REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
FACULTY OF ELECTRICAL-ELECTRONICS ENGINEERING
DEPARTMENT OF CONTROL AND AUTOMATION ENGINEERING

GRADUATION PROJECT

Lane Changing Application with Discrete Time State Space LTI Bicycle Model and MPC Controller

Advisor: Assist. Prof. Claudia Fernanda YAŞAR

19015036 – Hasan Kaan TUNA

Istanbul, 2024

T.C.

YILDIZ TECHNICAL UNIVERSITY

DEPARTMENT OF CONTROL AND AUTOMATION ENGINEERING

Lane Changing Application with Discrete Time State Space LTI Bicycle Model and MPC Controller

A graduation project submitted by Hasan Kaan TUNA in partial fulfilment of the requirements for the degree of bachelor is approved by the committee on ………………………...... in Department of Control and Automation Engineering, Faculty of Electrical and Electronics Engineering.

Examining Committee

Assist. Prof. Claudia Fernanda YAŞAR

Yıldız Technical University ______________________

Yıldız Technical University ______________________

Yıldız Technical University ______________________

I hereby declare that I have obtained the required legal permissions during data collection and exploitation procedures, that I have made the in-text citations and cited the references properly, that I haven't falsified and/or fabricated research data and results of the study and that I have abided by the principles of the scientific research and ethics during my graduation project under the title of Lane Changing Application with Discrete Time State Space LTI Bicycle Model and MPC Controller supervised by my advisor, Assist. Prof. Claudia Fernanda YAŞAR. In the case of a discovery of false statement, I am to acknowledge any legal consequence.

Hasan Kaan TUNA

ABSTRACT

In my thesis, I aimed to create a educational and instructive application by integrating the concept of Model Predictive Control, one of the advanced control techniques that I find intriguing, with advanced driving systems. Choosing to model it based on a bicycle model, which is simpler than a four-wheeled vehicle but still adequately represents vehicle dynamics, I implemented a lane-changing algorithm, one of the advanced driving techniques, using my controller and vehicle model. Thanks to my thesis, which further increased my interest in Model Predictive Control, I am very pleased to have taken my first step in the field.

I am grateful to my advisor, Assist. Prof. Claudia Fernanda YAŞAR, for her support and assistance.

January 2024
Hasan Kaan Tuna

ÖNSÖZ

Bitirme tezimde ilgi duyduğum ileri kontrol tekniklerinden biri olan model öngörülü kontrol konseptini, ileri sürüş sistemleri ile birleştirerek eğitici ve öğretici bir uygulama yapmak istedim. Dört tekerlekli araçla göre modellemesi daha basit ancak araç dinamiklerini yeteri kadar iyi temsil edebilen bisiklet modelini kullanarak, ileri sürüş tekniklerinden birisi olan şerit değiştirme algoritmasını kontrolcü ve araç modelimle gerçekleştirdim. Model öngörülü kontrole olan ilgimi daha da artıran bitirme tezim sayesinde, alandaki ilk adımımı atmış olmaktan çok memunum.

Danışmanım Dr. Öğr. Üyesi Claudia Fernanda YAŞAR'a destekleri için minnettarım.

Ocak 2024

Hasan Kaan Tuna

CONTENT

LIST OF SYMBOLS

V_x	Longitudinal velocity on of bicycle model on the fixed frame.
m	Vehicle mass.
I_z	Vehicle’s yaw moment of inertia
l_f	Longitudinal distance from centre of gravity to front tire.
l_r	Longitudinal distance from centre of gravity to rear tire.
$C_{\alpha f}$	Cornering stiffness of front tyre
$C_{\alpha f}$	Cornering stiffness of rear tyre
δ	Steering angle
y	Lateral position
ψ	Yaw angle
C_z	Control horizon
H_z	Prediction horizon
$\dot{Y}$	Lateral speed on the fixed frame
$\dot{x}$	Longitudinal speed of vehicle

LIST OF ABBREVIATIONS

MPC	Model Predictive Control
ADAS	Advanced Driver Assistance System
LTI	Linear Time Invariant
LTV	Linear Time Variant

LIST OF FIGURES

Page

LIST OF TABLES

Page

Lane Changing Application with Discrete Time Bicycle State Space LTI Model and MPC Controller

Hasan Kaan Tuna

Department of Control and Automation Engineering

Graduation Project

Advisor: Assist. Prof. Claudia Fernanda YAŞAR

This project aims to enhance reference tracking applications like lane change processes in the automotive industry, particularly in the realm of driver assistance systems, by utilizing Model Predictive Control (MPC) on a bicycle model.

Bicycle model, simpler than the traditional four-wheel vehicle model. This choice renders the project more manageable and feasible, providing an effective platform for designing and implementing control strategies. The bicycle model, while simplified, serves as a reliable foundation by adequately reflecting the fundamental dynamic characteristics of the vehicle, particularly in the lane change scenarios focused on by the project [1].

The method employed in the project involves the use of a bicycle model and an MPC controller in the MATLAB platform. The optimization nature of MPC strategically contributes to improving lane change processes and optimizing the dynamic behaviour of the vehicle. Simulations will be utilized to assess the suitability of control strategies for real-world scenarios, serving as a tool to showcase the effectiveness of the developed system design.

The widespread impact of the project lies in the potential innovation brought by the developed MPC-based lane change system to the automotive industry and driver assistance systems. This system has the capacity to offer safer, more effective, and automated lane change processes, thereby enhancing traffic safety and improving the overall driver experience. The results of the project can serve as an example not only in this field but also in integrating similar control strategies into other modes of transportation, broadening the application spectrum of these technologies.

Keywords: MPC, system dynamics and modelling, bicycle model, ADAS, smart design

Ayrık Zamanlı Doğrusal Zamanlı Değişmez Durum Uzayı Modeli ve Model Öngörümlü Deneyleyici ile Şerit Değiştirme Uygulaması

Hasan Kaan Tuna

Kontrol ve Otomasyon Mühendisliği Bölümü

Bitirme Çalışması

Danışman: Dr. Öğr. Üyesi Claudia Fernanda YAŞAR

Bu proje, özellikle sürücü destek sistemleri alanında, otomotiv endüstrisindeki referans takip uygulamalarını geliştirmeyi amaçlamaktadır, özellikle şerit değiştirme süreçlerini. Bunun için Bisiklet Modeli üzerinde Model Tahmin Kontrolü (MPC) kullanılmaktadır.

Bisiklet modeli, geleneksel dört tekerlekli araç modelinden daha basittir. Bu tercih, projeyi daha yönetilebilir ve uygulanabilir hale getirir, kontrol stratejileri tasarlamak ve uygulamak için etkili bir platform sunar. Bisiklet modeli, basitleştirilmiş olmasına rağmen, projenin odaklandığı şerit değiştirme senaryolarında aracın temel dinamik karakteristiklerini yeterince yansıtarak güvenilir bir temel olarak hizmet eder [1].

Projede kullanılan yöntem, MATLAB platformunda bir bisiklet modeli ve bir MPC denetleyicisinin kullanımını içerir. MPC'nin optimizasyon doğası, şerit değiştirme süreçlerini iyileştirmeye ve aracın dinamik davranışını optimize etmeye stratejik olarak katkıda bulunur. Simülasyonlar, kontrol stratejilerinin gerçek dünya senaryoları için uygunluğunu değerlendirmek için kullanılacak ve geliştirilen sistem tasarımının etkinliğini sergilemek için bir araç olarak hizmet edecektir.

Projenin geniş etkisi, geliştirilen MPC tabanlı şerit değiştirme sisteminden otomotiv endüstrisine ve sürücü destek sistemlerine getirilen potansiyel yenilikte yatar. Bu sistem, daha güvenli, daha etkili ve otomatik şerit değiştirme süreçleri sunma kapasitesine sahiptir, bu da trafik güvenliğini artırır ve genel sürücü deneyimini iyileştirir. Projenin sonuçları, sadece bu alanda değil, aynı zamanda benzer kontrol stratejilerini diğer ulaşım modlarına entegre etme konusunda da bir örnek olarak hizmet edebilir.

Anahtar Kelimeler: MPC, sistem dinamiği ve modelleme, bisiklet modeli, İSDS, akıllı tasarım

SECTION 1

INTRODUCTION

1.1 Importance of the MPC and Advanced Driver Assistance Systems

The advent of autonomous vehicles and advanced driver assistance systems has propelled the need for robust and efficient lane-changing algorithms. Lane changing, a fundamental aspect of dynamic driving scenarios, requires a careful balance between safety, efficiency, and comfort. In this context, the integration of a Discrete Time State Space Linear Time-Invariant (LTI) Bicycle Model and a Model Predictive Control (MPC) strategy holds promise for enhancing the precision and adaptability of lane-changing manoeuvres.

Lane-changing applications are critical for ensuring the fluidity of traffic and optimizing vehicle trajectories in response to dynamic environmental conditions. Traditional methods for modelling vehicle dynamics often rely on continuous-time models, but the discrete-time formulation offers advantages in terms of computational efficiency and ease of integration with control algorithms. The LTI Bicycle Model, with its simplified yet representative dynamics, provides a suitable foundation for capturing the essential characteristics of vehicular motion during lane changes [2].

The use of Model Predictive Control further advances the state of the art by enabling real-time decision-making based on predictive models of the vehicle's behaviour. MPC considers not only the current state of the vehicle but also anticipates future states, allowing for proactive adjustments to optimize the lane-changing process. By combining the discrete-time LTI Bicycle Model with MPC, this research seeks to address the challenges associated with lane-changing applications, such as trajectory planning, collision avoidance, and overall driving comfort.

This thesis aims to contribute to the existing body of knowledge by exploring the integration of the model and controller in the context of lane changing. The following sections will delve into the theoretical underpinnings of the Discrete Time State Space LTI Bicycle Model and Model Predictive Control, detailing their integration, implementation, and subsequent evaluation through simulation scenarios. Through this exploration, we seek to provide insights into the effectiveness and potential applications of this approach in enhancing the performance of lane-changing algorithms in autonomous vehicles and advanced driver assistance systems [3].

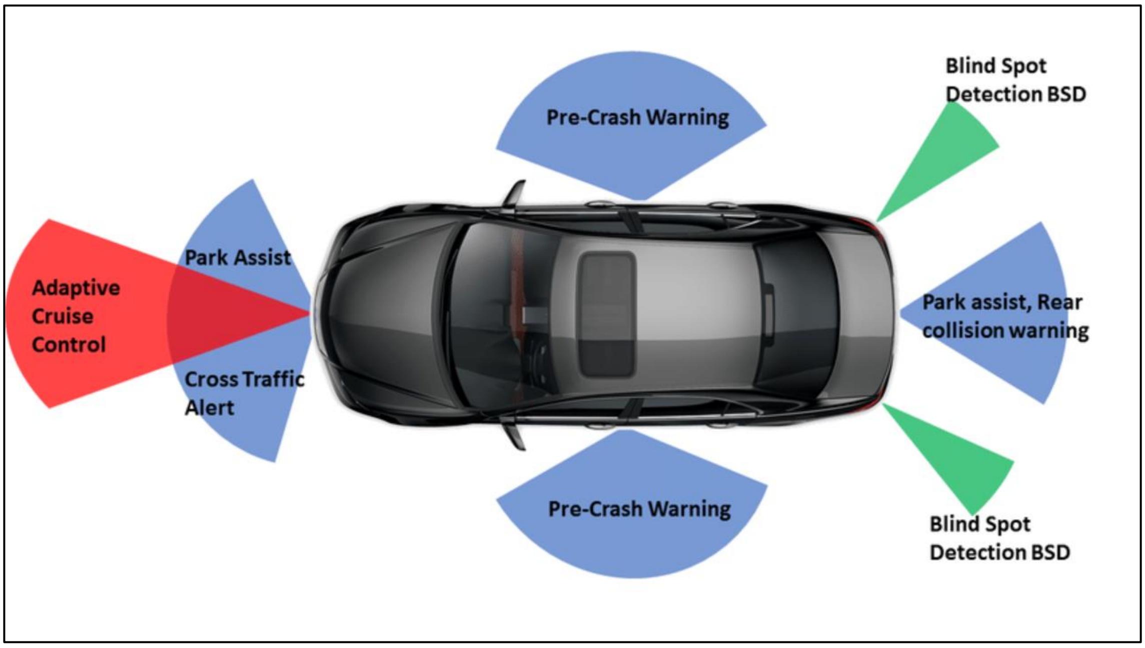


Figure 1.1: Advanced Driver Assistance Systems [3]

1.2 Literature Survey

1.2.1 Lane Changing Algorithms

Lane changing is a critical aspect of vehicular dynamics, influencing traffic flow, safety, and overall driving experience. Various algorithms have been proposed to address the challenges associated with efficient and safe lane changes. Classic methods, such as rule-based and behaviour-based approaches, have been widely explored. Rule-based algorithms rely on predefined sets of rules to govern lane-changing decisions, while behaviour-based approaches leverage heuristics and decision trees based on observed vehicle behaviours.

Recent advancements have seen a shift towards more sophisticated techniques, including machine learning-based algorithms. Reinforcement learning, in particular, has shown promise in learning optimal lane-changing policies from data, allowing vehicles to adapt to diverse and dynamic traffic scenarios.

1.2.2 Vehicle Modelling for Lane Changing

Accurate vehicle dynamics modelling is essential for the success of lane-changing algorithms. Continuous-time models, such as the dynamic bicycle model, have traditionally been employed to capture the intricate details of vehicle motion. However, the computational demands associated with these models can be a limiting factor in real-time applications.

The Discrete Time State Space Linear Time-Invariant (LTI) Bicycle Model offers an attractive alternative by providing a simplified yet sufficiently accurate representation of vehicle dynamics. This model discretizes time, making it computationally efficient, and offers flexibility in integration with control algorithms. The choice of an appropriate vehicle model is crucial for achieving a balance between accuracy and computational efficiency in the context of lane-changing applications.

1.2.3 Model Predictive Control (MPC) for Lane Changing

Model Predictive Control has emerged as a powerful paradigm for controlling dynamic systems, including vehicles. MPC is particularly well-suited for lane-changing applications due to its ability to consider predictive models and optimize control actions over a finite time horizon. The predictive nature of MPC allows for anticipatory decision-making, essential for safe and efficient lane changes.

Numerous studies have explored the application of MPC in lane-keeping and lane-changing scenarios. These studies highlight the benefits of MPC in handling uncertainties, adapting to varying driving conditions, and addressing safety concerns. The integration of MPC with different vehicle models, including continuous and discrete-time formulations, has been investigated to enhance the overall performance of lane-changing algorithms.

1.2.4 Integration of Discrete Time LTI Bicycle Model and MPC

While individual components of the proposed approach—Discrete Time LTI Bicycle Model and MPC—have been studied independently, the literature lacks a comprehensive exploration of their integration specifically for lane-changing applications. This thesis aims to bridge this gap by examining the synergies between the discrete-time vehicle model and MPC, offering insights into the collective impact on the precision and adaptability of lane-changing manoeuvres.

In summary, the literature review underscores the importance of advanced lane-changing algorithms, the significance of accurate vehicle dynamics modelling, and the potential of MPC in enhancing control strategies. The subsequent sections of this thesis will build upon this foundation, exploring the integration of the Discrete Time LTI Bicycle Model and MPC for lane-changing applications.

1.3 Objective of Study

As of the present moment, motor vehicles worldwide are undergoing a significant transformation driven by technological advancements and shifting paradigms in the automotive industry. The transition towards electric and hybrid vehicles has gained momentum, with many major automakers investing heavily in the development and production of electric models. The integration of autonomous driving technologies is also rapidly progressing, with an increasing focus on enhancing vehicle safety and efficiency. Additionally, sustainability has become a key consideration, leading to the development of eco-friendly alternatives and a growing emphasis on reducing the environmental impact of traditional combustion engine vehicles. The convergence of connectivity and smart technologies is reshaping the in-car experience, with features such as advanced infotainment systems, connectivity options, and vehicle-to-everything communication gaining prominence. Overall, the current state of motor vehicles reflects a dynamic landscape characterized by innovation, sustainability initiatives, and a gradual shift towards cleaner, smarter, and more connected transportation solutions.

The need for innovation in the control systems of motor vehicles is propelled by the ongoing evolution in the automotive landscape, characterized by technological advancements and changing consumer preferences. Control theory plays a pivotal role in addressing several critical challenges and opportunities within the automotive industry.

The rising prominence of electric and hybrid vehicles necessitates sophisticated control systems to manage the intricate dynamics of these alternative powertrains. Control theory enables the optimization of energy flow, battery management, and the coordination of multiple power sources, ensuring efficient and seamless transitions between electric and internal combustion modes.

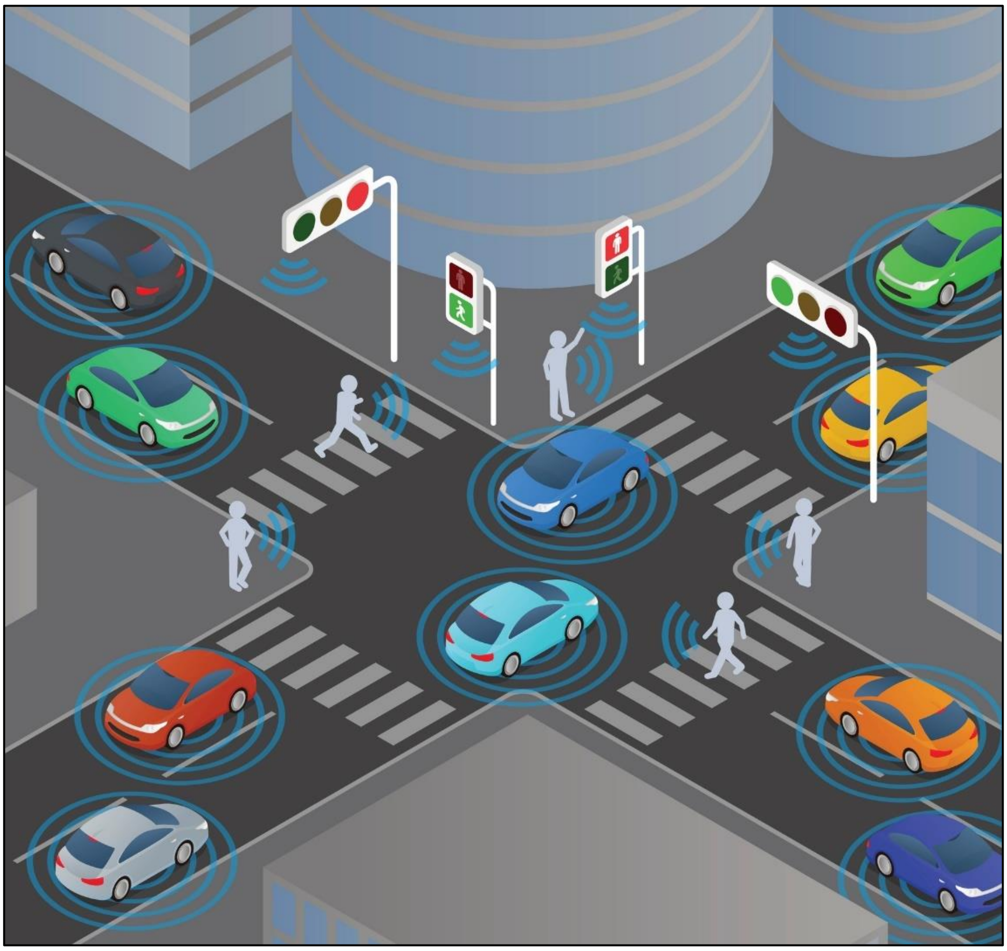


Figure 1.2: Advanced Driver Systems' Enviromental Representation [3]

The pursuit of autonomous vehicles demands advanced control algorithms for navigation, obstacle avoidance, and decision-making. Control theory is essential in developing robust systems that can interpret sensor data, adapt to dynamic environments, and make real-time decisions to ensure the safety and reliability of self-driving vehicles.

Control theory contributes to enhancing the overall energy efficiency of vehicles, especially regarding combustion engines. Innovations in engine control systems, such as advanced fuel injection strategies and adaptive cruise control, play a crucial role in optimizing fuel consumption and reducing emissions.

The integration of smart technologies and connectivity features in vehicles requires sophisticated control systems. Control theory is applied to manage communication networks within the vehicle, ensuring seamless interactions between various components and external systems. This facilitates the implementation of features like advanced driver assistance systems (ADAS) and in-car connectivity platforms.

Control systems must adapt to dynamic traffic conditions to optimize vehicle performance, enhance safety, and improve overall traffic flow. Adaptive cruise control, traffic-aware navigation, and cooperative adaptive cruise control systems utilize control theory principles to respond to real-time changes in traffic patterns.

