as discussed in the previous chapter, OSQP is the chosen solver. For the ease of testing multiple solvers, a parser is required. A parser accepts a standard form of MPC formulation. Depending on the solver chosen in the parser interface, the written MPC formulation is automatically converted to the format required by each solver. ACADOS and YALMIP are two such software packages. ACADOS is relatively a recent software, which has solvers like qpOASES, OSQP, HPIPM, qpDUNES, etc. YALMIP has been available for free for several years now and it can work with tens of solvers ranging from GUROBI, MOSEK, OSQP, etc. for several types of optimization problems. Both ACADOS and YALMIP were given a fair amount of try during implementation. Though ACADOS can be implemented for this work, it did not prove to be robust enough for this application. ACADOS is targeted for systems that require quick calculations, however, the problem formulation needs to be very precise. Due to this problem, disturbances or mismatches in initializing or several other reasons could easily run into infeasible solution problems. Another disadvantage with ACADOS implementation is, that when the plant model is a Simulink model, it is necessary to create S-functions. YALMIP on the other hand is much easier to use, can be run on MATLAB, and can be called from Simulink through a MATLAB function. YALMIP with its variety of solvers can be helpful in developing maintainable code for future work. For such reasons, YALMIP was chosen as the parser and OSQP was chosen as the solver. Other solvers can be easily tried by renaming the solver name in the YALMIP interface.

By initializing the required constants and feeding reference track information in the MATLAB workspace, the YALMIP solver is called through a MATLAB function via Simulink. YALMIP uses semidefinite programming variables (SDP variables) to calculate cost functions and run predictions. The best practice to use YALMIP is by initializing the required SDP variables once and running the test case in a loop until the end. Since VTM is not available on MATLAB, YALMIP had to be called as a function. By calling it through a function, YALMIP is forced to create, run, and destroy its SDP variables for every timestep. Though this is an inefficient method of testing, there is no other choice to run with VTM. It proved to be a liability
while testing with the dynamic bicycle model as the prediction model, where the
simulation took several hours to run. Due to this problem, once the performance of
the kinematic and dynamic bicycle model with VTM is checked, the integration of
the collision avoidance system with MPC was done in MATLAB without the VTM
model as the plant.
Chapter 6

Simulation Results

6.1 Introduction to scenarios

This section compares the different path-tracking controllers for performance under various test scenarios. To achieve this, the most popularly recommended configuration of the controller is employed, and the performance of the controllers is assessed for three distinct driving scenarios, as described below.

- Low-speed path following on a predefined track without CAS system at 25 km/h.
- Low-speed path following on a customized track along with the CAS system at 35 km/h.
- High-speed path following on a customized track along with the CAS system at 90 km/h.

To begin with, the designed controllers were tested on the truck with the standard curvature path at a speed of 25 km/h by integrating with the Volvo transport model (VTM) as a plant model in Simulink. In this, the truck replicates precise path tracking and thus behavior represents the real-world case.

6.2 Description of evaluation

The control methods chosen for the simulation are listed in it, along with the inspirations from which they were selected. The following criteria are used to assess the controller’s performance.

Maximum lateral cross-track error:

\[
\epsilon_{d,\text{max}} = \max_{t \in [0,T]} |\epsilon_d(t)|
\]  

(6.1)
Chapter 6 – Simulation Results

Maximum Heading error:

\[ \epsilon_{\theta,\text{max}} = \max_{t \in [0,T]} |\epsilon_{\theta}(t)| \]  

(6.2)

Also the average root mean square (RMS) error has been calculated as,

Average lateral cross-track error:

\[ \epsilon_{d,\text{rms}} = \sqrt{\frac{1}{T} \int_0^T (\epsilon_d(t))^2 \, dt} \]  

(6.3)

Similarly, the average heading error is given as,

\[ \epsilon_{\theta,\text{rms}} = \sqrt{\frac{1}{T} \int_0^T (\epsilon_{\theta}(t))^2 \, dt} \]  

(6.4)

The tuning parameters that minimize the RMS and maximum error are optimized in order to increase the accuracy and performance of the controller. For all simulation results, the forward speed of the truck and the track references vary for the situations at a given speed and were displayed under ideal circumstances which are initially free from outside disturbances. The RMS values of the heading and lateral errors were then reported. The simulation results demonstrate similar properties for the developed controllers that can be found in the literature.

6.3 Path tracking results of controllers without CAS

The reference trajectory and the truck’s actual path tracking of PPC can be seen in Figure 6.1.
Figure 6.1: PPC tracking performance at 25 km/h.

Figure 6.2: PPC performance tracking metrics at 25 km/h.

Similarly, the Stanley controller has been tested in short curved roads at 25 km/h. The reference trajectory and the truck’s actual path tracking can be seen in Figure 6.3.
To quantify and compare the performance of the geometry-based PPC and Stanley controllers, lateral (cross-track) error $e$, relative heading error $\psi$, and steering angles are plotted in Figure 6.2 and in Figure 6.4. It is evident that both are following the
Chapter 6 – Simulation Results

reference trajectory in a quite satisfactory way. Thus, the tracking performance of them has been simulated and verified. Following, the designed LQR controller path tracking can be seen in Figure 6.5, where the reference trajectory along with the truck and the semi-trailer body is depicted.

![LQR Path Tracking](image)

(a) LQR Path tracking of a Tractor Body  
(b) LQR Path tracking of a Semitrailer Body

Figure 6.5: LQR tracking performance at 25 km/h.