In essence, innovation in control theory for motor vehicles is indispensable to address the diverse challenges posed by the industry's transformation. From managing alternative power sources to enabling autonomous capabilities and enhancing overall vehicle efficiency, control theory serves as the foundation for developing smart, efficient, and sustainable transportation solutions.

Motor vehicles equipped with Model Predictive Control (MPC) controllers have the potential to bring about transformative and positive changes in various aspects of the world. The capabilities of such vehicles extend beyond traditional control systems, offering a range of benefits that contribute to safety, efficiency, and sustainability.

MPC controllers excel in predicting and responding to dynamic conditions, allowing vehicles to make real-time decisions to avoid collisions and ensure safer driving experiences. The ability to anticipate and react to potential hazards contributes significantly to road safety, reducing the frequency and severity of accidents. Also, MPC controllers can optimize the operation of internal combustion engines, hybrid powertrains, and electric vehicles by dynamically adjusting parameters such as throttle response, gear shifts, and energy distribution. This optimization leads to improved fuel efficiency and reduced environmental impact, supporting sustainability efforts and decreasing overall fuel consumption.

MPC controllers enable vehicles to adapt their speed and behaviour based on real-time traffic conditions. This contributes to smoother traffic flow, minimizes congestion, and reduces travel times, enhancing overall transportation efficiency. And they play a crucial role in the development of autonomous vehicles by providing a framework for predictive decision-making. These controllers allow autonomous cars to anticipate the behaviour of surrounding vehicles, pedestrians, and other dynamic elements, ensuring safe and efficient navigation. Therefore, MPC controllers can be utilized in adaptive cruise control systems, allowing vehicles to maintain a safe following distance from the vehicle ahead. This feature enhances highway driving comfort, reduces the likelihood of accidents, and optimizes traffic flow.

For electric vehicles, MPC controllers help manage the energy consumption of the vehicle, extending the range by optimizing the use of the battery. This is crucial for promoting the widespread adoption of electric vehicles and reducing reliance on traditional fossil fuels.

Through precise control of combustion processes and energy management, MPC controllers contribute to minimizing emissions from vehicles. This is especially relevant as societies globally strive to reduce air pollution and combat climate change. Additionally, MPC controllers can enhance the overall driving experience by ensuring smoother transitions between different driving conditions, leading to increased driver and passenger comfort. This is particularly relevant in the context of the growing emphasis on in-car connectivity and advanced driver assistance systems.

In summary, motor vehicles equipped with MPC controllers can change the world positively by fostering safer roads, reducing environmental impact, and enhancing overall transportation efficiency. The predictive and adaptive nature of MPC technology makes it a key enabler for the ongoing transformation of the automotive industry towards safer, more sustainable, and intelligent mobility solutions.

Bicycle model is often considered as simplified versions of four-wheeled vehicles, have evolved from basic transportation. Nevertheless, enhancing the dynamic control and performance optimization of bicycles, particularly during complex manoeuvres, poses significant challenges compared to traditional four-wheeled vehicles. Overcoming these challenges and improving the control systems of bicycles are critical for enhancing both safety and comfort. This essay delves into the objectives and significance of a control project aimed at implementing Model Predictive Control (MPC) for the lane-changing or various reference tracking process of bicycles.

1.4 Steps to be Taken in This Project

1.4.1 Dynamic Modelling and Analysis

Understanding and analysing the complex dynamic model of a bicycle is a foundational step in determining the control strategy. This phase involves examining the interaction of factors such as bicycle speed, turning angle, and lane-changing, establishing the groundwork for a dynamic model that will inform the control strategy.

1.4.2 Integration of Model Predictive Control (MPC)

MPC, as a control strategy, predicts the future behaviour of a system and optimizes control movements based on these predictions. In this phase, the MPC controller will be integrated with the dynamic model of the bicycle, and control algorithms suitable for the lane-changing process will be designed.

1.4.3 Lane Change Algorithm

A lane change algorithm will be designed to facilitate a safe and effective lane-changing manoeuvre by considering various parameters of the bicycle model, including speed, road conditions, and environmental factors. This algorithm will utilize control signals generated by the MPC controller in real-time to determine optimal manoeuvre parameters.

1.4.4 Parameter Optimization and Control

The bicycle model created in the MATLAB environment and the MPC controller, integrated with the lane change algorithm, will undergo project simulation. Simulation results will be evaluated to assess the lane-changing performance of the bicycle model, and if necessary, control parameters will be optimized.

1.5 Technological Advantages of the Project

Key technologies employed in this project include Model Predictive Control (MPC), and MATLAB. The MPC controller stands out for its predictive capabilities in MATLAB's analytical and computational tools will be utilized to meet the project's requirements.

1.5.1 Safe and Efficient Lane Changing

The developed control strategy aims to enable vehicles to change lanes safely and efficiently. This ensures that vehicle drivers can better adapt to the complexity of traffic, resulting in safer journeys.

1.5.2 Technological Advancements and Applicability

The project showcases the applicability of the MPC control approach to vehicles, potentially inspiring technological advancements in this field. These advancements may not only benefit vehicles but also offer insights for control technologies in other transportation vehicles.

1.5.3 Support for Sustainable Transportation

The widespread use of motor vehicles promotes sustainable transportation. By improving vehicle control systems, this project contributes to more effective and safe sustainable transportation practices.

Optimizing the lane-changing process of vehicles is a crucial step towards promoting sustainable and safe transportation. The objectives and significance of this project underscore the potential of integrating the MPC control strategy into vehicle control systems, emphasizing its effectiveness in enhancing the mobility and safety of vehicles. By offering a fresh perspective on control technologies for vehicles and potentially influencing advancements in the field, this project contributes to the broader landscape of transportation systems.

1.6 Methodology of the Project

This section elucidates the methodologies employed in the development of an advanced vehicular system capable of executing lane-changing manoeuvres through the application of the Model Predictive Control (MPC) strategy. The primary aim of this project is to rigorously articulate the dynamic model of the vehicle in mathematical terms and subsequently integrate this model with an MPC controller. The resulting control strategy is meticulously designed to facilitate safe and efficient lane-changing operations. The inaugural phase entails the comprehensive definition of the dynamic model, encompassing various essential parameters expressed through differential equations. After that, the dynamic model harmonizes with the MPC control strategy, culminating in the formulation of a specialized lane-changing algorithm. The evaluation and optimization of the control strategy's efficiency are undertaken through simulations conducted using MATLAB. Ultimately, the optimized control strategy undergoes empirical validation through real-time testing on a prototype vehicle. This methodological sequence encapsulates the foundational approach utilized in the systematic development of vehicles adept at executing lane-changing manoeuvres, guided by the precision of an MPC controller.

1.6.1 Obtaining Dynamic Model of the Vehicle

The fundamental cornerstone of this project involves the formulation of a comprehensive dynamic model to characterize the behaviour of the vehicle during lane changing manoeuvres. The dynamic model aims to capture the intricate interplay of essential parameters that define the vehicle's motion. This includes the translational and rotational dynamics, incorporating variables such as vehicle speed, lateral acceleration, steering angle, and tire-road friction coefficients.

1.6.1.1 Kinematics and Dynamics Equations

The kinematics equations describe the motion of the vehicle without considering the forces involved. These equations involve relationships between the vehicle's position, velocity, and steering angle. On the other hand, the dynamic equations delve into the forces and torques acting on the vehicle, considering factors such as mass, inertia, and tire characteristics. The combined set of kinematics and dynamics equations formulates a holistic representation of the vehicle's motion. Vehicle dynamics for a four-wheeled vehicle are given below.

The dynamic modelling of a four-wheeled vehicle is highly complex, involving a multitude of parameters and intricate interactions between the vehicle components. The comprehensive representation of all relevant factors, including individual wheel dynamics, suspension characteristics, and the complex geometry of a four-wheeled vehicle, can lead to a highly intricate mathematical model. Recognizing the challenges associated with this level of complexity, I have chosen to adopt a simplified yet effective approach by working with a bicycle model. A bicycle model offers a reasonable compromise, capturing the essential dynamics of a vehicle during lane-changing manoeuvres without the computational burden associated with a full four-wheeled vehicle model. This simplification allows for a more focused and streamlined development process, facilitating a clearer understanding of the core principles involved in the control strategy and enhancing the feasibility of real-time implementation.

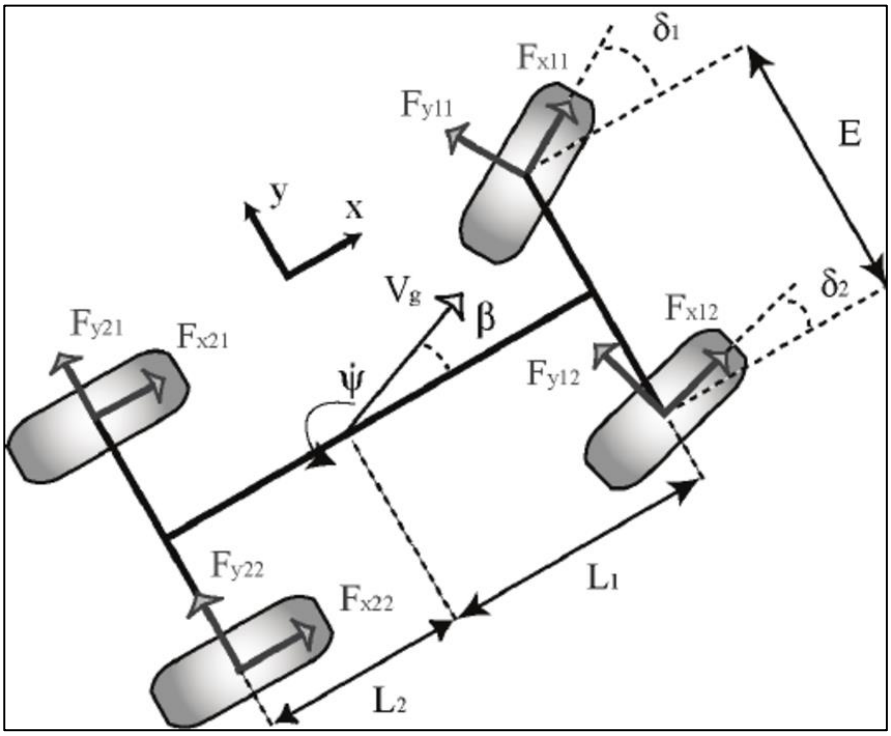


Figure 1.3: Modern Four-Wheel Vehicle Dynamics [4]

The bicycle model is a simplified representation employed in vehicle dynamics and control systems, particularly for lateral manoeuvres like turning or lane changing. This model divides the vehicle's motion into longitudinal and lateral components, assuming a single point of contact with the road typically located at the centre of the wheelbase. It incorporates key concepts such as yaw rate and slip angles to describe the vehicle's rotation around the vertical axis and the deviation between its pointing direction and actual movement. The primary control input is the steering angle, allowing simulation of the vehicle's directional changes. Despite its simplicity, the bicycle model is valuable for analysing basic vehicle dynamics. Mathematically described through differential equations, it serves as a foundation for more complex vehicle models in research and control strategy design. This simplified approach facilitates a clearer understanding of essential principles and is particularly useful for scenarios where a detailed, computationally intensive model may not be necessary. Therefore, general vectoral representation of a bicycle model is given in Figure (1.4) and Figure (1.5).

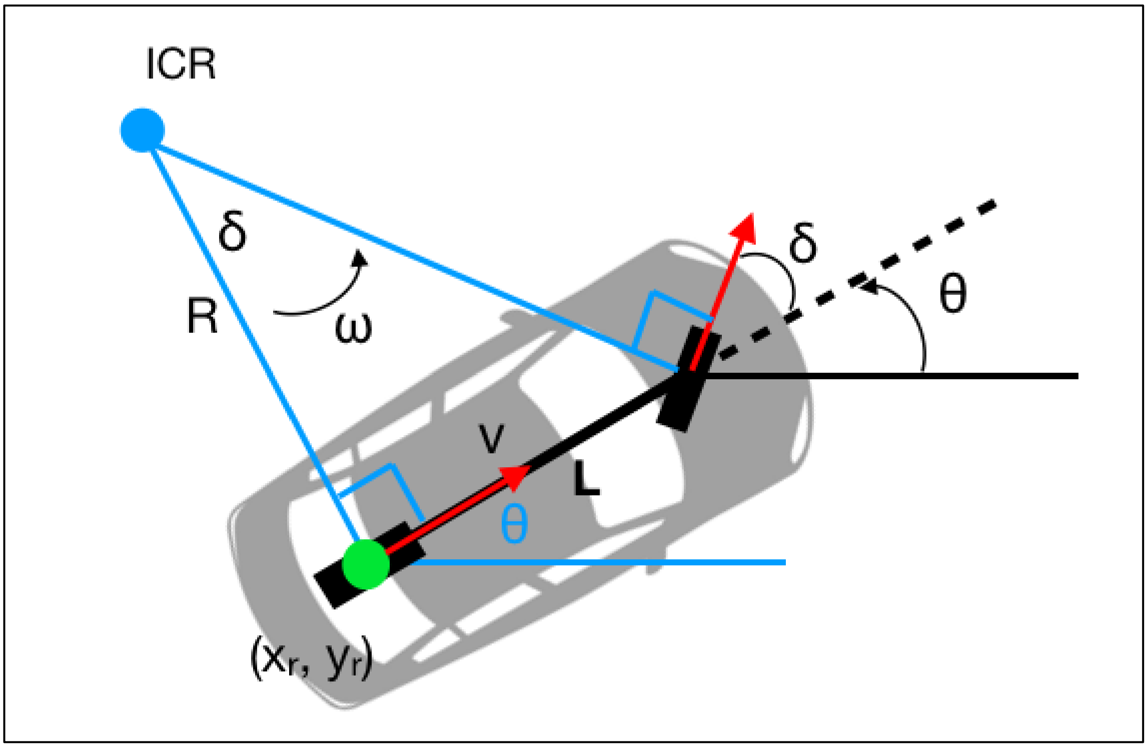


Figure 1.4: Dynamics of The Bicycle Model [1]

In Figure (1.4), Instantaneous centre of rotation (ICR) represents the intersection point of the perpendicular axes drawn to the wheels. X_r, Y_r and X_f, Y_f represents selected reference points of rear and front tyre which belongs to the bicycle model. The dynamic equations, which describe the motion of the vehicle without considering the forces involved, have provided a foundational understanding of the bicycle model's behaviour. As a next step, facilitate integration with control strategies such as the Model Predictive Control (MPC), the transition to a state-space representation is imperative. This transformation involves expressing the system in terms of state variables that encapsulate the essential aspects of the vehicle's motion and dynamics. The state-space representation is pivotal for the subsequent phases of the project, enabling a more nuanced and flexible approach to the implementation of control strategies and simulations [1].

1.6.2 MPC Control Strategy

Model Predictive Control (MPC) stands as a paradigmatic approach in advanced control systems, renowned for its theoretical underpinnings and practical efficacy in managing complex, multivariable systems subject to dynamic constraints. At its essence, MPC is a receding-horizon optimization strategy that leverages a dynamic model of the system to predict its future states. The methodology strategically integrates a cost function that encapsulates desired system performance metrics, constraints, and optimization criteria. A key feature of MPC is its ability to iteratively compute optimal control inputs over a finite time horizon, continually adjusting the control trajectory as the system evolves dynamically. This predictive and adaptive nature distinguishes MPC from other control strategies, making it particularly well-suited for applications requiring real-time responsiveness and adaptability to uncertain and varying environments [4].

The mathematical formulation of MPC involves the discretization of the dynamic system model, often represented by state-space equations derived from system dynamics or identified through empirical data. This discretization facilitates the computation of predicted future states based on the current system state and applied control inputs. The optimization problem within each iteration of MPC seeks to minimize the cost function, accounting for both control and state variables, subject to a set of constraints that reflect system limitations, environmental conditions, and operational requirements.

MPC has found widespread application in diverse domains, notably in automotive control systems where its ability to handle complex multivariable systems, account for constraints such as actuator limits, and adapt to varying driving conditions makes it instrumental. Beyond automotive applications, MPC has proven effective in process industries, robotics, energy systems, and aerospace, showcasing its versatility across a spectrum of dynamic systems [4].

The theoretical foundations of MPC involve a deep understanding of control-oriented models, system identification techniques, and numerical optimization methods. The control-oriented models, derived from the physics of the system or acquired through identification processes, play a pivotal role in ensuring an accurate representation of the underlying system dynamics. The optimization methods employed within MPC often include numerical techniques such as quadratic programming or convex optimization to solve the constrained optimization problem efficiently.

In summary, MPC stands as a formidable control strategy characterized by its predictive, adaptive, and optimization-centric nature. The methodology's technical intricacies span dynamic system modelling, discretization, optimization, and theoretical underpinnings, making it a cornerstone in the control systems landscape, particularly for applications demanding precision, adaptability, and robust performance in the face of complexity and uncertainties. Figure (1.6) provides an overview of the most important parameters that should be selected according to the control system being studied.

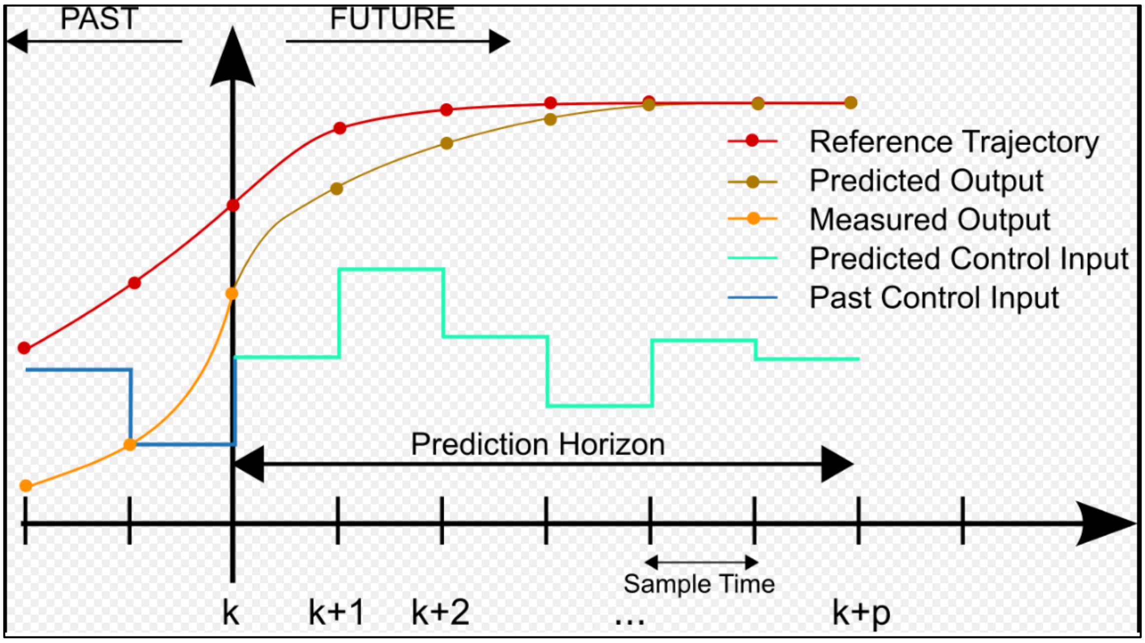


Figure 1.5: General Working Principle of MPC [4]

1.6.2.1 Definition of the Controller Parameters

Sample time (Ts): A typical starting guess consists of setting the controller sample time so that 10 to 20 samples cover the rise time of the plant.

Prediction horizon (Hz): The number of future samples over which the controller tries to minimize the cost. It should be long enough to capture the transient response and cover the significant dynamics of the system. A longer horizon increases both performance and computational requirements. A typical prediction horizon is 10 to 20 samples.

Control Horizon (Cz): The control horizon, a critical parameter in Model Predictive Control (MPC), denotes the number of consecutive control moves considered by the controller to minimize the cost over the prediction horizon. Analogous to the prediction horizon, an extended control horizon enhances both control performance and computational requirements. A recommended guideline is to set the control horizon between 10% to 20% of the prediction horizon, ensuring a minimum of two to three steps. This balance facilitates effective optimization while maintaining computational efficiency.

Nominal Values: In instances where the plant is derived from the linearization of a nonlinear model around an operating point, establishing nominal values for input, state, state derivative (if nonzero), and output is advisable. This practice enables the specification of constraints on actual inputs and outputs rather than deviations from nominal values.

Scale Factors: It is recommended to define scale factors for each plant input and output, particularly when their ranges and magnitudes significantly differ. Appropriate scale factors enhance the numerical stability of the optimization problem and simplify weight tuning. A general guideline is to set a scale factor approximately equal to the span (the difference between the maximum and minimum value in engineering units) of the relevant signal.

Constraints: Constraints in MPC often mirror physical limits, and they can be categorized as hard (non-negotiable in optimization) or soft (can be violated within limits). Optimal practice involves imposing hard constraints, if necessary, on inputs or their rates of change, while treating output constraints, if necessary, as soft constraints. Imposing hard constraints on both inputs and outputs can lead to infeasibility and is generally discouraged.

Weights: Fine-tuning the performance of an MPC controller involves adjusting the tuning weights within the cost function. Larger output weights typically enhance aggressive reference tracking, while increased weights on manipulated variable rates promote smoother control moves, enhancing robustness.

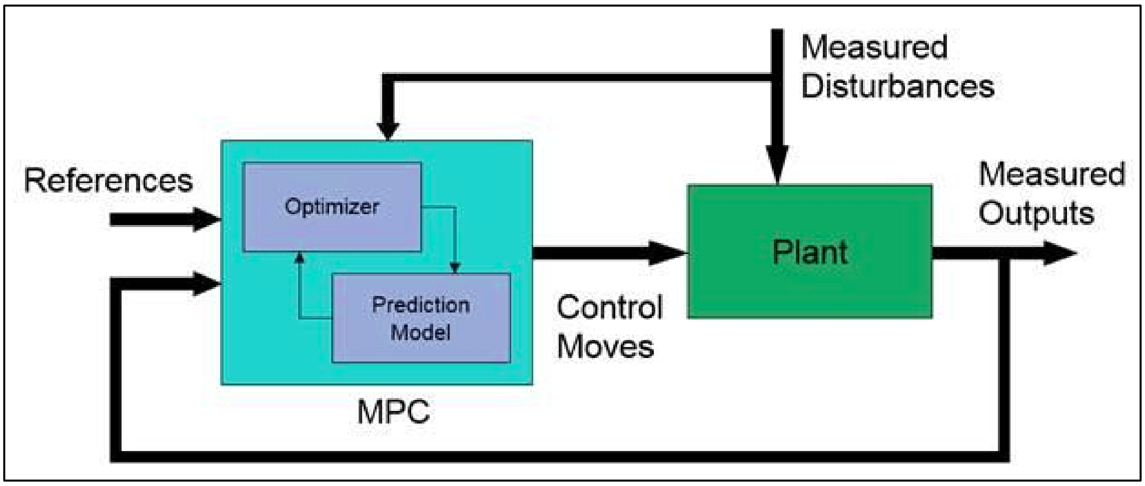


Figure 1.6: General System Suggestion with MPC and Plant Model [4]

Model Predictive Control (MPC) is known as a control strategy aiming to optimize system behaviour and typically utilizes a state-space model. However, when measured outputs cannot be directly fed into MPC, situations may arise where state estimation or measurement inference methods become crucial. MPC generally seeks to directly measure the states and outputs of a system. However, there are instances where not all outputs are measurable, or measurement errors may occur. In such cases, instead of directly feeding measurements into MPC, it can be advantageous to use a state estimation method to predict the internal states of the system. A classical state estimation method, the Kalman Filter, predicts the internal states of the system using measured outputs and the system model. These predictions enable MPC to access real-time updated state information. The Kalman Filter stands out with its ability to correct measurement errors and uncertainties in the system model. These methods can help MPC cope with measurement errors or non-linearities in the system model more effectively. In conclusion, when situations arise where measurements cannot be directly used in MPC applications, integrating a state estimation method can be a crucial strategy. Methods like the Kalman Filter can enhance the performance of MPC by predicting system states and assisting in handling measurement uncertainties. This approach contributes to the design of more robust and reliable MPC systems in real-world applications.

1.6.3 Possible Reference Tracking Algorithm and MATLAB/Simulink Simulations

In a Simulink implementation of a Model Predictive Control (MPC) system utilizing a state-space model of a vehicle, the dynamics of the vehicle are typically represented through a set of interconnected state-space blocks. These blocks encapsulate the linearized equations governing the motion of the vehicle, considering factors such as longitudinal and lateral dynamics. The state-space representation is then integrated into Simulink using appropriate block State-Space. The MPC controller, which is going to be designed and tuned, is implemented as a custom block, or using Simulink's Control System Toolbox. Reference points, representing desired vehicle behaviour, are incorporated into the system as input signals. The simulation unfolds by iteratively solving the MPC optimization problem at each time step, adjusting the control inputs based on the difference between the current and desired states. Simulink's simulation engine dynamically updates the vehicle states, allowing for a real-time assessment of the MPC controller's impact. Outputs in the form of reference trajectories, actual trajectories, and control inputs provide visual insights into the system's behaviour, aiding in the fine-tuning of the MPC algorithm and ensuring the vehicle responds optimally to the reference points in various driving scenarios. This Simulink-based approach offers a comprehensive and visually intuitive platform for the development, analysis, and validation of MPC controllers for vehicle dynamics [5].

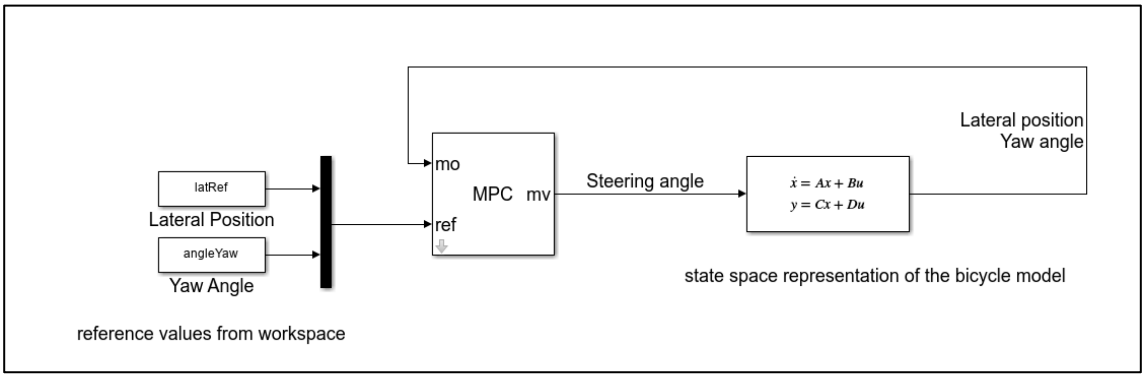


Figure 1.7: System Suggestion with MPC and Plant Model for Simulink [5]

VEHICLE MODELLING PRINCIPLES

Modelling is huge part of control area. Plant models mostly determine the outputs, and those outputs are used to calculate error values and finally, the controllers determine how to they shall operate according to the errors. Therefore, vehicle modelling is the starting point of this Project.

2.1 Line Changing Manoeuvre

Line changing manoeuvre is very common in the modern days of humanity. While humans driving vehicles, they usually perform lane changing manoeuvres. But they do not really pay attention to dynamics and physics behind it. Lane changing manoeuvre can be defined as the action of a vehicle moving from one lane to another on a road or highway. To perform such action, front Wheel shall turn with angle of “δ”. While the vehicle is turning with angle of “δ”, it also has “ψ” as yaw angle [2].

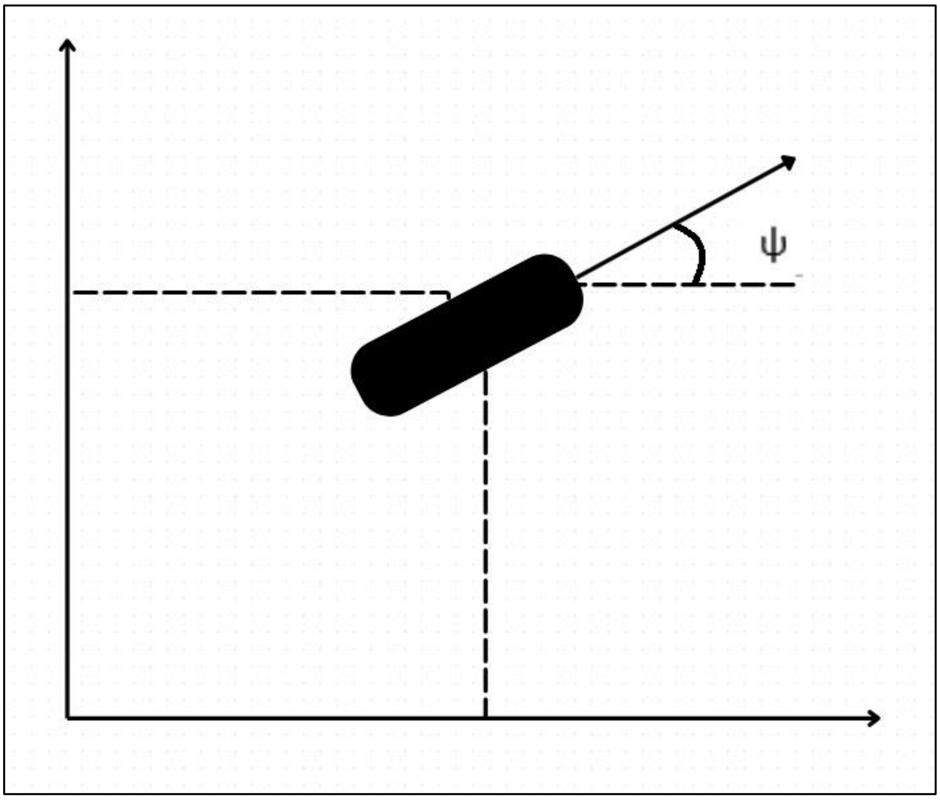


Figure 2.1: Definition of Yaw Angle of The Vehicle On The Fixed Frame

2.2 Forces and Moments

2.2.1 Force

In physics, force is a vector quantity that describes the interaction between objects, causing a change in their motion or deformation. It is measured in newtons (N). Force can be classified into categories such as gravitational, electromagnetic, and contact forces. It can be derived from the following formulation.

$$F = \mathrm{m.a} \tag{2.1}$$

The formulation above is origin of the automotive and vehicle dynamics. In that formulation, m stands for vehicle mass with unit of (kg) and a stands for vehicle acceleration in the International System of Units (SI). ψ

2.2.2 Moment (Torque)

Moment (Torque) is the tendency of a force to rotate an object around an axis or pivot point. It is also a vector quantity and is measured in (N.m) in the International System of Units (SI). Moments can cause objects to rotate or experience angular acceleration.

$$M = \mathrm{I}\,\ddot{\alpha} \tag{2.2}$$

The formulation above is also very commonly used to determine dynamics of rotating parts. In that formulation, I, stands for moment of inertia with unit of (kg.m^2) and $\ddot{\alpha}$ stands for second derivative of angle (angular acceleration). Moment of Inertia differs from mass with its rotational feature. So, it can be defined as rotating mass. Therefore, moment of inertia is highly important for rotating parts and vehicles. If the two objects with same mass are imagined as the Figure (2.2), the object has hole in it has larger moment of inertia. Because, it has larger amount of mass away from its gravitational centre.

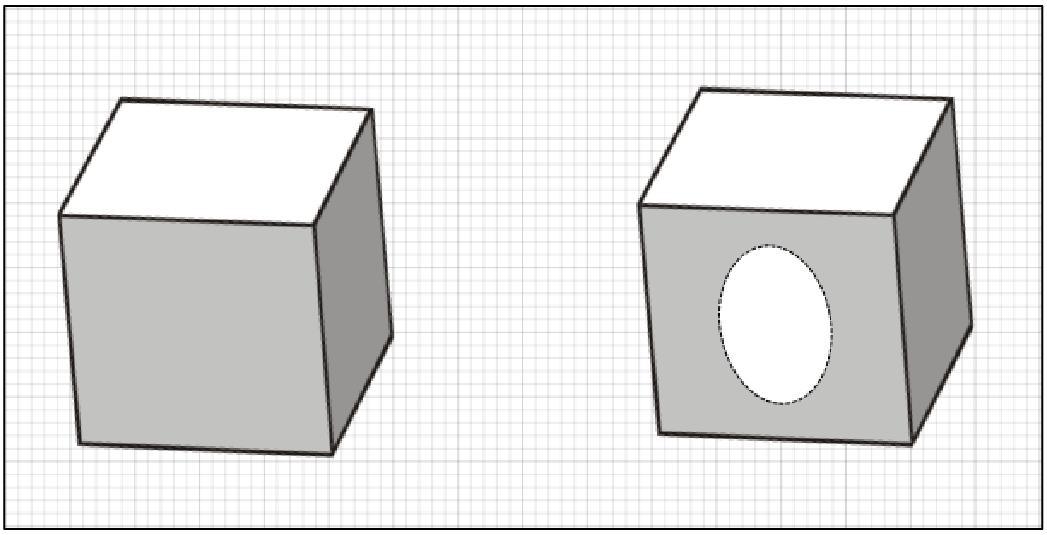


Figure 2.2: Moment of Inertia Example of Two Cubes with Same Mass

2.3 Visualization of Lane Changing Manoeuvre

When a car changes lanes, the driver initiates the manoeuvre by steering in the desired direction, possibly adjusting the speed, and activating the turn signal to signal their intention. Throughout the process, lane change involves smoothly adjusting the car's position within the lane and merging into the target lane while maintaining a consistent speed. That indicates the successful execution of the manoeuvre.

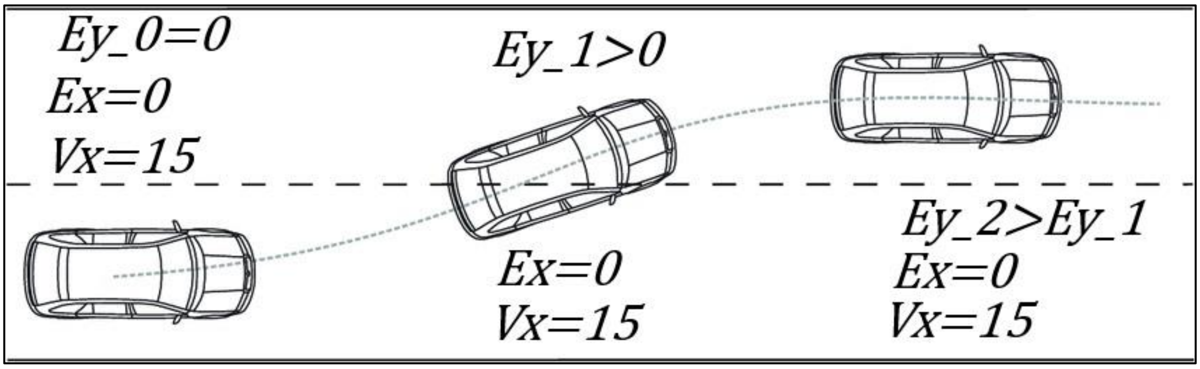


Figure 2.3: Lane Changing Manoeuvre and Error Determination [7]

When a vehicle that following a line imagined, the speed vectors of it can be determined as the Figure (2.4). According to those vectors, equations (2.3) and (2.4) can be obtained.

$$\dot{x} = v_x \cos\alpha - v_y \sin\alpha \tag{2.3}$$

$$\dot{y} = v_y \cos\alpha + v_x \sin\alpha \tag{2.4}$$

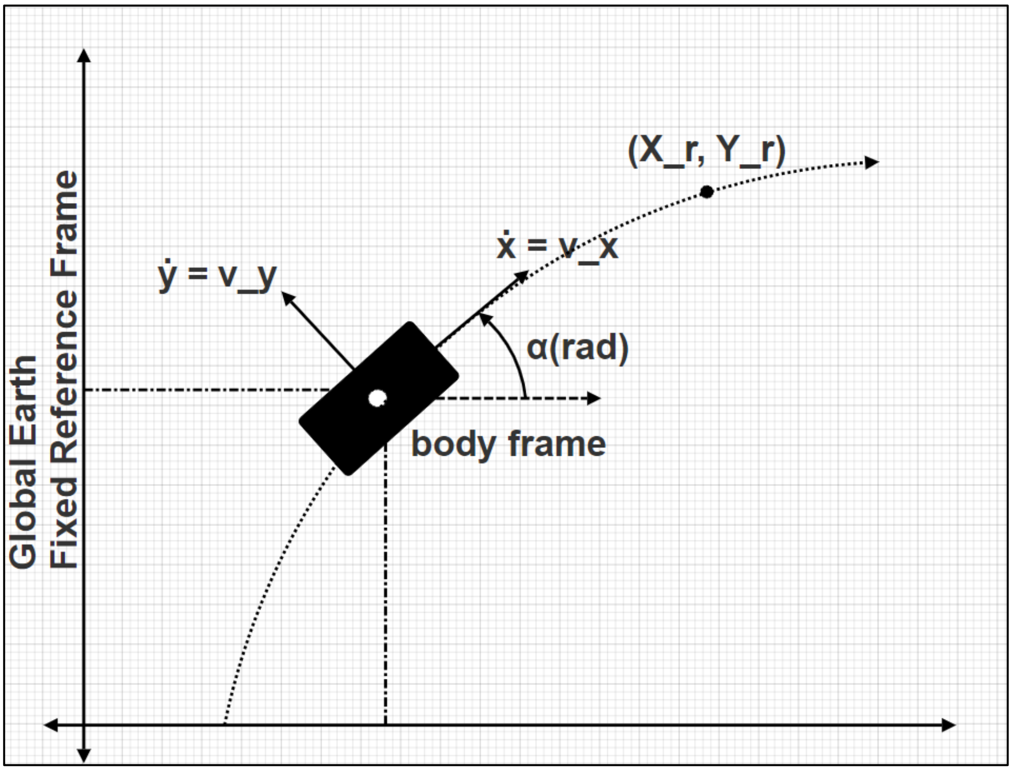


Figure 2.4: Determination of Speed Vectors of Moving Vehicle and A Random Point On Its Trajectory

Those equations are highly important for the determination of equation of motion and state space representation of the vehicle.

Additionally, vehicle's error equations can be derived as equations (2.5) and (2.6) according to reference points.

$$e_y = Y_R - y \tag{2.5}$$

$$e_x = X_R - x \tag{2.6}$$

So, the control schematics of lane changing manoeuvre can be represented as Figure (2.5).

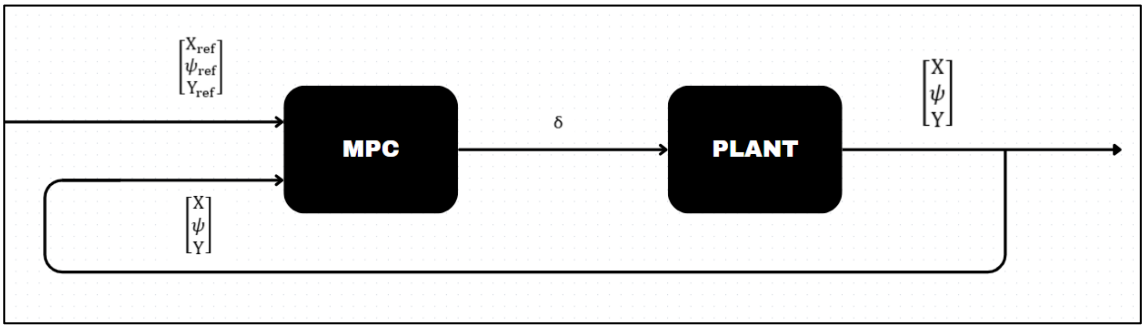

$\begin{bmatrix} X_{ref} \\ \psi_{ref} \\ Y_{ref} \end{bmatrix}$
MPC
δ
PLANT
$\begin{bmatrix} X \\ \psi \\ Y \end{bmatrix}$
$\begin{bmatrix} X \\ \psi \\ Y \end{bmatrix}$

Figure 2.5: Block Diagram of Lane Changing Manoeuvre

In the Figure (2.3), it can be seen that lane changing vehicle keeps its movement in x axis while it has constant longitudinal velocity. If the tyre slips are ignored, lane changing manoeuvre has no effect on longitudinal velocity. Therefore, final look of the block diagram can be represented as Figure (2.6). δ

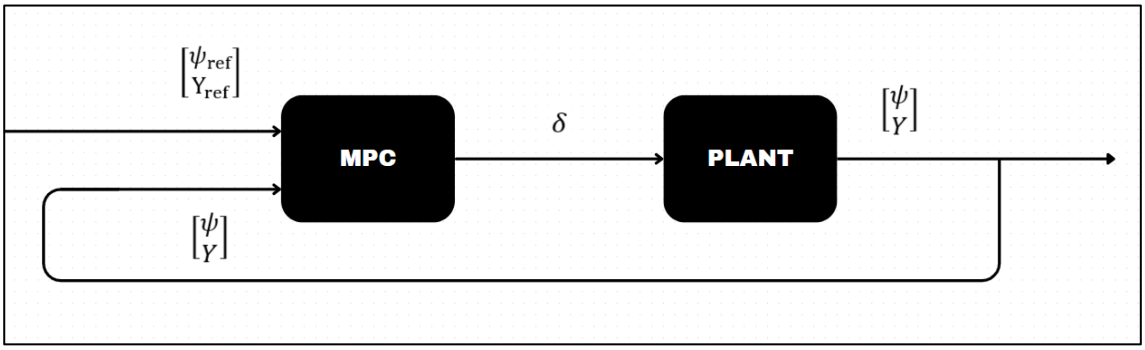

$\begin{bmatrix}\psi_{\text{ref}}\\ Y_{\text{ref}}\end{bmatrix}$
MPC
δ
PLANT
$\begin{bmatrix}\psi\\ Y\end{bmatrix}$
$\begin{bmatrix}\psi\\ Y\end{bmatrix}$

Figure 2.6: Block Diagram of Lane Changing Manoeuvre with Constant Longitudinal Movement

2.4 Vehicle Modelling and Determination of Motion Equations

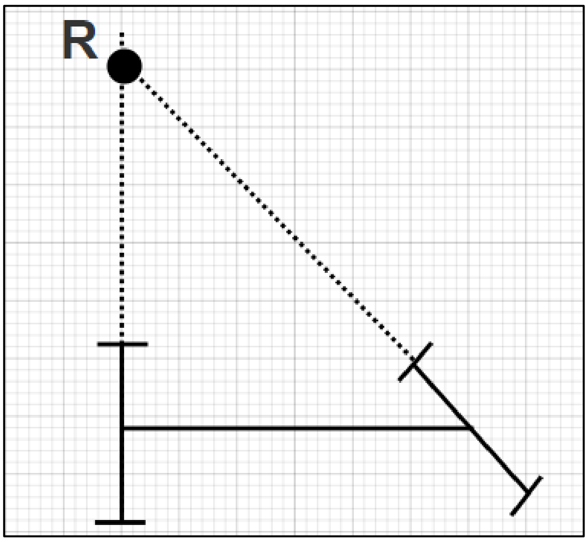


Figure 2.7: Whole Axle Steering and Centre of Rotation

The usage of four-wheel vehicles in modern life is integral for personal and societal mobility. Cars, trucks, and SUVs provide a convenient and efficient means of transportation, facilitating daily commutes, business activities, and leisure travel. They play a crucial role in connecting people to workplaces, schools, and recreational destinations, contributing to the functionality of urban and rural environments alike. Additionally, four-wheel vehicles are essential for transporting goods, supporting commerce, and sustaining economic activities. Despite challenges related to environmental impact and congestion, the widespread use of four-wheel vehicles remains a cornerstone of contemporary life, offering unparalleled flexibility and accessibility in meeting diverse transportation needs. The four-wheel vehicles must be able to perform turns and lane changes. In such actions, the entire axle or only the wheels rotations can be assumed. When vehicles turn, the rear and front axles intersect axially at a point called the centre of rotation, can be seen in Figure (2.7).