Furthermore, the designed SMC controller has been tested on the previously discussed case scenario. In this, the truck replicates precise tracking of the path and thus behavior represents the real-world case. The reference trajectory, the truck, and the semi-trailer’s actual path tracking can be seen in Figure 6.7.
(a) SMC Path tracking of a Tractor Body.  
(b) SMC Path tracking of a Semitrailer Body.

Figure 6.7: SMC tracking performance at 25 km/h.

To quantify and compare the performance of the model-based controllers, lateral error $e$, heading error $\psi$, and steering angles are plotted in Figure 6.6 and in Figure 6.8. It is evident that both the controllers are following the reference trajectory in an effective way. Thus, the tracking performance of them has been simulated and verified.

The results show that the truck behaves in a very stable and controlled manner to the designed controllers, with an over-damped output and minimal cross-track errors. There are no unnecessary oscillations and the maximum deviation from the
trajectory is around 0.11 m. As the VTM plant represents a real-life truck model that has more accurate vehicle properties and behavior in real-world situations that replicates in the Simulink interface. The VTM plant model accounts for slip, non-linearities, and dynamic properties. As a result, the lateral error is still way better than what can be achieved even by a professional human driver. Overall, the controllers’ performance metrics evaluation in the short curved road at 25 km/h can be seen in Table 6.1.

Table 6.1: Controllers performance metrics evaluation in short curved road at 25 km/h.

<table>
<thead>
<tr>
<th>Controller</th>
<th>Max. Lateral Error (m)</th>
<th>RMS Lateral Error (m)</th>
<th>Max. Heading Error (deg)</th>
<th>RMS Heading Error (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPC</td>
<td>0.1</td>
<td>0.08</td>
<td>0.6</td>
<td>0.47</td>
</tr>
<tr>
<td>Stanley</td>
<td>0.11</td>
<td>0.10</td>
<td>0.8</td>
<td>0.76</td>
</tr>
<tr>
<td>LQR</td>
<td>0.12</td>
<td>0.10</td>
<td>1.1</td>
<td>1.05</td>
</tr>
<tr>
<td>SMC</td>
<td>0.09</td>
<td>0.73</td>
<td>0.85</td>
<td>0.81</td>
</tr>
</tbody>
</table>

According to the data observed, the developed controllers operated admirably under optimal driving conditions. The effectiveness of the controllers can also be assessed when considering the impact of external factors like disturbances, such as introducing random noise to the truck’s output. Similar results for MPC will be discussed in the later half of this chapter.

6.4 Path tracking results of controllers with CAS

Furthermore, the tuned geometrical and model-based controllers by the side of the longitudinal controller are integrated with VTM along with the CAS with a static obstacle to run on the standard tracks at a lower speed of 35 km/h and at a higher speed of 90 km/h. The custom track from a predefined track file represents dynamic curvature with right and left turns while avoiding collision with the obstacle on the path at given speeds.

6.4.1 Path tracking results of controllers with CAS at 35 km/h

For PPC, the reference trajectory and the truck’s actual path tracking can be seen in Figure 6.9. Corresponding lateral deviation and heading deviation from the base frame along with the steering angles are plotted in Figure 6.10. However, as evident from the plot, the controller results in quite fewer oscillations than usual in steering angle commands. Physically, this would result in uncomfortable rides for passengers in the vehicle. This is one of the drawbacks of the PPC. This is expected as the
controller, by design, is kinematic, and does not account for vehicle dynamics and any comfort reduction.

Figure 6.9: Pure Pursuit lateral control of a truck with collision avoidance system at 35 km/h.

Figure 6.10: Performance evaluation of a Pure Pursuit lateral control for a truck with collision avoidance system at 35 km/h.
Chapter 6 – Simulation Results

As seen from Fig. 6.9 and Fig. 6.10, the maximum lateral deviation is 1.95 m with a maximum heading deviation of 0.45 deg and a collision safe distance of 0.45 m. Similarly, the tuned Stanley controller along with the CAS has been simulated, where the reference trajectory with the truck path tracking along with the collision avoidance can be seen in Figure 6.11. Corresponding lateral deviation, heading deviation from the base frame, and steering angles are plotted in Figure 6.12. It is evident from the plot, that the Stanley controller also results in oscillations in steering angle commands as compared to the PPC. This is expected as the controller, by design, is kinematic, and does not account for vehicle dynamics and any comfort reduction. As seen from Figure 6.11 and Figure 6.12, the maximum lateral deviation is 2.41 m with a maximum heading deviation of 3.2 deg and a collision safe distance of 1.79 m.

Figure 6.11: Stanley lateral control of a truck with collision avoidance system at 35 km/h.
Chapter 6 – Simulation Results

Following, the LQR controller has been tested with the CAS at a speed of 35 km/h. The reference trajectory and the truck’s actual path tracking can be seen in Figure 6.13. Corresponding lateral deviation, heading deviation from the base frame, and steering angles are plotted in Figure 6.14. From the plot, the controller results in slower response in steering angle commands, but lesser oscillations compared to the geometrical controllers. Physically, this would result in a comfortable ride for passengers in the vehicle but takes a longer time to respond because of the system minimization issue. This is one of the drawbacks of the LQR controller. This is expected as the controller, by design linearise the non-linear truck model, thus increasing uncertainty and also not robust in the presence of uncertainty.
Chapter 6 – Simulation Results

Figure 6.13: LQR lateral control of a truck with collision avoidance system at 35 km/h.

Figure 6.14: Performance evaluation of a LQR lateral control for a truck with collision avoidance system at 35 km/h.
Chapter 6 – Simulation Results

As seen from Figure 6.13 and Figure 6.14, the maximum cross-track error is 3.31 m with a maximum heading deviation of 0.5 deg and a collision safe distance of 2.74 m. Furthermore, the SMC controller has been tested custom track from predefined track files to test the CAS along with the safe positioning and comfort of the driver at a lower speed of 35 km/h and at a higher speed of 90 km/h.

Figure 6.15: SMC lateral control of a truck with collision avoidance system at 35 km/h.
As seen from the Table 6.8, the maximum lateral deviation is 3.75 m with maximum heading deviation of 1.1 deg and a collision safe distance of 3.01 m.

The reference trajectory and the truck’s actual path tracking can be seen in Figure 6.15. Corresponding lateral deviation, heading deviation, and steering angles are plotted in Figure 6.16. Initially, the controller resulted in oscillations in steering angle commands. This is one of the drawbacks of the SMC, as it resulted in a chattering effect and has a tendency to excite high-frequency oscillations due to unmodelled dynamics. However, as evident from the plot this chattering effect has been minimized with the implementation of STA with SMC. Overall, the controller’s performance metrics evaluation in customized curved road at a high speed of 35 km/h with CAS can be seen in Table 6.2. It displays the performance assessment of the designed controllers at the chosen driving scenario. For all simulation results, the forward speed of the truck and the track references vary for the situations at low and high speeds. The RMS values of the heading and lateral deviations were then reported. The simulation results demonstrate similar properties for the developed controllers that can be found in the literature review. According to the data observed, the developed controllers operated admirably under optimal driving conditions. In these circumstances, lateral and heading errors decrease as vehicle speed increases, which is consistent with earlier published findings.
Table 6.2: Overall controller performance metrics evaluation in a curved road with CAS at 35 km/h with a static obstacle.

<table>
<thead>
<tr>
<th>Controller with CAS at 35km/h</th>
<th>Max. Lateral Deviation (m)</th>
<th>RMS Lateral Deviation (m)</th>
<th>Max. Heading Deviation (deg)</th>
<th>RMS Heading Deviation (deg)</th>
<th>CA Safe Distance (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPC</td>
<td>1.95</td>
<td>2.08</td>
<td>0.45</td>
<td>0.78</td>
<td>0.45</td>
</tr>
<tr>
<td>Stanley</td>
<td>2.41</td>
<td>2.35</td>
<td>3.2</td>
<td>3.11</td>
<td>1.79</td>
</tr>
<tr>
<td>LQR</td>
<td>3.31</td>
<td>1.52</td>
<td>0.5</td>
<td>0.44</td>
<td>2.74</td>
</tr>
<tr>
<td>SMC</td>
<td>3.75</td>
<td>1.05</td>
<td>1.1</td>
<td>1.06</td>
<td>3.01</td>
</tr>
</tbody>
</table>

6.4.2 Path tracking results of controllers with CAS at 90 km/h

Followed by, the PPC has been tested on a higher-speed custom track from a pre-defined track file to test the collision avoidance system along with the safe positioning and comfort of the driver. As seen in Figure 6.17, the PPC performs decently with acceptable minimum heading and cross-track error.

![Path tracking of a Heavy Truck with Pure Pursuit Controller](image)

Figure 6.17: Pure Pursuit lateral control of a truck with collision avoidance system at 90 km/h.
Chapter 6 – Simulation Results

Figure 6.18: Performance evaluation of a Pure Pursuit lateral control for a truck with collision avoidance system at 90 km/h.

As seen from Figure 6.17 and Figure 6.18, the maximum lateral deviation is 1.53 m with a maximum heading deviation of 1.05 deg and a collision safe distance of 1.05 m. Here, Collision Avoidance safe distance is defined as the distance between the outermost bodies of the truck and the obstacle when they are in adjacent to each other on the road. Next, as seen in Figure 6.17, the Stanley controller results in minimal heading and lateral deviation which affects the safe distance while crossing the obstacle.
Figure 6.19: Stanley lateral control of a truck with collision avoidance system at 90 km/h.

Figure 6.20: Performance evaluation of a Stanley lateral control for a truck with collision avoidance system at 90 km/h.
As seen from Figure 6.19 and Figure 6.20, the maximum lateral deviation is 2.36 m with a maximum heading deviation of 0.12 deg and a collision safe distance of 2.19 m. Succeeding, as seen in Figure 6.21, the LQR controller performs well and gives decent collision avoidance and comfort driving.

Figure 6.21: LQR lateral control of a truck with collision avoidance system at 90 km/h.
Figure 6.22: Performance evaluation of a LQR lateral control for a truck with collision avoidance system at 90 km/h.

As seen from Figure 6.21 and Figure 6.22, the maximum lateral deviation is 3.62 m with a maximum heading deviation of 1.3 deg and a collision-safe distance of 2.52 m. Lastly, as seen in Figure 6.23, the SMC Controller performs well with minimum heading deviation and maximum lateral deviation.
Chapter 6 – Simulation Results

Figure 6.23: SMC lateral control of a truck with collision avoidance system at 90 km/h.

Figure 6.24: Performance evaluation of a Sliding Mode lateral control for a truck with collision avoidance system at 90 km/h.
As seen from Figure 6.23 and Figure 6.24, the maximum lateral deviation is 5.10 m with a maximum heading deviation of 0.23 deg and a collision safe distance of 3.14 m.

Overall, the controllers’ performance metrics evaluation in customized curved roads at a high speed of 90 km/h with CAS can be seen in Table 6.3.