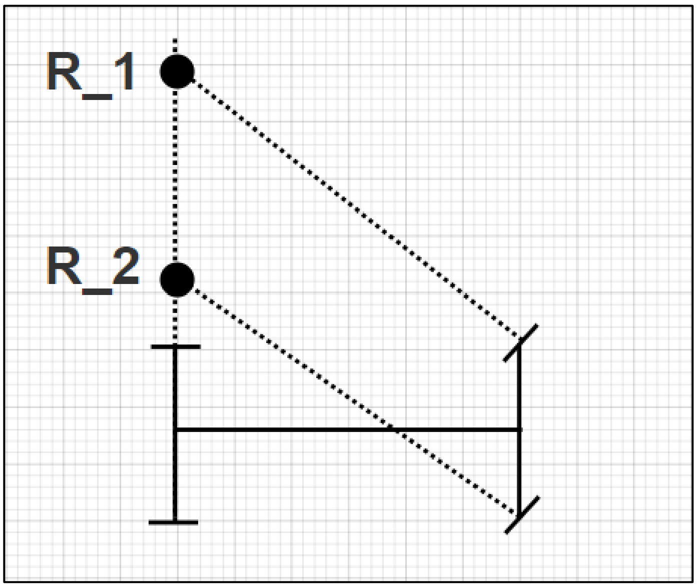


Figure 2.8: Modern Vehicle Steering Representation

Contrary to the Figure (2.7), in the modern vehicles, only the wheels perform the rotation, not the entire axle. But in that case, as in the Figure (2.8) there are two different centre of rotations are created. For this reason, one of the wheels may slip the other, or both may slip. This can pose a major dynamic problem.

To prevent the mentioned wheel slippage problem, in modern cars the two wheels do not rotate at the same angle. This principle is called Ackermann Steering Principle. Its general solution to this problem is visualised in Figure (2.9) [6].

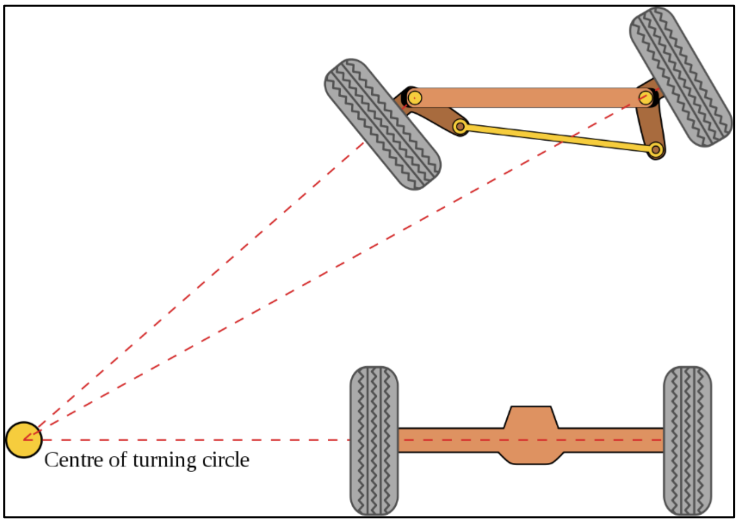


Figure 2.9: Ackermann Steering Geometry [7]

As can be seen in the examples, expressing vehicle dynamics mathematically is an extremely difficult task. Number of simplifications are inevitable in creating the plant model, which is very important for the controller design phase. At this point, the bicycle model comes to the rescue. Very often, controller designers use simplified versions of cars. That is called bicycle model. Studies and many real-life scenarios proved that bicycle model represents vehicle dynamics well enough to design controllers.

2.4.1 Bicycle Model

The term "bicycle model" in the context of vehicle dynamics typically refers to a simplified mathematical representation used for analysing and simulating the behaviour of a vehicle. In this model, a vehicle's dynamics are approximated by considering it as a single-track vehicle, similar to a bicycle. The bicycle model represents the vehicle's motion with essential parameters like mass, wheelbase, and tire characteristics. It simplifies complex vehicle dynamics into a form that is easier to analyse and allows for the prediction of the vehicle's response to steering inputs and external forces. The bicycle model is commonly employed in control systems and vehicle dynamics research to develop algorithms for stability control, path planning, and other advanced driver-assistance systems. In this project, it was used for the lane changing manoeuvre, which can be considered as a driver assistance system. General representation of bicycle model can be seen in Figure (2.10). [7]

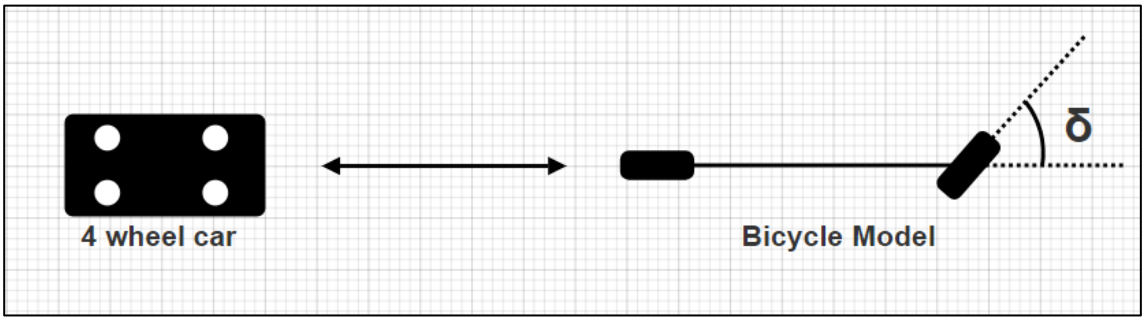


Figure 2.10: General Representation of Four-Wheel Car and Bicycle Model

Since this project focuses on the lane changing manoeuvre, the vectoral depiction of the bicycle model at any moment of turning looks like the Figure (2.11).

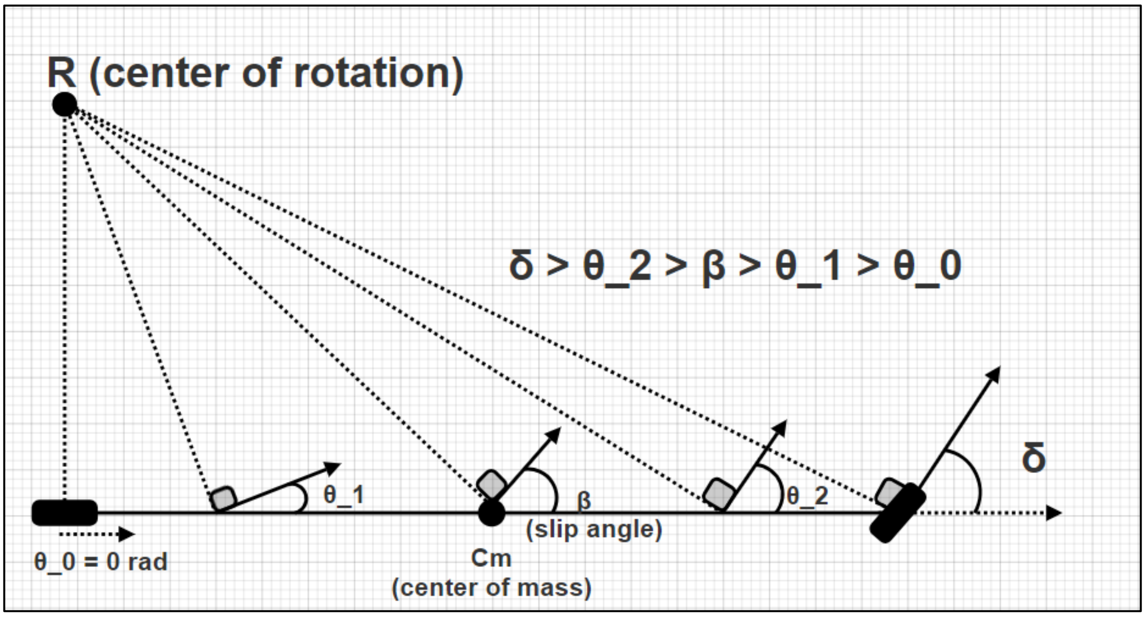


Figure 2.11: Bicycle Model's Comparison of Rotation Angles At Random Points

Perpendicular lines drawn from the centre of rotation to random points of the bicycle model correspond to the motions that the vehicle is subjected to in the turning manoeuvre.

2.4.2 Equations of Motion

The equation of motion is pivotal in control system design as it serves as the mathematical foundation for understanding, modelling, and analysing the dynamic behaviour of a system. This equation establishes the relationship between system inputs, outputs, and internal states, guiding engineers in creating accurate models that are essential for designing effective control strategies. Through analysis of the equation of motion, designers can assess system stability, transient response, and steady-state behaviour, influencing controller design decisions. The equation also facilitates simulation, enabling the validation of control strategies and the assessment of system performance under various conditions. In essence, the equation of motion is a fundamental tool that informs control system design, optimization, and robustness analysis, ensuring the reliable and efficient operation of the controlled system. Since only lateral movements are desired to be controlled by the MPC controller in this project, it will be sufficient to obtain the equations of motion only for lateral movements. The force vectors that the bicycle model is exposed to while in a rotation action and the components of the equations of motion that need to be obtained are as shown in Figure (2.12). [8]

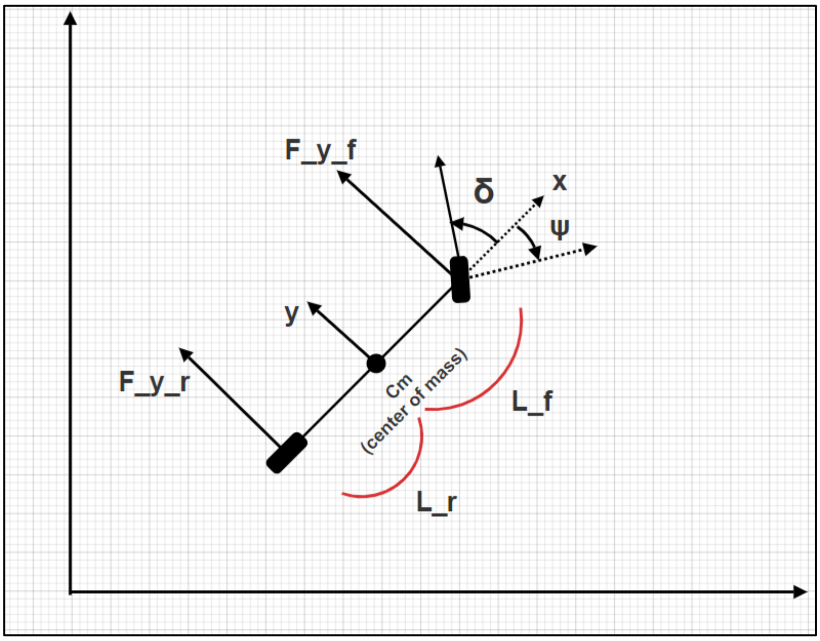


Figure 2.12: Lateral Force Vectors And Equations of Motion Elements Acting On The Bicycle Model

2.4.2.1 Lateral Direction Equations of Motion

Based on Figure (2.12), the following equations can be written.

$$\sum F = m \cdot \alpha \tag{2.7}$$

$$\sum M = I \cdot \ddot{\psi} \tag{2.8}$$

$$F_{yr} + F_{yf} = \mathrm{m} \cdot \alpha_y \tag{2.9}$$

$$F_{yr} \cdot \ell_r - F_{yf} \cdot \ell_f = I \cdot \ddot{\psi} \tag{2.10}$$

Centripetal Acceleration (α_Y):

Centripetal acceleration results from displacement at the centre of the vehicle. This displacement vector is shown as “β” in the Figure (2.11). Specific to lane changing problems, the angles the vehicle is exposed to are very small. Therefore, some assumptions need to be made to simplify the bicycle model. Therefore, calculations should be continued as if the slip angle expressed by “β” at the centre of the vehicle is very close to zero. [9]

$$\alpha_y = \frac{\vartheta^2}{R} \tag{2.11}$$

$$\frac{\vartheta^2}{R} = \frac{\ddot{x}}{R} \tag{2.12}$$

$$\dot{\psi} = \frac{\dot{x}}{R} \tag{2.13}$$

$$\alpha_y = \frac{\vartheta^2}{R} = \frac{\dot{x^2}}{R} = \frac{\dot{x}\dot{\psi}R}{R} = \dot{x} \cdot \dot{\psi} \tag{2.14}$$

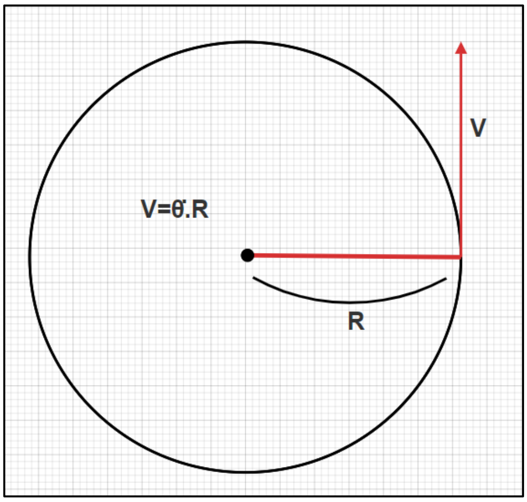


Figure 2.13: Circular Motion’s Visualization [9]

2.4.2.2 Tyre Modelling Equations of Motion

Tyre modelling is an important topic for controller design and the bicycle model in the context of vehicle dynamics because it provides an accurate representation of how the vehicle interacts with the road surface. The tire is the primary contact point between the vehicle and the road, and its behaviour significantly influences the overall dynamics of the system. Although it is handled in a greatly simplified manner since road surfaces are not considered in this project, the wheel model occupies an important place in the equations of motion. By incorporating a detailed tire model into the bicycle model, engineers can better simulate and understand the complex interactions involving tire forces, including lateral, longitudinal, and vertical forces. This detailed representation is essential for designing effective control strategies, as it allows for more accurate predictions of the vehicle's response to steering inputs, accelerations, and braking forces. The wheels of a turning vehicle change their positions on the fixed reference frame as shown in the Figure (2.14). [7]

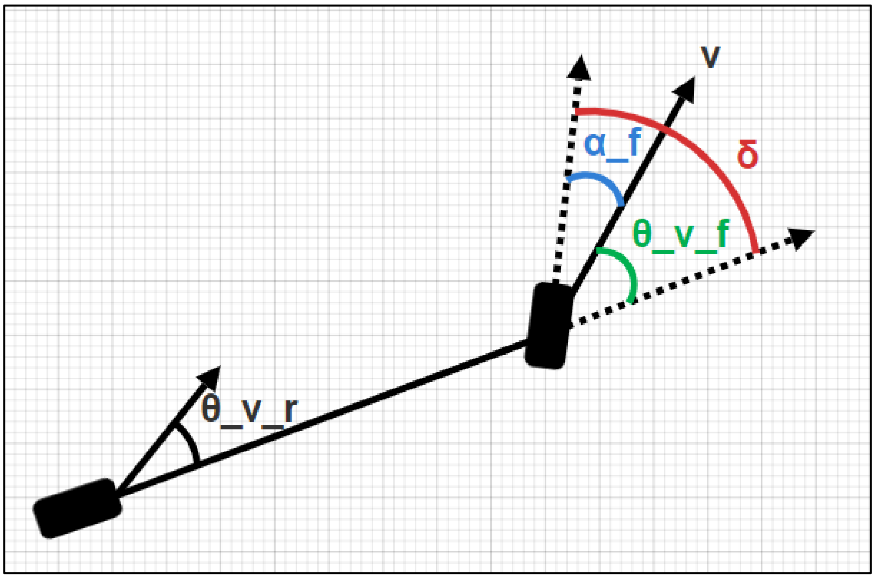


Figure 2.14: Displacement Angles of The Wheels of A Turning Vehicle

V value in the Figure (2.14) stands for vehicle's actual movement direction due to tyre slip and deformation. The α_f stands for front wheel slip angle while α_r stands for rear wheel slip angle.

$$\alpha_f = \delta - \theta_{vf} \tag{2.15}$$

$$\alpha_r = 0 - \theta_{vr} = -\theta_{vr} \tag{2.16}$$

The lateral forces occurring on the wheels of the turning vehicle are as in equations (2.17) and (2.18). Although the bicycle model is used for simplification in these equations, the equations are multiplied by two for two front and two rear wheels, appropriate for a four-wheeled vehicle. Also, C_α stands for cornering stiffness with unit of $\left[\frac{N}{rad}\right]$. [10]

$$F_{yf} = 2 \cdot C_{\alpha f} \cdot \alpha_f \tag{2.17}$$

$$\mathrm{F_{yr}} = 2 \cdot \mathrm{C_{\alpha r}} \cdot \mathrm{\alpha_r} \tag{2.18}$$

$$\tan(\theta_{\mathrm{vf}}) = \frac{\dot{y} + \ell_f \cdot \dot{\psi}}{\dot{x}} \tag{2.19}$$

$$\tan(\theta_{\mathrm{vr}}) = \frac{\dot{y} - \ell_r \cdot \dot{\psi}}{\dot{x}} \tag{2.20}$$

Small Angle Approximation for Lane Change:

The manoeuvre of changing lanes was previously discussed in terms of the bicycle model moving with small angles. This movement with small angles facilitates ease of calculation for the rear and front wheel slip angles, as mentioned in the preceding section. The tangent function yields result similar to the y=x function for small angles. Therefore, it is possible to simplify equations (2.19) and (2.20) as in (2.21) and (2.22). [9]

$$\theta_{vf} = \frac{\dot{y}+\ell_f \cdot \dot{\psi}}{\dot{x}} \tag{2.21}$$

$$\theta_{vr} = \frac{\dot{y}-\ell_r \cdot \dot{\psi}}{\dot{x}} \tag{2.22}$$

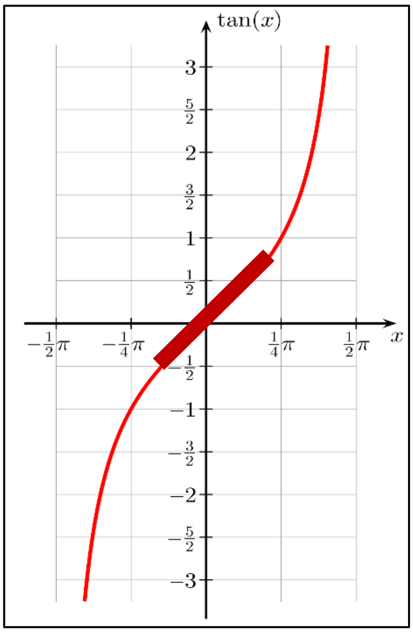


Figure 2.15: Tangent Function

2.5 Vehicle's State Space LTI Model

A state space model can be created with the equations obtained in the previous sections. The State Space model representing the plant model of the project is generally focused on two key equations. These are the angular and linear acceleration equations (2.23) and (2.24).