Table 6.3: Overall controllers performance metrics evaluation in a curved road with CAS at 90 km/h with a static obstacle.

<table>
<thead>
<tr>
<th>Controller with CAS at 90km/h</th>
<th>Max. Lateral Deviation (m)</th>
<th>RMS Lateral Deviation (m)</th>
<th>Max. Heading Deviation (deg)</th>
<th>RMS Heading Deviation (deg)</th>
<th>CA Safe Distance (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPC</td>
<td>1.53</td>
<td>1.48</td>
<td>1.05</td>
<td>1.01</td>
<td>1.05</td>
</tr>
<tr>
<td>Stanley</td>
<td>2.36</td>
<td>2.24</td>
<td>0.12</td>
<td>0.1</td>
<td>2.19</td>
</tr>
<tr>
<td>LQR</td>
<td>3.62</td>
<td>1.37</td>
<td>0.5</td>
<td>0.44</td>
<td>2.74</td>
</tr>
<tr>
<td>SMC</td>
<td>5.10</td>
<td>1.12</td>
<td>0.23</td>
<td>0.21</td>
<td>3.14</td>
</tr>
</tbody>
</table>

Table 6.3 displays the performance assessment of the designed controllers using error metrics for the high-speed driving scenario. For all simulation results, the forward speed of the truck and the track references vary for the situations at low and high speeds. The RMS values of the heading and lateral errors were then reported. The simulation results demonstrate similar properties for the developed controller that can be found in the literature. According to the data observed, the developed controller operated admirably under optimal driving conditions. In these circumstances, lateral and heading errors decrease as vehicle speed increases, which is consistent with earlier published findings.
6.5 Path tracking results of Model Predictive Control

In MPC, first, the effectiveness and difference in performance of both kinematic bicycle model and dynamic bicycle model for path tracking application with VTM has to be checked. For this, MPC with only path tracking capability was integrated with the VTM model in Simulink. After tuning for a short curved track at a low speed of 25\(km/h\), the results obtained for MPC with kinematic and dynamic bicycle models, (MPC-Kin and MPC-Dyn) integrated with VTM are as shown in Figures 6.25, 6.26, 6.27 and Figures 6.28, 6.29, 6.30 respectively.

![Path tracking of a Heavy Truck with MPC - Kinematic](image)

Figure 6.25: Path tracking by MPC-Kinematic integrated with VTM.
Figure 6.26: Error in path tracking by MPC-Kinematic integrated with VTM.

Figure 6.27: Input by MPC-Kinematic integrated with VTM while path tracking.
Figure 6.28: Path tracking by MPC-Dynamic integrated with VTM.
Figure 6.29: Error in path tracking by MPC-Dynamic integrated with VTM.

Figure 6.30: Input by MPC-Dynamic integrated with VTM while path tracking.
The basic comprehension from these results is that the dynamic bicycle model is more capable than the kinematic bicycle model in achieving comfortable driving and accurate path tracking, however, it is computationally expensive. Moving forward with this understanding, for the ease of tuning and integration with the collision avoidance system, the plant which is the VTM is replaced by the kinematic and dynamic bicycle models, (KBM and DBM) respectively. After tuning for a short curved track at a low speed of 25 km/h, the results obtained for MPC with kinematic and dynamic bicycle models integrated with kinematic and dynamic bicycle plant models, respectively are as shown in Figures 6.31, 6.32, 6.33 and Figures 6.34, 6.35, 6.36 respectively.

![Path tracking with MPC - kinematic model](image)

Figure 6.31: Path tracking by MPC-Kinematic with kinematic bicycle model as plant.
Chapter 6 – Simulation Results

Figure 6.32: Error in path tracking by MPC-Kinematic with kinematic bicycle model as the plant.

Figure 6.33: Input by MPC-Kinematic while path tracking with kinematic bicycle model as the plant.
Figure 6.34: Path tracking by MPC-Dynamic with dynamic bicycle model as the plant.
Chapter 6 – Simulation Results

Figure 6.35: Error in path tracking by MPC-Dynamic with dynamic bicycle model as the plant.

Figure 6.36: Input by MPC-Dynamic while path tracking with dynamic bicycle model as the plant.
6.6 Collision avoidance results of Model Predictive Control

Now the MPC with kinematic and dynamic prediction models integrated with kinematic and dynamic plant models are ready to be integrated with the collision avoidance system. After tuning for a short curved track at a low speed of 25km/h with a static obstacle on the reference track at 30m from entry, the results obtained for MPC with kinematic and dynamic bicycle models integrated with kinematic and dynamic bicycle plant models integrated with collision avoidance system, respectively are as shown in Figures 6.37, 6.38, 6.39 and Figures 6.40, 6.41, 6.42 respectively.

Figure 6.37: Path tracking by MPC-Kinematic integrated with CAS with kinematic bicycle model as the plant.
Figure 6.38: Deviation in path tracking by MPC-Kinematic integrated with CAS with kinematic bicycle model as the plant.

Figure 6.39: Input by MPC-Kinematic integrated with CAS while path tracking with kinematic bicycle model as the plant.
Figure 6.40: Path tracking by MPC-Dynamic integrated with CAS with dynamic bicycle model as the plant.
Figure 6.41: Deviation in path tracking by MPC-Dynamic integrated with CAS with dynamic bicycle model as the plant.