$$\ddot{y} = \left(\frac{2 \cdot C_{\alpha f}}{m}\right) \cdot \delta - \left(\frac{2 \cdot C_{\alpha f} + 2 \cdot C_{\alpha r}}{\dot{x} \cdot m}\right) \cdot \dot{y} - \left(x - \frac{2 \cdot C_{\alpha f} - 2 \cdot C_{\alpha r} \cdot \ell_r}{\dot{x} \cdot m}\right) \cdot \dot{\psi} \tag{2.23}$$

$$\ddot{\psi} = \left(\frac{\ell_f + 2 \cdot C_{\alpha f}}{I}\right) \cdot \delta - \left(\frac{\ell_f \cdot 2 \cdot C_{\alpha f} + \ell_r 2 \cdot C_{\alpha r}}{\dot{x} \cdot I}\right) \cdot \dot{y} - \left(\frac{{\ell_f}^2 \cdot 2 \cdot C_{\alpha f} - {\ell_r}^2 \cdot 2 \cdot C_{\alpha r}}{\dot{x} \cdot I}\right) \cdot \dot{\psi} \tag{2.24}$$

The initial form of the state space model is as (2.25). To enable this model to produce the outputs in Figure (2.6), a state augmentation process needs to be applied.

$$\begin{bmatrix} \ddot{y} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} -\left(\frac{2 \cdot C_{\alpha f} + 2 \cdot C_{\alpha r}}{\dot{x} \cdot m}\right) & -\left(x - \frac{2 \cdot C_{\alpha f} - 2 \cdot C_{\alpha r} \cdot \ell_r}{\dot{x} \cdot m}\right) \\ -\left(\frac{\ell_f \cdot 2 \cdot C_{\alpha f} + \ell_r 2 \cdot C_{\alpha r}}{\dot{x} \cdot I}\right) & -\left(\frac{{\ell_f}^2 \cdot 2 \cdot C_{\alpha f} - {\ell_r}^2 \cdot 2 \cdot C_{\alpha r}}{\dot{x} \cdot I}\right) \end{bmatrix} \begin{bmatrix} \dot{y} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} \left(\frac{2 \cdot C_{\alpha f}}{m}\right) \\ \left(\frac{\ell_f + 2 \cdot C_{\alpha f}}{I}\right) \end{bmatrix} \delta \tag{2.25}$$

The following equations should be used to add the required states according to the block diagram.

$$\dot{\psi} = \dot{\psi}$$

$$\dot{Y} = \dot{y} \cdot \cos \psi + \dot{x} \cdot \sin \psi \tag{2.26}$$

Equation (2.26) causes the state space model to lose its LTI (Linear Time-Invariant) property because one of its variables is dependent on a change of ψ (yaw angle). This implies that the state space system becomes time-varying. However, control system designers often use simplification methods to overcome such issues. By considering the small yaw angle values required for the lane-changing manoeuvre, we can use the simplification action in Figure (2.16) as a basis for addressing this problem.

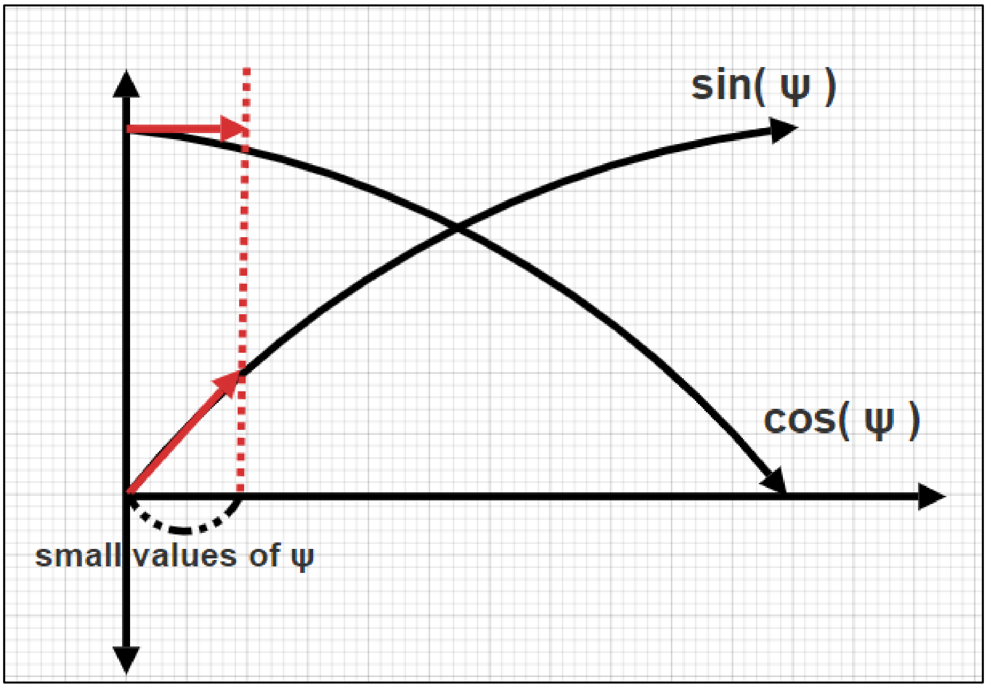


Figure 2.16: sin(ψ) and cos(ψ) Functions for Small Values of ψ

As seen in Figure (2.16), for small psi values, the cosine function behaves like y=1, while the sine function behaves like y=x. Considering these simplifications, equation (2.27) is obtained.

$$\dot{Y} = \dot{y} \cdot 1 + \dot{x} \cdot \psi \tag{2.27}$$

In the light of equations (2.25) and (2.27), the state space LTI model is obtained in its final form, as shown in the (2.28) and (2.29).

$$\begin{bmatrix} \ddot{y} \\ \dot{\psi} \\ \ddot{\psi} \\ \dot{Y} \end{bmatrix} = \begin{bmatrix} -\left(\frac{2 \cdot C_{\alpha f} + 2 \cdot C_{\alpha r}}{\dot{x} \cdot m}\right) & 0 & -\left(x - \frac{2 \cdot C_{\alpha f} - 2 \cdot C_{\alpha r} \cdot \ell_r}{\dot{x} \cdot m}\right) & 0 \\ 0 & 0 & 1 & 0 \\ -\left(\frac{\ell_f \cdot 2 \cdot C_{\alpha f} + \ell_r 2 \cdot C_{\alpha r}}{\dot{x} \cdot I}\right) & 0 & -\left(\frac{{\ell_f}^2 \cdot 2 \cdot C_{\alpha f} - {\ell_r}^2 \cdot 2 \cdot C_{\alpha r}}{\dot{x} \cdot I}\right) & 0 \\ 1 & \dot{x} & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{y} \\ \psi \\ \dot{\psi} \\ Y \end{bmatrix} + \begin{bmatrix} \left(\frac{2 \cdot C_{\alpha f}}{m}\right) \\ 0 \\ \left(\frac{\ell_f + 2 \cdot C_{\alpha f}}{I}\right) \\ 0 \end{bmatrix} \delta \tag{2.28}$$

$$\begin{bmatrix} \psi \\ Y \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{y} \\ \psi \\ \dot{\psi} \\ Y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \delta \tag{2.29}$$

If the vehicle parameters on Table 2.1 are substitute in the equations (2.28) and (2.29) equation (2.30) and (2.31) are obtained. They are the final look of the state space model of the bicycle model.

Table 2.1: Vehicle Model Parameters

Parameter Name	Parameter Value
Vehicle mass (m)	**1575 (kg)**
Vehicle's yaw moment of inertia (I_z)	**2875 (kg.m^2)**
Longitudinal distance from centre of gravity to front tire (l_f)	**1.2 (m)**
Longitudinal distance from centre of gravity to rear tire (l_r)	**1.6 (m)**
Cornering stiffness of front tyre (C_αf)	**19000 (N/rad)**
Cornering stiffness of rear tyre (C_αr)	**33000 (N/rad)**
Constant longitudinal speed (ẋ)	**15 (m/s)**

$$\begin{bmatrix} \ddot{y} \\ \dot{\psi} \\ \ddot{\psi} \\ \dot{Y} \end{bmatrix} = \begin{bmatrix} -1.65 & 0 & -38.94 & 0 \\ 0 & 0 & 1 & 0 \\ -0.52 & 0 & -1.94 & 0 \\ 1 & 15 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{y} \\ \psi \\ \dot{\psi} \\ Y \end{bmatrix} + \begin{bmatrix} 24.12 \\ 0 \\ 15.86 \\ 0 \end{bmatrix} \delta \tag{2.30}$$

$$\begin{bmatrix} \psi \\ Y \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{y} \\ \psi \\ \dot{\psi} \\ Y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \delta \tag{2.31}$$

2.6 Transformation of Continuous State Space LTI Model to the Discrete State Space LTI Model.

MPC controllers work sample by sample principally. Therefore, MPC controllers are called as discrete time controllers. MPC controllers are work as computers which try to minimize the cost function at each sample and give best possible inputs to the plant model. For this reason, the continuous time system obtained in section 2.5 must be converted to discrete time [9].

Continuous-Time State Space LTI System:

$$\dot{x}(t) = \mathrm{A} \cdot \mathrm{x(t)} + \mathrm{B} \cdot \delta(\mathrm{t}) \tag{2.30}$$

$$\dot{y}(t) = C \cdot x(t) + D \cdot \delta(t) \tag{2.31}$$

Discrete-Time State Space LTI System:

$$x_{k+1} = A_d \cdot x_k + B_d \cdot \delta_k \tag{2.32}$$

$$y_k = C_d \cdot x_k + D_d \cdot \delta_k \tag{2.33}$$

A continuous-time state space LTI system can be made discrete-time as follows.

$$\dot{x} = \frac{\vec{x}_{k+1} - \vec{x}_k}{T_s} \tag{2.34}$$

$$\frac{\vec{x}_{k+1} - \vec{x}_k}{T_s} = A \cdot x_k + \mathrm{B} \cdot \delta_k \tag{2.35}$$

$$\vec{x}_{k+1} = \vec{x}_k + A \cdot T_s \cdot \vec{x}_k + \mathrm{B} \cdot T_s \cdot \delta_k \tag{2.36}$$

$$\vec{x}_{k+1} = (I + A \cdot T_s)\vec{x}_k + \mathrm{B} \cdot T_s \cdot \delta_k \tag{2.37}$$

$$A_d = (I + A \cdot T_s) \;\&\; B_d = (\mathrm{B} \cdot T_s) \tag{2.38}$$

The state vector with horizon period with five samples $(H_z = 5)$ can be written based on the newly obtained discrete-time state space LTI model.

$$\vec{x}_k = A^k \cdot \vec{x}_0 + [A^{k-1}B \quad A^{k-2}B \cdots B][\delta_{k-1}] \tag{2.39}$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} A^1 \\ A^2 \\ A^3 \\ A^4 \\ A^5 \end{bmatrix} \cdot \vec{x}_0 + \begin{bmatrix} B & 0 & 0 & 0 & 0 \\ AB & B & 0 & 0 & 0 \\ A^2B & AB & B & 0 & 0 \\ \vdots & \cdots & \cdots & B & 0 \\ A^4B & \cdots & \cdots & \cdots & B \end{bmatrix} \begin{bmatrix} \delta_0 \\ \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix} \tag{2.40}$$

As a result of discrete-time processing efforts, the block diagram is transformed into the state shown in Figure (2.18). Additionally, the block diagram can be simplified as in Figure (2.18). This block diagram is the MPC controller diagram, which is familiar from various toolboxes.

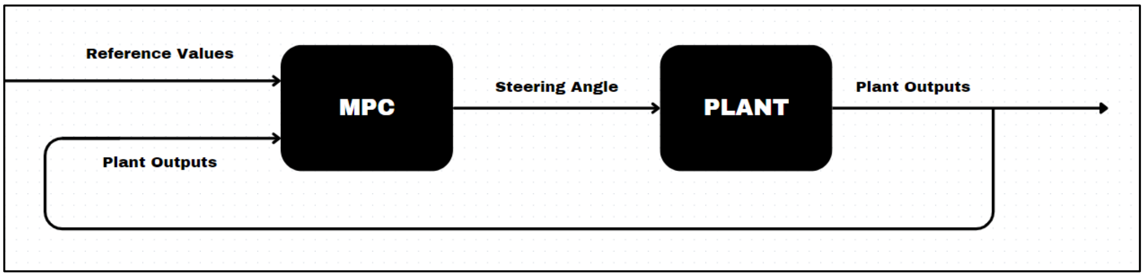


Figure 2.17: The Conventional MPC Block Diagram Where The Discrete-Time Compensator is Included Within The MPC Controller

SECTION 3

CONTROLLER DESIGN

The cost function is crucial in Model Predictive Control (MPC) and controller design as it quantifies the performance objectives and trade-offs the controller seeks to achieve. The cost function incorporates the system's dynamics, constraints, and desired behaviours, allowing the MPC algorithm to optimize future control actions over a prediction horizon. By formulating a suitable cost function, designers can effectively balance competing control objectives such as tracking a reference trajectory, minimizing control effort, avoiding constraints violations, and ensuring stability.

3.1 The Cost Function

The MPC controller iteratively solves an optimization problem using the cost function, adjusting control inputs at each step based on real-time measurements. This predictive nature enables MPC to handle complex systems and address time-varying dynamics, making it a powerful tool for applications in process control, robotics, and autonomous systems. The cost function for this project is shown in the equation in (3.1).

$$J = \frac{1}{2} \cdot \vec{e}_{k+n}^{T} \cdot S \cdot \vec{e}_{k+n} + \frac{1}{2} \sum_{i=0}^{n-1} \left[\vec{e}_{k+i}^{T} \cdot Q \cdot \vec{e}_{k+i} + \vec{\delta}_{k+i}^{T} \cdot R \cdot \vec{\delta}_{k+i} \right] \tag{3.1}$$

In section 2, horizon period was used as $(H_z = 5)$. Therefore equation (3.1) can be rearrangeable as (3.2).

$$J = \frac{1}{2} \cdot \vec{e}_{k+5}^{T} \cdot S \cdot \vec{e}_{k+5} + \frac{1}{2} \sum_{i=0}^{4} \left[\vec{e}_{k+i}^{T} \cdot Q \cdot \vec{e}_{k+i} + \vec{\delta}_{k+i}^{T} \cdot R \cdot \vec{\delta}_{k+i} \right] \tag{3.2}$$

The S, Q and R matrices in the cost function shown in (3.1) and (3.2) are weight matrices used to prioritize the error and rotation angle. These matrices are diagonal in nature.

3.2 Augmented Discretised State Space LTI Model

When performing lane changes, calculations can be made for an additional four samples based on the horizon period for each sample individually. Since these calculations are included in the cost function, they are separately computed for the error and the steering angle. However, aggressive changes in the steering angle may occur due to changes in the error vector. To mitigate this issue, the cost function and the MPC controller shall work with the input "$\Delta\delta$". This adjustment helps prevent aggressive steering changes associated with error fluctuations during lane-changing manoeuvres. In addition, it is guaranteed that both the error and the vehicle's turning angle are minimum. So, the final form of the MPC controller can be seen in Figure (3.1).

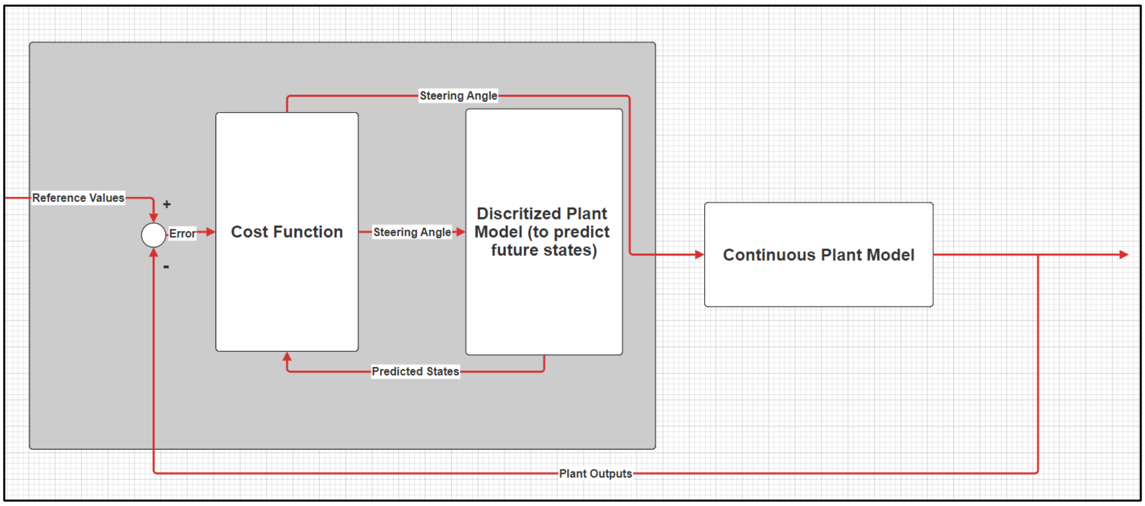


Figure 3.1: Final form of MPC with the most recent addition to the cost function

To obtain the block diagram in Figure (3.1), additional augmentations shall be done.

$$\delta_{k+1} = \delta_k + \delta_{k-1} \tag{3.3}$$

The equation in (3.3) determines the change in turn angle. In that equation the "δ_{k-1}" stands for the new extra state and the "δ_k" stands for the control input. So, expanded discretised LTI state space representation can be seen in (3.4), (3.5) and (3.6).

$$\vec{x}_{k+1} = A \cdot \vec{x}_k + \mathrm{B} \cdot \delta_k = A \cdot \vec{x}_k + \mathrm{B} \cdot (\delta_{k-1} - \delta_k) \tag{3.4}$$

$$\vec{x}_{k+1} = A \cdot \vec{x}_k + \mathrm{B} \cdot \delta_{k-1} + \mathrm{B} \cdot \delta_k \tag{3.5}$$

$$\delta_k = 0 \cdot \delta_{k-1} + \delta_k \tag{3.6}$$

$$\begin{bmatrix} \vec{x}_{k+1} \\ \delta_k \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \vec{x}_k \\ \delta_{k-1} \end{bmatrix} + \begin{bmatrix} B \\ I \end{bmatrix} \delta_k \tag{3.7}$$

$$[\vec{y}_k] = [C \quad 0] \begin{bmatrix} \vec{x}_k \\ \delta_{k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \delta_k \tag{3.8}$$

Following definitions can be made as a result of equations between (3.5) to (3.8).

$$\tilde{x}_{k+1} = \begin{bmatrix} \vec{x}_{k+1} \\ \delta_k \end{bmatrix} \ \& \ \tilde{A} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \ \& \ \tilde{B} = \begin{bmatrix} B \\ I \end{bmatrix} \ \& \ \tilde{C} = [C \quad 0] \tag{3.9}$$

In conclusion, new state vector is obtained as $[\dot{y} \quad \psi_k \quad \dot{\psi}_k \quad Y_k \quad \delta_{k-1}]^T$. Then, the new augmented discretised state space LTI is obtained as equations (3.11) and (3.12).

$$\tilde{x}_{k+1} = \tilde{A} \cdot \widetilde{\vec{x}_k} + \tilde{B} \cdot \delta_k \tag{3.11}$$

$$\vec{y}_k = \tilde{C} \cdot \widetilde{\vec{x}_k} \tag{3.12}$$

After the new augmented discrete-time state space LTI model is obtained, the cost function shall be expanded. Equation (3.13) can be obtained with the replacement of error variable with $(\vec{r}_{k+n} - \tilde{C} \cdot \widetilde{\vec{x}_{k+n}})$.

$$J = \frac{1}{2} \cdot (\vec{r}_{k+n} - \tilde{C} \cdot \widetilde{\vec{x}_{k+n}})^T \cdot S \cdot (\vec{r}_{k+n} - \tilde{C} \cdot \widetilde{\vec{x}_{k+n}}) + \frac{1}{2} \sum_{i=0}^{n-1} \Big[(\vec{r}_{k+i} - \tilde{C} \cdot \widetilde{\vec{x}_{k+\imath}})^T \cdot Q \cdot (\vec{r}_{k+i} - \tilde{C} \cdot \widetilde{\vec{x}_{k+\imath}}) + \vec{\delta}_{k+i}^T \cdot R \cdot \vec{\delta}_{k+i} \Big] \tag{3.13}$$

With further expansion, (3.14) can be obtained.

$$J = \frac{1}{2} \cdot \vec{r}_{k+n}{}^T \cdot S \cdot \vec{r}_{k+n} - \vec{r}_{k+n}{}^T \cdot S \cdot \tilde{C} \cdot \widetilde{\vec{x}_{k+n}} + \frac{1}{2} \cdot \widetilde{\vec{x}_{k+n}}^T \cdot \tilde{C}^T \cdot S \cdot \tilde{C} \cdot \widetilde{\vec{x}_{k+n}} + \sum_{i=0}^{n-1} \Big[\frac{1}{2} \vec{r}_{k+i}{}^T \cdot Q \cdot \vec{r}_{k+i} - \vec{r}_{k+i}{}^T \cdot Q \cdot \tilde{C} \cdot \widetilde{\vec{x}_{k+\imath}} + \frac{1}{2} \cdot \widetilde{\vec{x}_{k+\imath}}^T \cdot \tilde{C}^T \cdot Q \cdot \tilde{C} \cdot \widetilde{\vec{x}_{k+\imath}} + \frac{1}{2} \vec{\delta}_{k+i}^T \cdot R \cdot \vec{\delta}_{k+i} \Big] \tag{3.14}$$

3.3 Mathematical Minimisation of Cost Function

Considering a quadratic polynomial as in figure (3.2), the equation of this polynomial should look like "$y = a(x - x_1)^2 + b$". For such polynomial, setting the b value to zero does not cause the minimum point to change, but it simplifies the operations to be performed with the polynomial. The polynomial in Figure (3.2) is similar to our cost function. For this reason, the final version of the cost function we obtained in equation (3.14) can be simplified further.

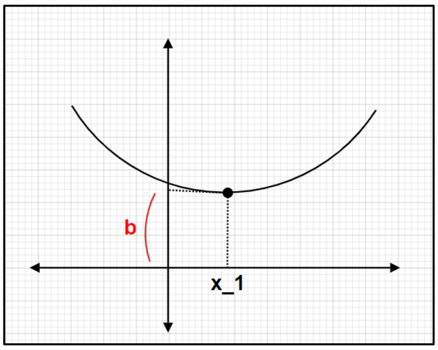


Figure 3.2: Random Quadratic Polynomial

Figure (3.3) shows the parts that will be simplified by setting them equal to zero.

$$\mathrm{J} = \overset{\text{equals to 0}}{\cancel{\frac{1}{2}\cdot \vec{r}_{k+n}{}^{\mathrm{T}} \cdot \mathrm{S} \cdot \vec{r}_{k+n}}} - \vec{r}_{k+n}{}^{\mathrm{T}} \cdot \mathrm{S} \cdot \tilde{C} \cdot \widetilde{\vec{x}_{k+n}} + \frac{1}{2} \cdot \widetilde{\vec{x}_{k+n}}{}^{\mathrm{T}} \cdot \tilde{C}^{\mathrm{T}} \cdot \mathrm{S} \cdot \tilde{C} \cdot \widetilde{\vec{x}_{k+n}} + \\ \sum_{\mathrm{i=0}}^{\mathrm{n\text{-}1}} \left[\overset{\text{equals to 0}}{\cancel{\frac{1}{2}\vec{r}_{k+\mathrm{i}}{}^{\mathrm{T}} \cdot \mathrm{Q} \cdot \vec{r}_{k+\mathrm{i}}}} \underbrace{- \vec{r}_{k+\mathrm{i}}{}^{\mathrm{T}} \cdot \mathrm{Q} \cdot \tilde{C} \cdot \widetilde{\vec{x}_{k+1}} + \frac{1}{2} \cdot \widetilde{\vec{x}_{k+1}}{}^{\mathrm{T}} \cdot \tilde{C}^{\mathrm{T}} \cdot \mathrm{Q} \cdot \tilde{C} \cdot \widetilde{\vec{x}_{k+1}}}_{\text{equals to 0 only in the first sample}} + \frac{1}{2}\Delta\vec{\delta}_{\mathrm{k+i}}^{\mathrm{T}} \cdot \mathrm{R} \cdot \right. \\ \left. \overrightarrow{\Delta\delta}_{\mathrm{k+i}}\right] \qquad (3.14)$$

Figure 3.3: Simplification Operation for Equation (3.14)

As a result, simplified cost function J' is obtained as in (3.15) for $(H_z = 5)$, which means for five samples.

$$J' = -\vec{r}_{k+n}{}^{T} \cdot S \cdot \tilde{C} \cdot \widetilde{\vec{x}_{k+n}} + \frac{1}{2} \cdot \widetilde{\vec{x}_{k+n}}{}^{T} \cdot \tilde{C}^{T} \cdot S \cdot \tilde{C} \cdot \widetilde{\vec{x}_{k+n}} + \left[\frac{1}{2}\Delta\vec{\delta}_{k+i}^{T} \cdot R \cdot \overrightarrow{\Delta\delta}_{k+i}\right]_{i=0} + \\ \left[-\vec{r}_{k+i}{}^{T} \cdot Q \cdot \tilde{C} \cdot \widetilde{\vec{x}_{k+\iota}} + \frac{1}{2} \cdot \widetilde{\vec{x}_{k+\iota}}{}^{T} \cdot \tilde{C}^{T} \cdot Q \cdot \tilde{C} \cdot \widetilde{\vec{x}_{k+\iota}} + \frac{1}{2}\Delta\vec{\delta}_{k+i}^{T} \cdot R \cdot \overrightarrow{\Delta\delta}_{k+i}\right]_{i=1} + \left[-\vec{r}_{k+i}{}^{T} \cdot \right. \\ \left. Q \cdot \tilde{C} \cdot \widetilde{\vec{x}_{k+\iota}} + \frac{1}{2} \cdot \widetilde{\vec{x}_{k+\iota}}{}^{T} \cdot \tilde{C}^{T} \cdot Q \cdot \tilde{C} \cdot \widetilde{\vec{x}_{k+\iota}} + \frac{1}{2}\Delta\vec{\delta}_{k+i}^{T} \cdot R \cdot \overrightarrow{\Delta\delta}_{k+i}\right]_{i=2} + \left[-\vec{r}_{k+i}{}^{T} \cdot Q \cdot \tilde{C} \cdot \right. \\ \left. \widetilde{\vec{x}_{k+\iota}} + \frac{1}{2} \cdot \widetilde{\vec{x}_{k+\iota}}{}^{T} \cdot \tilde{C}^{T} \cdot Q \cdot \tilde{C} \cdot \widetilde{\vec{x}_{k+\iota}} + \frac{1}{2}\Delta\vec{\delta}_{k+i}^{T} \cdot R \cdot \overrightarrow{\Delta\delta}_{k+i}\right]_{i=3} + \left[-\vec{r}_{k+i}{}^{T} \cdot Q \cdot \tilde{C} \cdot \widetilde{\vec{x}_{k+\iota}} + \frac{1}{2} \cdot \right. \\ \left. \widetilde{\vec{x}_{k+\iota}}{}^{T} \cdot \tilde{C}^{T} \cdot Q \cdot \tilde{C} \cdot \widetilde{\vec{x}_{k+\iota}} + \frac{1}{2}\Delta\vec{\delta}_{k+i}^{T} \cdot R \cdot \overrightarrow{\Delta\delta}_{k+i}\right]_{i=4} \qquad (3.15)$$

The reference, output, and input vectors present in the cost function should be generalized as arguments for five samples corresponding to the horizon period $(H_z = 5)$. This process can be carried out as equations (3.16), (3.17) and (3.18).

$$\vec{r}_G = \begin{bmatrix} \vec{r}_{k+1} \\ \vec{r}_{k+2} \\ \vec{r}_{k+3} \\ \vec{r}_{k+4} \\ \vec{r}_{k+5} \end{bmatrix} \qquad (3.16)$$

$$\widetilde{\vec{x}_G} = \begin{bmatrix} \widetilde{\vec{x}_{k+1}} \\ \widetilde{\vec{x}_{k+2}} \\ \widetilde{\vec{x}_{k+3}} \\ \widetilde{\vec{x}_{k+4}} \\ \widetilde{\vec{x}_{k+5}} \end{bmatrix} \qquad (3.17)$$

$$\overrightarrow{\Delta\delta_G} = \begin{bmatrix} \Delta\delta_k \\ \Delta\delta_{k+1} \\ \Delta\delta_{k+2} \\ \Delta\delta_{k+3} \\ \Delta\delta_{k+4} \end{bmatrix} \tag{3.18}$$

If the cost function is rewritten with the general matrices obtained, the equation in (3.19) is obtained.

$$J' = \tfrac{1}{2} \cdot \widetilde{\vec{x}_G}^{\,T} \cdot \begin{bmatrix} \tilde{C}^T \cdot Q \cdot \tilde{C} & 0 & \dots & 0 \\ 0 & \tilde{C}^T \cdot Q \cdot \tilde{C} & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \dots & 0 & \tilde{C}^T \cdot Q \cdot \tilde{C} \end{bmatrix} \cdot \widetilde{\vec{x}_G} - \vec{r}_G \cdot \begin{bmatrix} Q \cdot \tilde{C} & 0 & \dots & 0 \\ 0 & Q \cdot \tilde{C} & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \dots & 0 & S \cdot \tilde{C} \end{bmatrix} \cdot \widetilde{\vec{x}_G} + \tfrac{1}{2} \cdot \overrightarrow{\Delta\delta_k}^{T} \cdot \begin{bmatrix} R & 0 & \dots & 0 \\ 0 & R & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \dots & 0 & R \end{bmatrix} \cdot \overrightarrow{\Delta\delta_G} \tag{3.19}$$

The equation (3.19) can be rewritten as (3.20).

$$J' = \tfrac{1}{2} \cdot \widetilde{\vec{x}_G}^{\,T} \cdot \bar{\bar{Q}} \cdot \widetilde{\vec{x}_G} - \vec{r}_G \cdot \bar{\bar{T}} \cdot \widetilde{\vec{x}_G} + \tfrac{1}{2} \cdot \overrightarrow{\Delta\delta_k}^{T} \cdot \bar{\bar{R}} \cdot \overrightarrow{\Delta\delta_G} \tag{3.20}$$

The matrices belong to equation (2.40) which is in section two, can be redefined as (3.21).

$$\bar{\bar{A}} = \begin{bmatrix} A^1 \\ A^2 \\ A^3 \\ A^4 \\ A^5 \end{bmatrix} \; and \; \widetilde{\vec{x}_k} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} and \; \bar{\bar{C}} = \begin{bmatrix} B & 0 & 0 & 0 & 0 \\ AB & B & 0 & 0 & 0 \\ A^2B & AB & B & 0 & 0 \\ \vdots & \cdots & \cdots & B & 0 \\ A^4B & \cdots & \cdots & \cdots & B \end{bmatrix} and \; \overrightarrow{\Delta\delta_G} = \begin{bmatrix} \delta_0 \\ \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix} \tag{3.21}$$

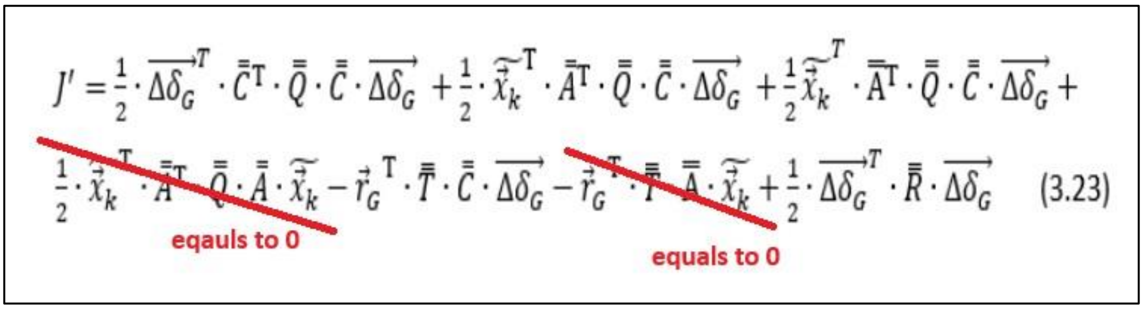
As a result of equations (3.20) and (3.21), (3.22) is obtained.

$$J' = \tfrac{1}{2} \cdot (\bar{\bar{C}} \cdot \overrightarrow{\delta_G} + \bar{\bar{A}} \cdot \widetilde{\vec{x}_k})^T \cdot \bar{\bar{Q}} \cdot (\bar{\bar{C}} \cdot \overrightarrow{\delta_G} + \bar{\bar{A}} \cdot \widetilde{\vec{x}_k}) - \vec{r}_G \cdot \bar{\bar{T}} \cdot (\bar{\bar{C}} \cdot \overrightarrow{\delta_G} + \bar{\bar{A}} \cdot \widetilde{\vec{x}_k}) + \tfrac{1}{2} \cdot \overrightarrow{\delta_G}^{T} \cdot \bar{\bar{R}} \cdot \overrightarrow{\delta_G} \tag{3.22}$$

J' can be expanded as in (3.23).

$$J' = \tfrac{1}{2} \cdot \overrightarrow{\delta_G}^{T} \cdot \bar{\bar{C}}^T \cdot \bar{\bar{Q}} \cdot \bar{\bar{C}} \cdot \overrightarrow{\delta_G} + \tfrac{1}{2} \cdot \widetilde{\vec{x}_k}^{\,T} \cdot \bar{\bar{A}}^T \cdot \bar{\bar{Q}} \cdot \bar{\bar{C}} \cdot \overrightarrow{\delta_G} + \tfrac{1}{2} \widetilde{\vec{x}_k}^{\,T} \cdot \bar{\bar{A}}^T \cdot \bar{\bar{Q}} \cdot \bar{\bar{C}} \cdot \overrightarrow{\delta_G} + \tfrac{1}{2} \cdot \widetilde{\vec{x}_k}^{\,T} \cdot \bar{\bar{A}}^T \cdot \bar{\bar{Q}} \cdot \bar{\bar{A}} \cdot \widetilde{\vec{x}_k} - \vec{r}_G^{\,T} \cdot \bar{\bar{T}} \cdot \bar{\bar{C}} \cdot \overrightarrow{\delta_G} - \vec{r}_G^{\,T} \cdot \bar{\bar{T}} \cdot \bar{\bar{A}} \cdot \widetilde{\vec{x}_k} + \tfrac{1}{2} \cdot \overrightarrow{\delta_G}^{T} \cdot \bar{\bar{R}} \cdot \overrightarrow{\delta_G} \tag{3.23}$$

If the equation obtained in (3.23) is simplified as in Figure (3.3), the J'' equation obtained as a result of this simplification is included in (3.24).



$$J' = \frac{1}{2}\cdot \overrightarrow{\Delta\delta_G}^T \cdot \bar{\bar{C}}^{\mathrm{T}} \cdot \bar{\bar{Q}} \cdot \bar{\bar{C}} \cdot \overrightarrow{\Delta\delta_G} + \frac{1}{2}\cdot \tilde{\vec{x}_k}^{\mathrm{T}} \cdot \bar{\bar{A}}^{\mathrm{T}} \cdot \bar{\bar{Q}} \cdot \bar{\bar{C}} \cdot \overrightarrow{\Delta\delta_G} + \frac{1}{2} \tilde{\vec{x}_k}^T \cdot \bar{\bar{A}}^{\mathrm{T}} \cdot \bar{\bar{Q}} \cdot \bar{\bar{C}} \cdot \overrightarrow{\Delta\delta_G} +$$

$$\underbrace{\cancel{\frac{1}{2}\cdot \vec{x}_k^{\mathrm{T}} \cdot \bar{\bar{A}}^{\mathrm{T}} \cdot \bar{\bar{Q}} \cdot \bar{\bar{A}} \cdot \tilde{\vec{x}_k}}}_{\text{eqauls to 0}} - \vec{r}_G^{\mathrm{T}} \cdot \bar{\bar{T}} \cdot \bar{\bar{C}} \cdot \overrightarrow{\Delta\delta_G} \underbrace{\cancel{- \vec{r}_G^{\mathrm{T}} \cdot \bar{\bar{T}} \cdot \bar{\bar{A}} \cdot \tilde{\vec{x}_k}}}_{\text{equals to 0}} + \frac{1}{2}\cdot \overrightarrow{\Delta\delta_G}^T \cdot \bar{\bar{R}} \cdot \overrightarrow{\Delta\delta_G} \qquad (3.23)$$

Figure 3.4: Simplification Operation for Equation (3.23)

$$J'' = \frac{1}{2}\cdot \overrightarrow{\delta_G}^T \cdot (\bar{\bar{C}}^T \cdot \bar{\bar{Q}} \cdot \bar{\bar{C}} + \bar{\bar{R}}) \cdot \overrightarrow{\delta_G} + \left[\tilde{\vec{x}_k}^T \cdot \vec{r}_G^{\,T}\right] \cdot \begin{bmatrix} \bar{\bar{A}}^T & \bar{\bar{Q}} & \bar{\bar{C}} \\ 0 & \bar{\bar{T}} & \bar{\bar{C}} \end{bmatrix} \cdot \overrightarrow{\delta_G} \qquad (3.24)$$

Equation (3.24) can be rearranged as in equation (3.25).

$$J'' = \frac{1}{2}\cdot \overrightarrow{\delta_g}^T \cdot \bar{\bar{H}} \cdot \overrightarrow{\delta_G} + \left[\tilde{\vec{x}_k}^T \cdot \vec{r}_G^{\,T}\right] \cdot \bar{\bar{F}}^T \cdot \overrightarrow{\delta_G} \qquad (3.25)$$

As a result, where the gradient of the simplest cost function J'' equals to zero is its minimum point as obtained in (3.26) and (3.27).

$$\nabla J'' = \bar{\bar{H}} \cdot \overrightarrow{\delta_G} + \bar{\bar{F}} \begin{bmatrix} \tilde{\vec{x}_k} \\ \vec{r}_G \end{bmatrix} = 0 \qquad (3.26)$$

$$\overrightarrow{\Delta\delta_G} = \bar{\bar{H}}^{-1} \cdot \bar{\bar{F}} \begin{bmatrix} \tilde{\vec{x}_k} \\ \vec{r}_G \end{bmatrix} \qquad (3.27)$$

The final look of the block diagram is as in Figure (3.5).

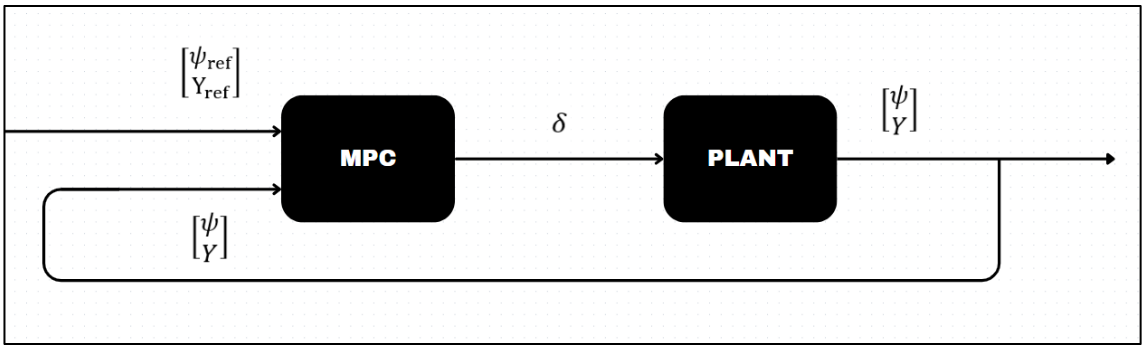

$\begin{bmatrix}\psi_{\mathrm{ref}} \\ \mathrm{Y}_{\mathrm{ref}}\end{bmatrix}$
MPC
δ
PLANT
$\begin{bmatrix}\psi \\ Y\end{bmatrix}$
$\begin{bmatrix}\psi \\ Y\end{bmatrix}$

Figure 3.5: Final look of the block diagram

SECTION 4

REALISTIC LIMITS, CONDITIONS AND CONSTRAINS TAKEN INTO CONSIDERATION IN THE DESIGN OF THE PROJECT

Designing a project for autonomous lane changing with MPC (Model Predictive Control) involves considering various realistic limits, conditions, and constraints. The enumerated tasks for consideration during the modelling process are delineated below.

4.1 Vehicle Dynamics

Bicycle Model: Chosen a bicycle model to simplify vehicle dynamics, it's important to acknowledge the limitations of this abstraction. Real-world vehicles have more complex dynamics, and ignoring certain factors may lead to inaccuracies.

Actuator Constraints: The physical limitations of the vehicle's steering and actuation systems shall be considered. Sudden or extreme manoeuvres may not be feasible or safe.