Figure 6.42: Input by MPC-Dynamic integrated with CAS while path tracking with dynamic bicycle model as the plant.
Table 6.4 presents the maximum lateral error, RMS lateral error, maximum heading error, and RMS heading error for the controllers based on MPC on a short curved road at low speed. As expected, the kinematic bicycle model performs poorly as a prediction model compared to the dynamic bicycle model when the VTM is used as the plant. The plant used here is a 4*2 tractor with a 3-axe semitrailer, whereas the kinematic and dynamic bicycle model can represent only a 4*2 tractor. Comparing their RMS error values, the MPC-Dyn performs almost twice as well than MPC-Kin with VTM. The results for MPC-Kin with KBM and MPC-Dyn with DBM contradict the above-made inference, it is necessary to look at Figures 6.31 and 6.34 for a better understanding. MPC-Dyn with DBM performs poorly only in the end, unable to converge. This is the reason for the unexpected higher maximum and RMS error values.

Table 6.4: MPC Controllers performance metrics evaluation in short curved road at 25\( \text{km/h} \).

<table>
<thead>
<tr>
<th>Controller</th>
<th>Max. Lateral Error (m)</th>
<th>RMS Lateral Error (m)</th>
<th>Max. Heading Error (deg)</th>
<th>RMS Heading Error (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPC-Kin with VTM</td>
<td>2.61</td>
<td>1.52</td>
<td>5.66</td>
<td>3.04</td>
</tr>
<tr>
<td>MPC-Dyn with VTM</td>
<td>1.18</td>
<td>0.63</td>
<td>4.35</td>
<td>1.65</td>
</tr>
<tr>
<td>MPC-Kin with KBM</td>
<td>0.67</td>
<td>0.33</td>
<td>5.79</td>
<td>2.45</td>
</tr>
<tr>
<td>MPC-Dyn with DBM</td>
<td>1.99</td>
<td>0.41</td>
<td>6.64</td>
<td>1.97</td>
</tr>
</tbody>
</table>

By feeding a vector of reference input for every time step which is generated by the collision avoidance system, the MPC is integrated with CAS. Unlike reactive controllers, once integrated with CAS, MPC seemed to become sensitive to tuning. Generally, MPC is tuned differently for different maneuvers. When CAS provides different curves of different curvatures for every timestep in one specific stretch of maneuver, it did not prove to be a very useful solution for collision avoidance unlike with other controllers. MPC-Kin was capable of running without falling into an infeasible solution at a sampling time of 0.1s whereas MPC-Dyn needed a sampling time as low as of 0.02s. The smaller the sampling time, the more and more different curves are to be generated and the more sensitive MPC gets with tuning. Thus, beyond a certain point, it was not possible to effectively tune MPC with CAS to avoid obstacles. With this understanding, as expected, MPC-Kin performed better with CAS than MPC-Dyn with CAS due to its higher sampling time. The results presented in Table 6.5 show that MPC-Kin with CAS was able to avoid the obstacle...
successfully with a safe distance of 0.60m and MPC-Dyn with CAS scrapes through the static obstacle by 0.21m.

Table 6.5: MPC Controllers with CAS performance metrics evaluation in the short curved road at 25 km/h with a static obstacle at 30m from entry.

<table>
<thead>
<tr>
<th>Controller</th>
<th>Max. Lateral Deviation (m)</th>
<th>RMS Lateral Deviation (m)</th>
<th>Max. Heading Deviation (deg)</th>
<th>RMS Heading Deviation (deg)</th>
<th>CA Safe Distance (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPC-Kin with KBM and CAS</td>
<td>3.75</td>
<td>1.77</td>
<td>7.84</td>
<td>3.72</td>
<td>0.60</td>
</tr>
<tr>
<td>MPC-Dyn with DBM and CAS</td>
<td>3.74</td>
<td>1.76</td>
<td>10.20</td>
<td>3.01</td>
<td>-0.21</td>
</tr>
</tbody>
</table>
Chapter 7

Inferences and Discussion

From the initial observations, noticed that the vehicle is tracking better and the path-tracking quality in the PPC is significantly impacted by the look-ahead distance $l_d$. A larger look-ahead distance allows the vehicle to smoothly converge to the target path, but it takes longer and travels over a greater distance. A smaller forward distance swiftly converges the vehicle to the desired path, but it also reduces the stability of the path tracking. This experience has taught us that a slight look-ahead distance should be used when the vehicle’s deviation is large in order to quickly adjust the vehicle’s position and converge to the chosen trajectory, a large look-ahead distance should be used when the vehicle’s deviation is small in order to prevent oscillation due to an overly sensitive system adjustment. Also, to minimize the cutting-the-corner effects, implemented the variable look ahead method which depends on the velocity of the truck. In addition from interpreted results, the path deviates non-linearly from linear as the look-ahead distance grows. It is also clear from the review study and simulation results that, even though these geometric controllers operate pretty well at lower velocities, they might not be suitable for highway driving.

As far as heavy vehicles are concerned, the main interest lies in reducing lateral error distance and directional errors as compared to the reference trajectory. Further, having the ability to track a given reference trajectory at a higher speed indicates a better controller as long as the ride is not uncomfortable. It is intuitive that these objectives can be achieved in a more convincing and easier way if four-wheel steering is used instead of just front-wheel steering. Overall, the work done in this thesis is best suitable for comparing lateral control schemes in front steering vehicles, as 4-wheel steering that includes both front and rear wheels is quite complicated due to the heavy powertrain and multiple axle design in heavy truck applications. Hence, only the front-wheel steering controller strategies were developed in the thesis for the lateral control of the heavy truck. The geometrical strategy of controller design does
not consider the vehicle dynamics into account which leads to the unsafe passage of the truck while avoiding the collision object on the path. Validating the controller on a real truck on the road is the next step, which is what Volvo GTT aims to work on. This implementation and validation, however, is beyond the scope of this project.