Sensor Limitations: The sensors (e.g., cameras, LiDAR, radar) have limitations in terms of range and resolution. The accuracy and reliability of the perception system depend on these factors, and they should be considered in the design.

4.2 Computational Constraints

Real-time Processing: The MPC algorithm needs to operate in real-time. Consider the computational resources required and ensure that the system can process data and make decisions within acceptable time frames.

Simulator Accuracy: The simulation environment accurately represents the real-world scenario shall be ensured. Simulations should cover a wide range of scenarios to validate the system's robustness.

Energy Consumption: The energy efficiency of the autonomous system shall be considered, especially if it involves electric vehicles. Performance and energy consumption should be balanced for further sustainability.

4.3 Additional Constraints

Human Interaction: Account for the possibility of human intervention. Define clear guidelines on how the autonomous system should handle scenarios where the driver decides to take control.

Weather Conditions: Consider the impact of adverse weather conditions such as heavy rain, snow, or fog on sensor performance and overall system reliability.

Legal and Liability: Address legal and liability concerns associated with autonomous vehicles. Clearly define responsibility in case of accidents or system failures.

Environment Constraints: The system should be robust to different road conditions such as wet, icy, or uneven surfaces. Changes in road friction can impact the performance of the lane-changing algorithm. Consideration of other vehicles on the road, their behaviour, and adherence to traffic rules is crucial. The system should be able to adapt to various traffic scenarios.

Safety Regulations: Shall be ensured that the autonomous system complies with local and international safety regulations and standards. This includes adherence to speed limits, signalling rules, and other traffic regulations.

Communication and Connectivity: If available, integrating V2V communication to enhance coordination with other vehicles on the road shall be considered. This can improve safety and efficiency during lane changes.

By carefully considering these factors, you can develop a robust and realistic autonomous lane-changing system that operates within defined boundaries and meets safety and regulatory standards.

4.4 Table of Benefits and Risks

The development and implementation of a Lane Changing Application with Discrete Time State Space LTI Bicycle Model and MPC Controller present a unique set of opportunities and challenges. In delving into the analysis of benefits and risks associated with this project, a broader spectrum of factors will be considered, encompassing health, manufacturability, ethical considerations, and security aspects.

In addition to addressing traditional aspects such as safety, traffic efficiency, and adaptability to dynamic conditions, this analysis aims to explore how the project might impact the well-being of drivers, the feasibility of manufacturing and deploying the solution across different vehicle platforms, ethical considerations related to decision-making algorithms, and security concerns to safeguard against unauthorized access or manipulation.

Table 4.1: Table of Some Benefits and Risks

Physical limitations, constraints, and boundaries	The simplified nature of the Discrete Time LTI Bicycle Model may introduce limitations in accurately representing certain complex vehicle dynamics. Consideration should be given to potential trade-offs between accuracy and computational efficiency.
Improved Safety	The computational demands associated with the Discrete Time LTI Bicycle Model and MPC may pose challenges, especially in real-time applications. Efforts should be made to optimize the algorithms for efficient execution.
Computational Complexity	The integration of a precise Discrete Time State Space LTI Bicycle Model and MPC controller can enhance the safety of lane-changing manoeuvres by considering predictive models and addressing uncertainties in real-time.
Engineering norms and standards	Industry standards for safety, performance, and interoperability are met by the Lane Changing Application by adherence to recognized engineering norms and standards.
Economic aspects	The optimized traffic flow and potential reduction in accidents may lead to economic benefits, including reduced travel times, lower fuel consumption, and decreased infrastructure maintenance costs.
Environmental aspects	Improved traffic flow can potentially reduce fuel consumption and emissions, contributing to a positive environmental impact by mitigating the carbon footprint associated with traffic-related activities.

Sustainability	By contributing to efficient traffic management, the Lane Changing Application may support sustainable transportation practices by minimizing environmental impacts associated with congestion and inefficient traffic flow.
Manufacturability	The modular design of the integrated model and controller can facilitate easier implementation and deployment across different vehicle platforms, contributing to manufacturability.
Health aspects	Improved traffic flow and reduced congestion may contribute to lower stress levels for drivers, potentially positively impacting their mental health during commuting.
Security aspects	Implementing robust cybersecurity measures is crucial to prevent unauthorized access or manipulation of the control system, ensuring the safety and security of the vehicles and their occupants.
Ethical aspects	Ethical considerations include addressing potential biases in decision-making algorithms, ensuring fair treatment of all road users, and transparent communication of the system's capabilities and limitations.

SECTION 5

RESULTS AND VALIDATIONS

5.1 MPC Parameters and Constraints Table

Control engineers play a pivotal role in designing systems that regulate and optimize the behaviour of dynamic processes, ranging from industrial machinery to advanced vehicle control systems. One of the key elements in ensuring the efficacy of these control systems is the process of tuning, a critical and iterative task undertaken by control engineers. Tuning involves adjusting the parameters of a control system to achieve desired performance, stability, and responsiveness. For Model Predictive Control (MPC), a cutting-edge control strategy widely employed in diverse applications, tuning is especially crucial. MPC inherently considers predictive models and optimizes control actions over a specified time horizon, making proper tuning essential to harness its full potential. Table 5.1 and Table 5.2 contains the constraints and the controller parameters of the discrete time MPC which is going to be compared with the continuous time MPC in the following sections.

Table 5.1 Constraints and Increments Table of the MPC 5.1

CONSTRAINTS TABLE					
I/O	Constraint Name	Min Value	Max Value	Min Rate	Max Rate
I	Steering Angle (rads)	-0,52	0,52	-0,17	0,17
O	Lateral Displacement (meters)	-10	10	-	-
O	Yaw Angle (rads)	-0,4	0,4	-	-

Table 5.2: Controller Parameters

Controller Parameters	Controller Parameter Values
Output Weights (Q)	$\begin{bmatrix} 100 & 0 \\ 0 & 10 \end{bmatrix}$
Final Horizon Output Weights (S)	$\begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$
Inputs Weight (R)	[100]
MPC Prediction Horizon (H_z)	5 samples (0.5 seconds)
MPC Control Horizon (Cz)	5 samples (0.5 seconds)
Sample Time (T_s)	0.1 seconds

5.2 Basic Lane Change Manoeuvres with Discrete Time MPC

5.2.1 Single Lane Change Manoeuvre

The car is predicted to change position based on the tangent function from its current position, assuming that it changes single lane. Using the "tanh ()" function as a reference to replicate the lateral movement from y=0 to y=3, Figure 5.1 was obtained.

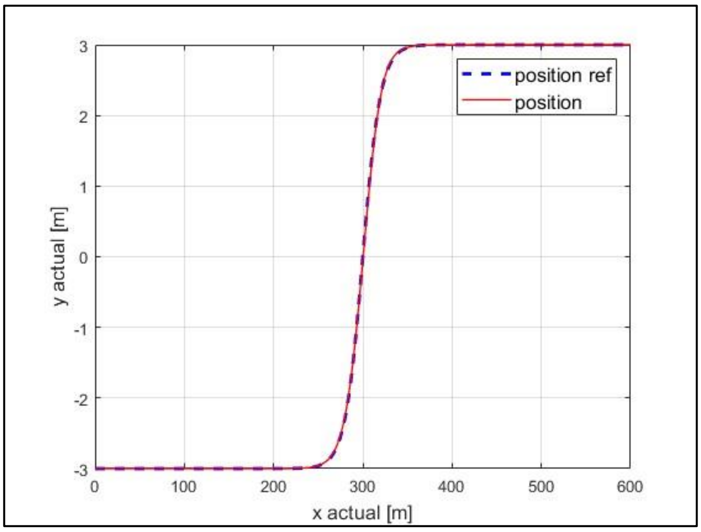


Figure 5.1: Single Lane Change Vehicle Position Comparison Between Reference and Actual Position

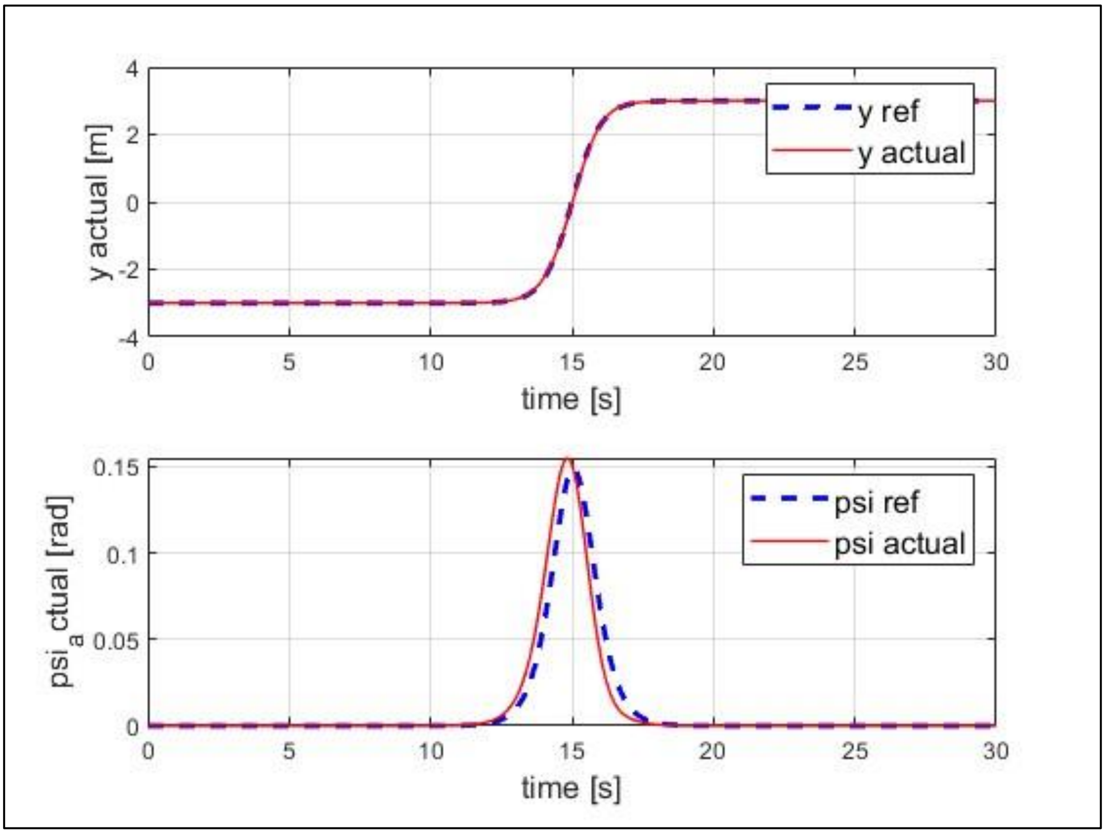


Figure 5.2: Single Lane Change MPC Inputs Comparison and Error Visualization

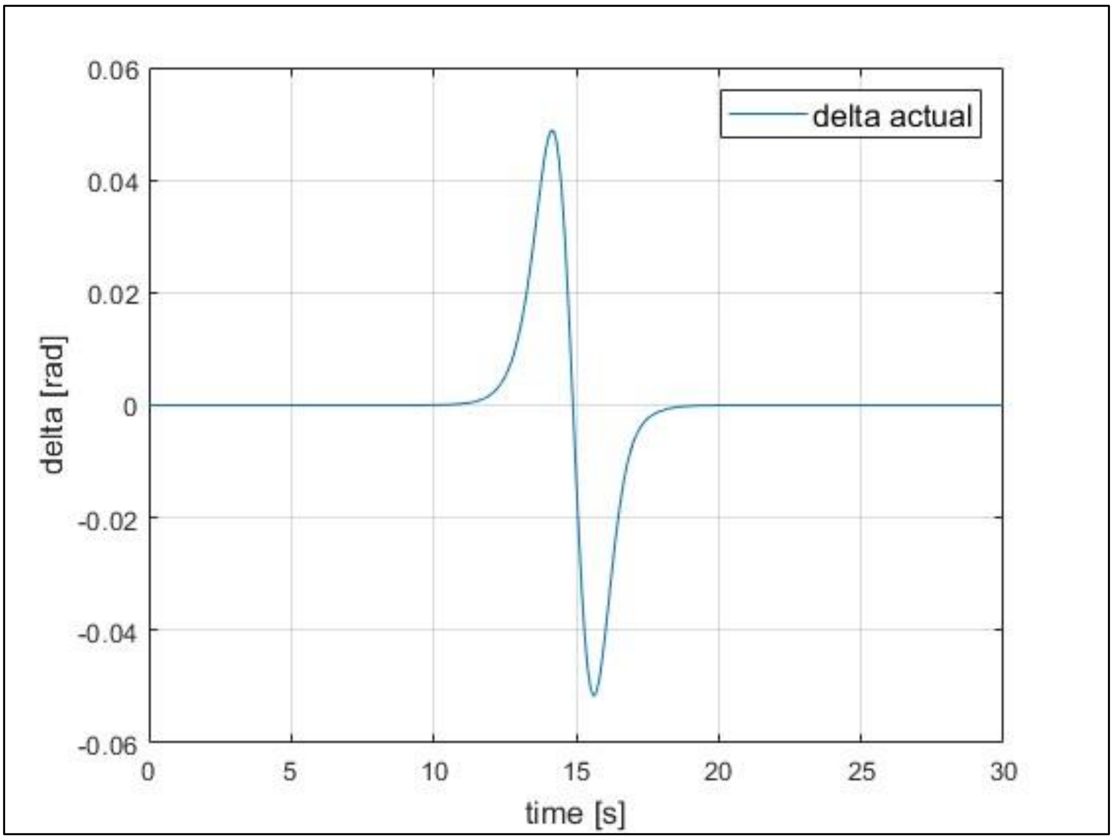


Figure 5.3: Single Lane Change Vehicle Steering Angle

5.2.2 Double Lane Change Manoeuvre

The manoeuvre in Section 5.1 was repeated with a higher turning angle. This manoeuvre is called double lane change manoeuvre. Figure 5.4 was obtained when the "tanh()" function was applied as a reference to simulate the lateral movement from y=-9 to y=9.

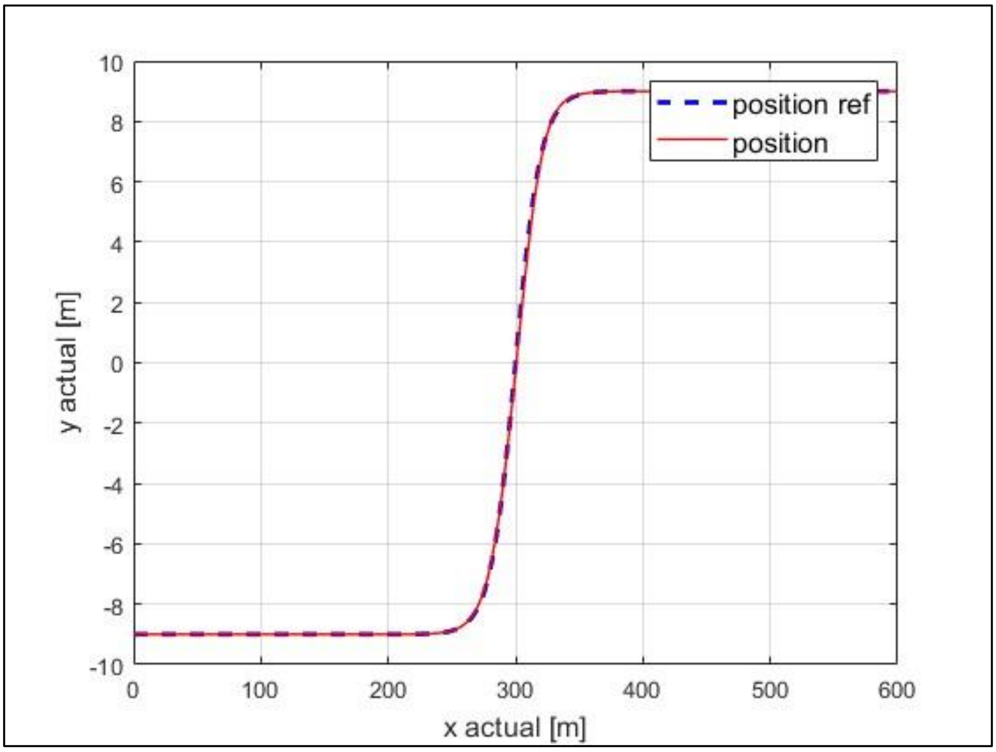


Figure 5.4: Double Lane Change Vehicle Position Comparison Between Reference and Actual Position

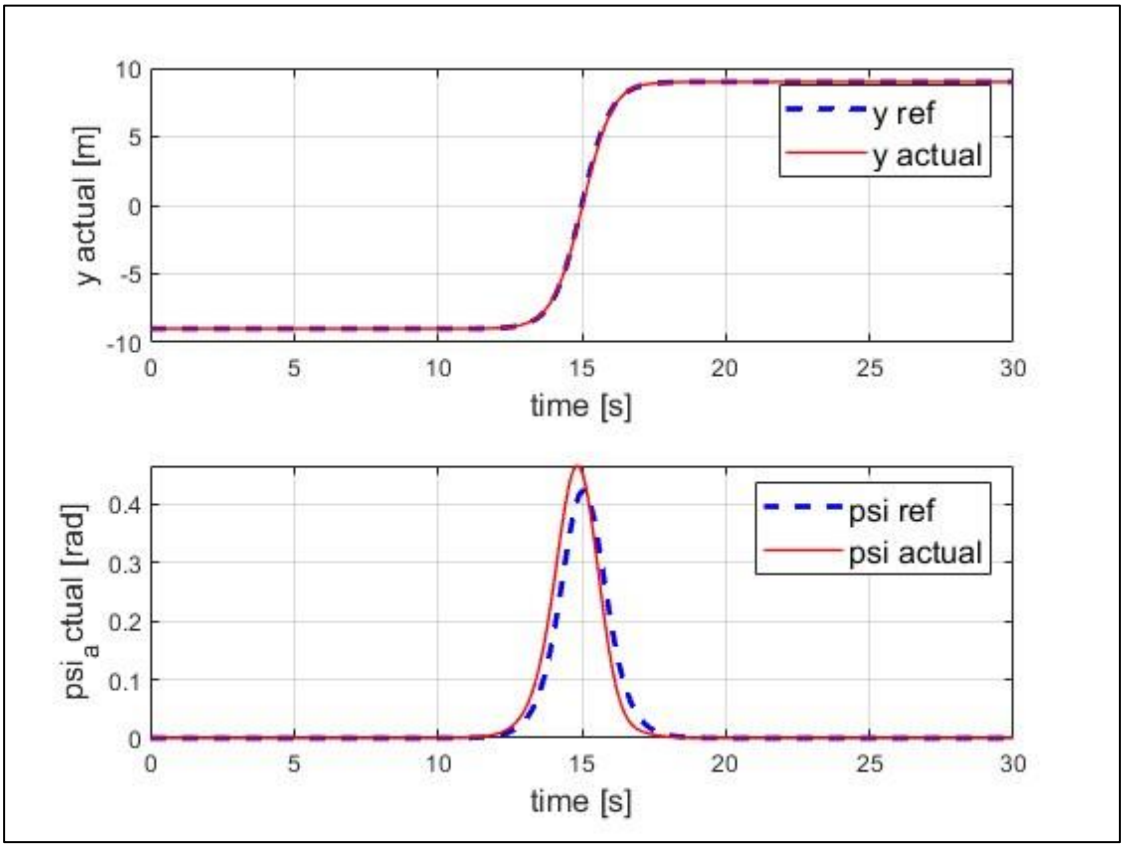


Figure 5.5: Double Lane Change MPC Inputs Comparison and Error Visualization

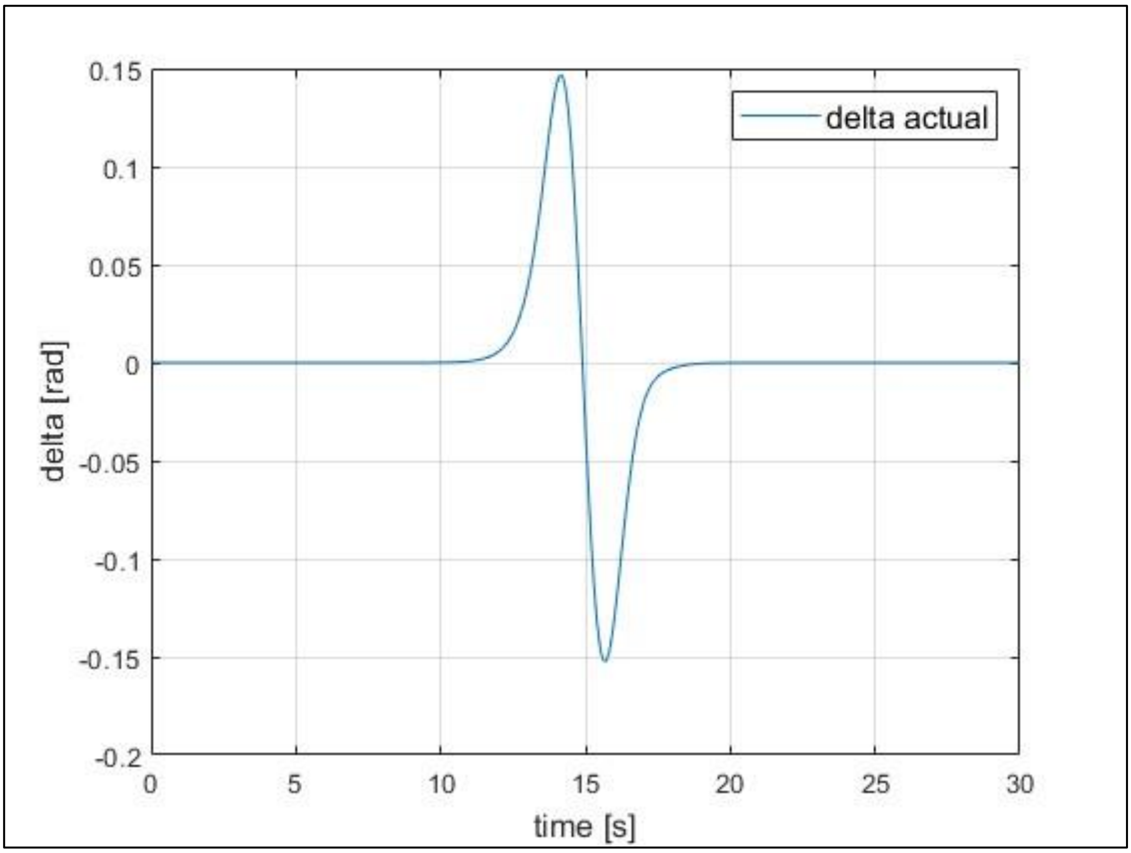


Figure 5.6: Double Lane Change Vehicle Steering Angle

5.3 Validation of the Discrete Time Controller MPC With Another the Well-known Continuous MPC

After controller design, its validation is also important to control engineers. In order to validate the discrete controller, a method was used to compare it with an MPC that is frequently used and known in the literature. This validated continuous time MPC controller belongs to the Model Predictive Control Toolbox in MATLAB Simulink [4].