In comparison to implemented geometrical controllers with and without collision avoidance, the optimization-based controller LQR offered relatively lower lateral cross-track and heading errors. It is simpler to formulate, and it is simple to incorporate a dynamic model of the vehicle. As the feedforward controller does not consider future path information disturbances of the system leads to slower system response especially while avoiding the obstacle. Also, LQR is not stable and robust in the presence of uncertainties. Yet the LQR is less reliable as a result of the adoption of a linearized model. The provision to include restrictions like the dynamic and physical practical limits of the truck cannot be provided, even though it solves optimization for a shorter horizon. The computation cost, on the contrary, is a key disadvantage of the optimisation-based controller. With a more intricate dynamic model of the vehicle, LQR cannot offer better accuracy; but, the computing cost is substantially higher.

For SMC, initially, it is evident that even under ideal circumstances, the SMC exhibits a substantial chattering impact at the truck’s varying speeds. With STA in place, the boundary layer thickness has been adjusted, significantly reducing the chattering effects. This will put a less notable strain on the hardware components while implemented in the truck and also help in increased comfort of the passenger or driver. Furthermore, the observed fact that small tuning parameter change results in unstable conditions very quickly. So, the stability of the system is highly dependent on the proper choice of positive constants like sliding gain and lambda in the SMC controller design. According to the findings, the SMC outperforms geometric and LQR controllers in terms of flawless heading tracking, better lateral deviation, and safe distance while avoiding obstacles. The optimal SMC controller proved to be more appropriate for driving on highways than geometrical and LQR controllers. Compared to the above-discussed controllers, MPC provides better control over the vehicle in terms of comfort and constraint handling. Path tracking accuracy is only a matter of how accurately the prediction model can capture the system dynamics of the plant. Thus, improving the prediction model can improve path tracking results, but at the cost of increased computation cost. Also, unlike the reactive controllers, MPC is able to provide both acceleration and steering input which are co-dependent. This provides the ideal scenario for achieving comfort and safety if formulated and tuned well. MPC still can offer more versatility like modeling the obstacle in the state space equations, where the collision avoidance system will not be needed anymore. This cannot be done in controllers without a predictive nature. Also, the ability to handle disturbances, the ability to learn and adapt, and several
such features of MPC can be explored for better results. Different types of MPC like explicit MPC, stochastic MPC, robust MPC, data-driven MPC, hybrid MPC, etc. can offer solutions to different problems. For example, explicit MPC can be used to reduce computation costs by gain scheduling.

Table 7.1: Comparison of a list of the control strategies that were examined in the current thesis project

<table>
<thead>
<tr>
<th>Controller</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPC</td>
<td>Good performance at lower vehicle speeds</td>
<td>Performance degrades at higher vehicle speeds</td>
</tr>
<tr>
<td>Stanley</td>
<td>Performs well at dynamic road conditions</td>
<td>Performs poorly when there is a path discontinuity</td>
</tr>
<tr>
<td>LQR</td>
<td>It is possible to maximize control effort and system response.</td>
<td>Not robust and increases uncertainty because of linearised model</td>
</tr>
<tr>
<td>SMC</td>
<td>Robust performance against uncertainties and external disturbances</td>
<td>Tendency to stimulate high-frequency dynamics that have not been predicted and are susceptible to mismatched disturbances</td>
</tr>
<tr>
<td>MPC</td>
<td>Capability to deal with various factors. States and control might involve constraints. Performance optimization based on a cost function</td>
<td>Solves the computationally expensive online optimization problem</td>
</tr>
</tbody>
</table>

Table 7.2: Summary of a list of the control strategies that were examined in the current thesis project

<table>
<thead>
<tr>
<th>Controller</th>
<th>Computing Cost</th>
<th>Versatile</th>
<th>Ease of Implementation</th>
<th>Robust</th>
<th>Safety and Comfort</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPC</td>
<td>Low</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Stanley</td>
<td>Low</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>LQR</td>
<td>High</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SMC</td>
<td>High</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MPC</td>
<td>Very High</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Chapter 8

Sustainability

The development and implementation of different control algorithms like Pure Pursuit, Stanley, Linear Quadratic Regulator, Sliding Mode Control, Model Predictive Control for path tracking, and collision avoidance for autonomous heavy articulated vehicles prove to align and contribute to several UN sustainability goals. Enhancing safety, efficiency and environmental impact of transportation plays an important role in improving sustainability standards of the automotive industry. The sustainability goals that can addressed are:

8.1 Goal 3: Good Health and Well-being

Road safety is the prime objective that the solutions developed offer to satisfy. By ensuring accurate path tracking and collision avoidance, road accidents can be avoided to a large extent. Especially in heavy vehicles where driving can become exhausting and prone to sleeping while driving, the developed control algorithms can take over to avoid road accidents. This directly supports the improvement of road safety and the well-being of all road users.

8.2 Goal 7: Affordable and Clean Energy

Algorithms like LQR and MPC can reduce energy consumption by providing accurate path tracking at smoother acceleration and deceleration. It also improves the life of mechanical components in the engine and brakes of the vehicle. This leads to reduced fuel consumption and greenhouse gas emissions, promoting clean and sustainable energy practices. Autonomous trucks with the objective of reducing fuel consumption are specific can be developed. Thus, it is also possible to say that this work can lay the foundation to address Goal 13: Climate Action.
8.3 Goal 9: Industry, Innovation, and Infrastructure

This project adds to the innovation in the transportation industry. It aids the growth of intelligent transportation systems and promotes sustainable infrastructure development. The automotive industry is moving forward with autonomous driving in search of a safe, efficient, and comfortable driving experience. Autonomous driving also requires a huge infrastructural development, leading to the overall development of the region. These algorithms enable the realization of autonomous heavy articulated vehicles, revolutionizing logistics and transportation systems.

8.4 Goal 11: Sustainable Cities and Communities

Self-driving heavy vehicles integrated with smart controllers align with Goal 11 by enhancing urban mobility and alleviating congestion on roadways. The accurate path tracking and collision avoidance mechanisms ensure seamless traffic movement, thereby decreasing the volume of vehicles on the streets and mitigating traffic gridlocks. This culminates in more effective urban transportation, elevating the living standards for inhabitants of urban areas.
Chapter 9

Conclusion

This thesis work has presented a critical analysis of a few particular lateral control design strategies utilized to create path tracking for autonomous heavy vehicles. Due to their widespread use in the lateral control design and suitability for heavy truck autonomous vehicles, these control systems were selected. These methods include the Pure Pursuit Controller (PPC), the Stanley Controller (SC) of the geometric based, the Linear Quadratic Regulator (LQR) from optimization based, the Sliding Mode Controller (SMC), and the Model Predictive Control (MPC) from model-based. Because of non-holonomic restrictions like mobile robots with differential drives and truck-like vehicles, it was expected that the AVs would have restricted controllability. The mathematical formulations of two prominent truck models were discussed. In order to assess the effectiveness of the chosen methodologies, a simulation study for highway path-tracking tasks with collision avoidance was also carried out. For the purpose of implementing and enhancing state-of-the-art lateral control of heavy trucks, the simulation results were thoroughly examined, and the benefits and drawbacks of each technique were demonstrated.

According to a thorough analysis of the literature and the outcomes of the controllers’ simulations, MPC appears to be the best option for an autonomous heavy vehicle when it comes to highway driving. Due to their subpar performance at higher speeds, and unsafe collision avoidance, the geometric controllers (i.e. PP and Stanley) were not ideal for highway driving. Whereas, LQR can increase uncertainty if uses a linearized model and the controller is not reliable when there is uncertainty and becomes less robust and also leads to heavy computation for optimization.

In contrast, the robust controller (SMC) performs better than the other controllers when there are uncertainties. However, they are more likely to chatter, which reduces passenger comfort and could put a strain on the hardware. In comparison to other controllers with uncertainties, constraints, and disturbances, optimization-
based controllers like MPC offer considerably smaller lateral and orientation errors. This conclusion supports the use of these approaches when combined with more intelligent learning-based techniques to create a robust and adaptive controller that can respond in real time, regardless of the complexity of the vehicle dynamic model. The current study’s next phase will be used to create a lateral controller that can realize these attributes.
Chapter 10

Future Work

The realm of automotive engineering is undergoing a transformative phase, driven by the rapid advancements in automation, electrification, and control systems. This thesis project marks a pivotal step in this evolution, diving deep into a range of control methodologies including Pure Pursuit, Stanley, Linear Quadratic Regulator, Sliding Mode Controller, Kinematic, and Dynamic Model Predictive Controllers. As it reflects on the present achievements of this study, it’s equally important to envision the exciting future scope it ushers in. Also, methods to measure the chattering/oscillations of the designed controllers are to be explored for effective comparison.

The integration of Artificial Intelligence and Machine Learning into lateral control systems is poised to bring about significant enhancements. Future research could explore the development of adaptive controllers that dynamically adjust their behavior based on real-time environmental conditions and vehicle dynamics. This would lead to safer and more efficient maneuvers, particularly in complex urban and unpredictable scenarios. Further, as vehicles become more connected and equipped with an array of sensors, the fusion of data from cameras, LiDAR, radar, and other sources becomes paramount. Future work could delve into the fusion of sensory inputs to enhance the accuracy of lateral control algorithms. Additionally, the incorporation of perception algorithms would enable vehicles to better understand and respond to the intentions of other road users.

Furthermore, the emergence of platooning, where vehicles autonomously follow one another in close formation, offers promising gains in fuel efficiency and traffic flow. Future research could focus on developing lateral control algorithms that enable seamless coordination and communication among platoon members, ensuring the safety and efficiency of heavy vehicles in closed and controlled environments in these formations. Also, with the progression toward higher levels of automation, ensuring a harmonious interaction between humans and autonomous systems becomes
imperative.

While considering more robustness and safety in challenging conditions, as autonomous vehicles are expected to operate in diverse and sometimes harsh environments, it becomes crucial to develop control systems that exhibit robustness in the face of challenging conditions like adverse weather, poor road surfaces, and unexpected obstacles. Future investigations could focus on the development of controllers that prioritize safety while maintaining optimal vehicle performance in such scenarios. Finally, the evolution of lateral control algorithms necessitates rigorous real-world testing to ensure their reliability and effectiveness. Future efforts could involve field trials and validating the controllers’ capabilities across a variety of real-world scenarios.

This thesis project is not merely a culmination of current methodologies; it is a stepping stone into a future where vehicles navigate with enhanced precision, safety, and efficiency. The insights garnered from this endeavor provide the foundation for a new era of vehicle control systems, unlocking the potential for a smarter, safer, and more connected mobility landscape.
Bibliography


[11] Zhichao Li, Junqiu Li, and Weichen Wang. “Path Planning and Obstacle Avoidance Control for Autonomous Multi-Axis Distributed Vehicle Based on...
Dynamic Constraints”. In: *IEEE Transactions on Vehicular Technology* 72.4 (2023), pp. 4342–4356. doi: [10.1109/TVT.2022.3227447](https://doi.org/10.1109/TVT.2022.3227447).


[34] S. Hamed Tabatabaei Oreh, Reza Kazemi, and Shahram Azadi. “Directional Control of Articulated Heavy Vehicles”. In: SAE International Jour-


[44] Chuan Hu et al. “Output Constraint Control on Path Following of Four-Wheel Independently Actuated Autonomous Ground Vehicles”. In: *IEEE Transaction...*


Appendix

Matrices of LQR

\[
M^* = \begin{bmatrix}
t_{11} & t_{12} & t_{13} & 0 \\
t_{21} & t_{22} & t_{23} & 0 \\
t_{31} & t_{32} & t_{33} & 0 \\
0 & 0 & 0 & 1
\end{bmatrix},
\]

\[
A^* = \begin{bmatrix}
a_{11} & a_{12} & a_{13} & 0 \\
a_{21} & a_{22} & a_{23} & 0 \\
a_{31} & a_{32} & a_{33} & 0 \\
0 & 0 & 0 & 1
\end{bmatrix},
\]

\[
C^* = \begin{bmatrix}
C_1 & C_2 & C_3 & 0 \\
C_1b_1 & -C_2\left(\frac{b_2}{2}\right) & C_3 & 0 \\
0 & 0 & -C_3b_4 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}.
\]

Matrices of SMC

\[
M = \begin{bmatrix}
m_1 + m_2 & -m_2(b_4 + b_5) & -m_2b_4 \\
-m_2(b_3 + b_5) & I_1 + I_2 + m_2(b_4 + b_5)^2 & I_2 + m_2b_4(b_4 + b_5) \\
-m_2b_4 & I_2 + m_2b_4(b_4 + b_5) & I_2 + m_2b_4^2
\end{bmatrix},
\]

\[
D = \begin{bmatrix}
m_1 + m_2 \\
-m_2(b_4 + b_5) \\
-m_2b_4
\end{bmatrix},
\]

\[
U = \begin{bmatrix}
F_{y1} + F_{y2} + F_{y3} \\
F_{y1}b_1 - F_{y2}\left(\frac{b_2+b_4}{2}\right) - F_{y3}(b_3 + b_4 + b_5) \\
-F_{y3}(b_3 + b_4)
\end{bmatrix}.
\]

\[
t_{11} = m_1 + m_2
\]

\[
t_{12} = -m_2(H_s + a_s)
\]

\[
t_{13} = -m_2a_s
\]

\[
t_{21} = -H_s a_s
\]

\[
t_{22} = I_1 + m_2 H_s(H_s + a_s)
\]
\[ t_{23} = m_2 H_s a_s \]
\[ t_{31} = -m_2 a_s \]
\[ t_{32} = I_2 + m_2 a_s (H_s + a_s) \]
\[ t_{33} = I_2 + m_2 a_s^2 \]
\[ a_{11} = -(C_1 + C_2 + C_3) \]
\[ a_{12} = -(m_1 + m_2) \dot{x} - (C_s + (C_3 \cdot \frac{H_t + L_2}{x})) \]
\[ a_{13} = \frac{C_3 \cdot L_2}{x} \]
\[ a_{21} = -\left(\frac{C_1 - C_1 \cdot H_t}{x}\right) \]
\[ a_{22} = H_s m_2 \ddot{x} - (C_q + (C_3 \cdot H_t + \frac{H_t + L_2}{x})) \]
\[ a_{23} = -\left(\frac{C_3 + H_t \cdot L_2}{x}\right) \]
\[ a_{31} = \frac{C_3 \cdot L_2}{x} \]
\[ a_{32} = a_x m_2 \ddot{x} - (C_3 \cdot L_2 \cdot \frac{H_t + L_2}{x}) \]
\[ a_{33} = -\left(\frac{C_3 \cdot L_2^2}{x}\right) \]

where,
\( m_1 \) Total mass of the tractor body
\( m_2 \) Total mass of the first semi-trailer
\( H_t \) Height of the centre of gravity of the tractor’s sprung mass to the roll axis
\( H_s \) Height of the centre of gravity of the semi-trailer’s sprung mass to the roll axis
\( a_s \) Longitudinal acceleration of the tractor unit
\( \dot{x} \) Longitudinal velocity of the tractor in body frame at centre of gravity
\( I_1 \) Yaw moment of inertia of the whole mass of the tractor
\( I_2 \) Yaw moment of inertia of the whole mass of the semi-trailer
\( C_1 \) Combined Cornering stiffness of the tyres for the front axles of the tractor
\( C_2 \) Combined Cornering stiffness of the tyres for the rear axles of the tractor
\( C_3 \) Combined Cornering stiffness of the tyres for the semi-trailer axles
\( C_t \) Damping of the tractor’s suspension
\( C_s \) Damping of the semitrailer’s suspension
\( C_q \) Damping of the coupling point between the tractor and the semitrailer
\( L_2 \) Distance between the fifth wheel point to the rear axle of semi-trailer
\( b_1 \) Distance between the tractor CG and the tractor front axle
\( b_2 \) Distance between the tractor CG and the tractor rear axle
\( b_3 \) Distance between the semi-trailer CG and the rear axle of semi-trailer
\( b_4 \) Distance between the semi-trailer CG and the fifth wheel point
\( b_5 \) Distance from tractor CG to the fifth wheel point