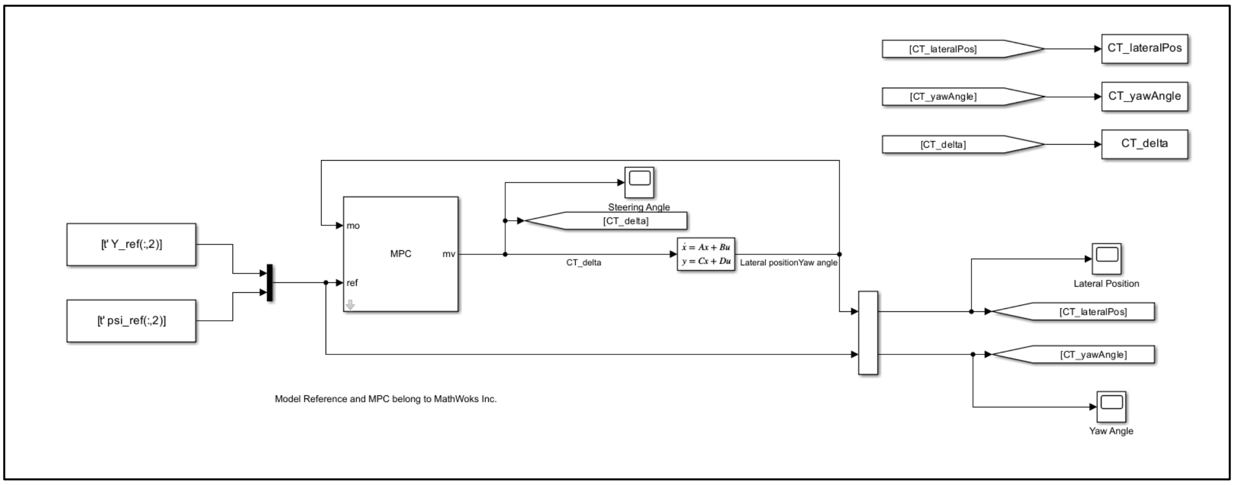


Figure 5.7: Model Predictive Contol Toolbox's MPC Application [4]

5.3.1 Results for Prediction Horizon is 5 Samples

It is important to determine the best prediction horizon for the plant model in which controllers can work. In order to test the performance of the controller, the prediction horizon was increased step by step while the control horizon remained constant. Figure 5.2 shows the comparison for prediction horizon is 5. Performances of both controllers look similar.

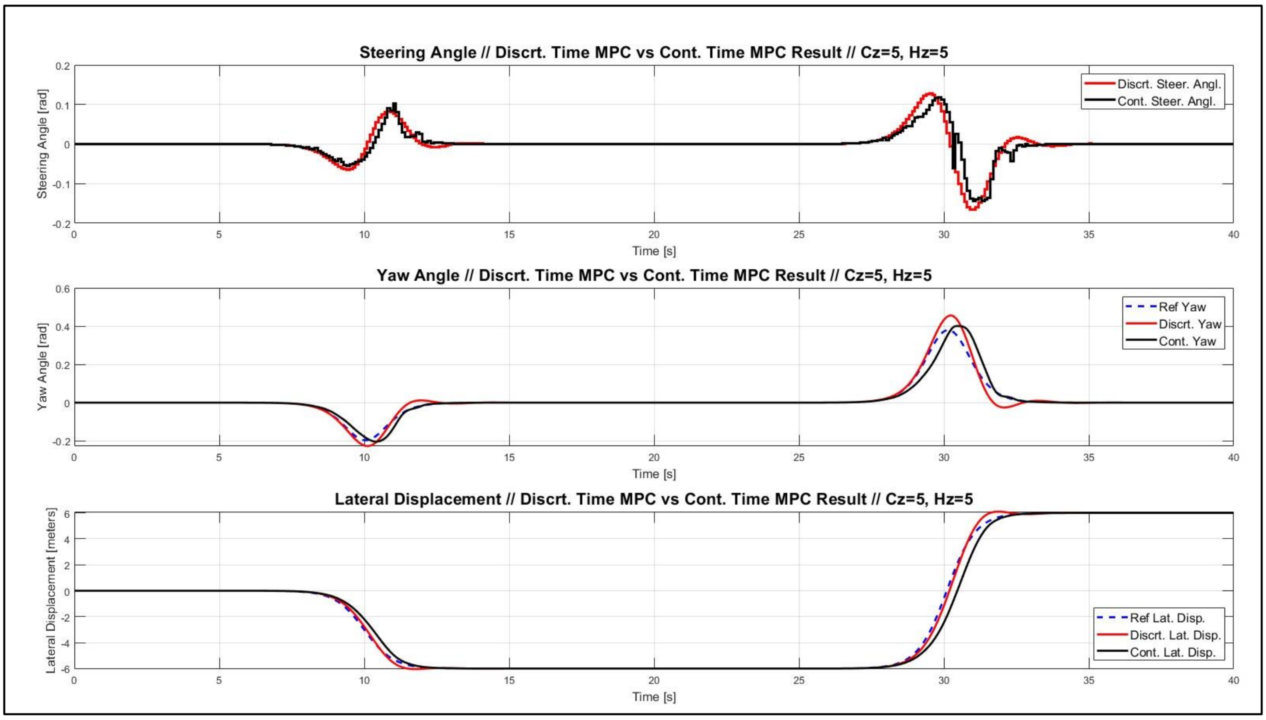


Figure 5.8: Comparison Graph of Continuous MPC and the Discrete MPC on the Same Reference and Constraints at Prediction Horizon is 5

5.3.2 Results for Prediction Horizon is 10 Samples

Figure 5.3 shows the comparison for prediction horizon is 10. Discrete time controller little bit struggles to follow reference yaw angle. But, all the constraints are in the limit.

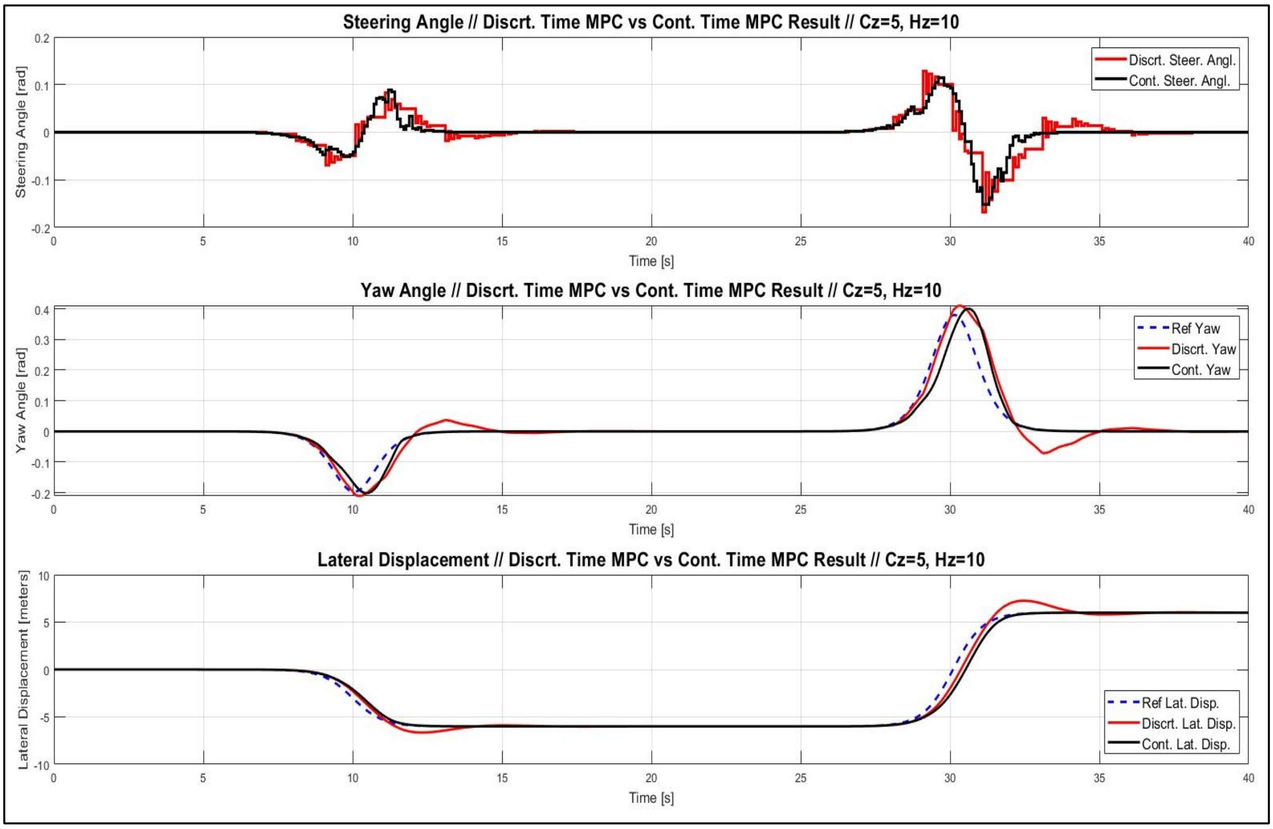


Figure 5.9: Comparison Graph of Continuous MPC and the Discrete MPC on the Same Reference and Constraints at Prediction Horizon is 10

5.3.3 Results for Prediction Horizon is 15 Samples

Figure 5.4 shows the comparison for prediction horizon is 15. Discrete time controller struggles to follow reference yaw angle. But, all the constraints are in the limit.

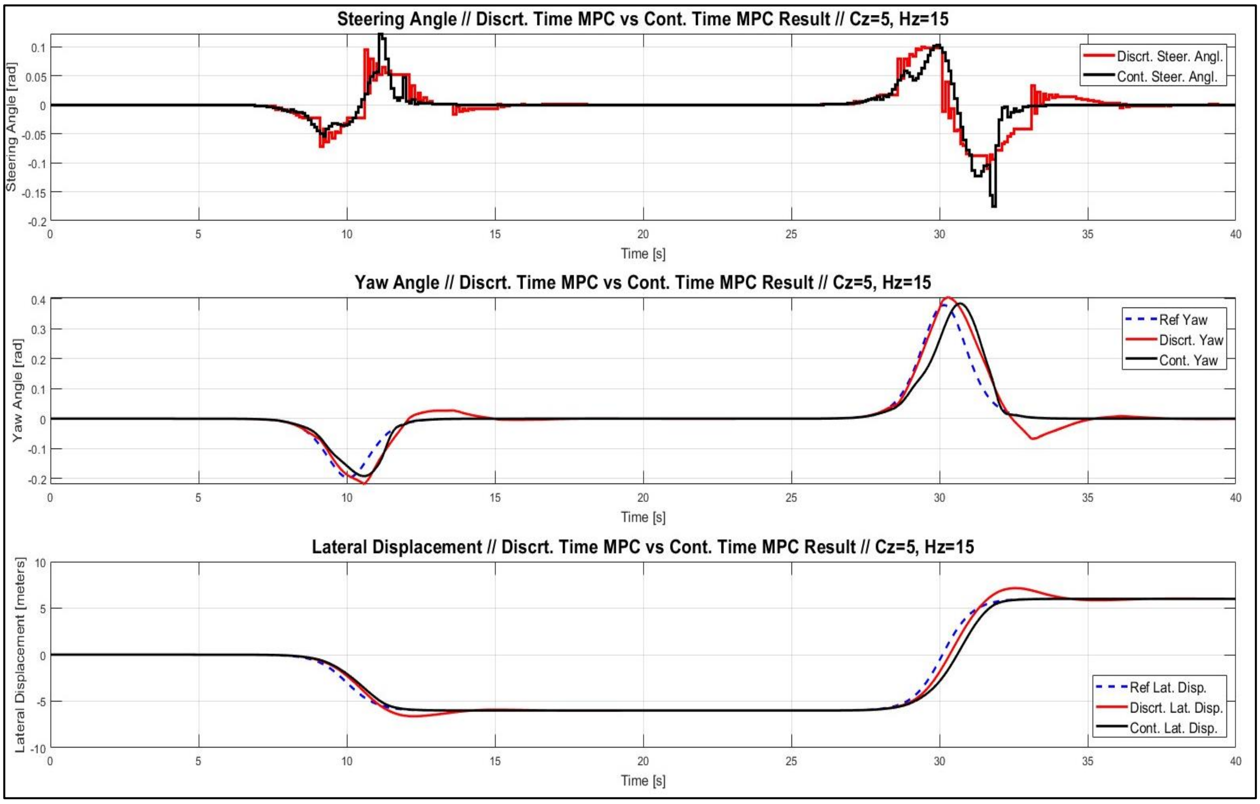


Figure 5.10: Comparison Graph of Continuous MPC and the Discrete MPC on the Same Reference and Constraints at Prediction Horizon is 15

5.3.4 Results for Prediction Horizon is 20 Samples

Figure 5.5 shows the comparison for prediction horizon is 20. Discrete time controller struggles to follow both reference yaw angle and lateral displacement. Additionally, steering angle hits the limit multiple times, which is 0.5 radians.

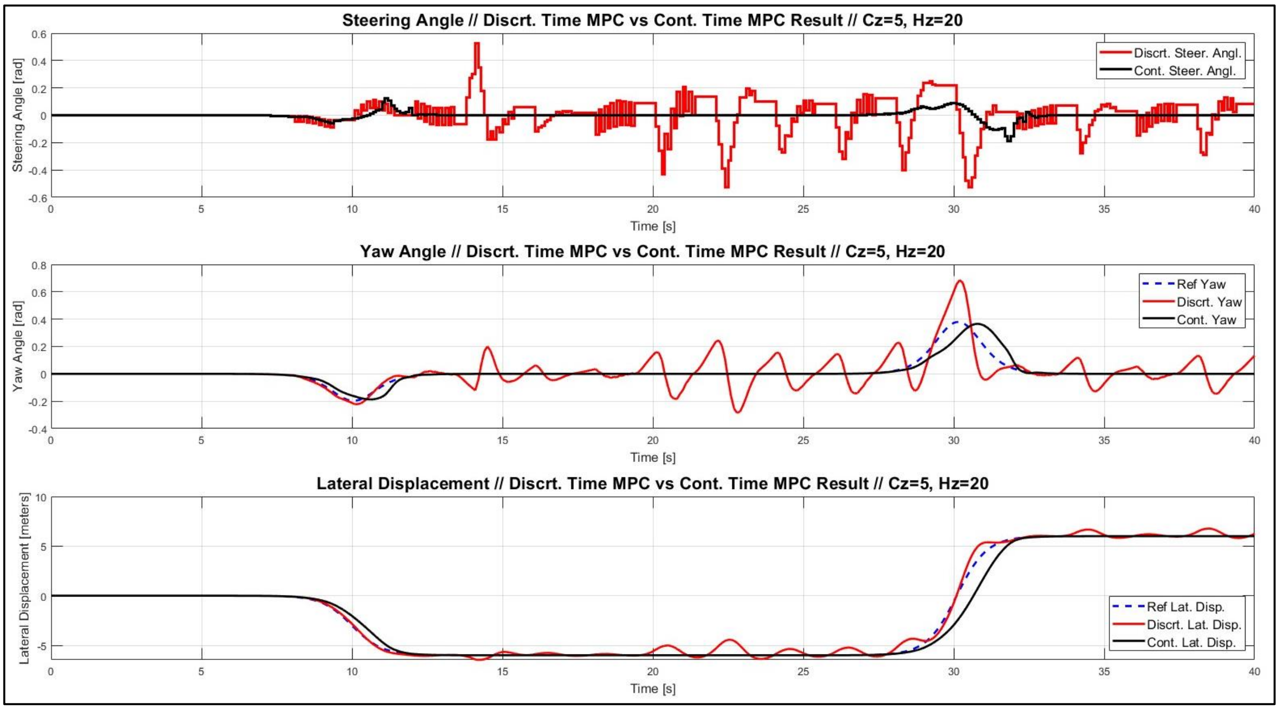


Figure 5.11: Comparison Graph of Continuous MPC and the Discrete MPC on the Same Reference and Constraints at Prediction Horizon is 20

5.4 Conclusion and Final Words

In conclusion, the findings of this thesis underscore the pivotal role played by the prediction horizon and control horizon in determining the efficacy of the Model Predictive Control (MPC) employed in our lane-changing application. The carefully conducted experiments and analyses have illuminated the critical balance required between these two horizons, as they fundamentally shape the controller's ability to predict future system behaviour and prescribe optimal control inputs. It is evident from the results that variations in the prediction and control horizons yield profound effects on the controller's performance.

A longer prediction horizon empowers the controller with a more comprehensive foresight into the system's dynamics, enhancing its predictive accuracy and adaptability to varying conditions. Simultaneously, the control horizon dictates the number of future control inputs employed to guide the system toward the desired trajectory. Striking the right balance between these horizons emerges as a nuanced challenge, as an excessively long prediction horizon may introduce computational complexities that could compromise the real-time applicability of the controller.

Furthermore, the observed impact of a prediction horizon of 20 on the discrete-time MPC's control over the plant model and reference trajectory highlights a critical threshold beyond which computational burdens outweigh performance benefits. These insights illuminate the delicate interplay between prediction and control horizons, emphasizing the need for thoughtful consideration and tuning to optimize the controller's performance while ensuring computational feasibility.

In essence, this research underscores that the prediction horizon and control horizon are not mere technical parameters but rather crucial design elements that profoundly influence the Model Predictive Control's performance in lane-changing applications. As we look to the future of control system design, the lessons learned from this investigation will undoubtedly contribute to the refinement and optimization of MPC controllers, ensuring their effectiveness in dynamic and real-world scenarios.

REFERENCES

[1] Y. Ding, “ShuffleAI,” 14 11 2021. [Online]. Available: https://www.shuffleai.blog/blog/Simple_Understanding_of_Kinematic_Bicycle_Model.html.

[2] Y. Z. J. L. Jiayu Fan, «A hierarchical control strategy for reliable lane changes considering optimal path and lane-changing time point,» Wiley, 2023.

[3] U. Chipengo, “Research Gate,” IEEE, November 2018. [Online]. Available: https://www.researchgate.net/figure/Advanced-driver-assistance-systems-ADAS-for-active-passive-safety-comfort-functionality_fig1_328920450.

[4] D. L. Guillaume Baffet, “Researchgate,” October 2008. [Online]. Available: https://www.researchgate.net/publication/221786406_Estimation_of_Tire-Road_Forces_and_Vehicle_Sideslip_Angle. [Accessed January 2024].

[5] The MathWorks, Inc., “Understanding the Model Predictive Control,” 2019. [Online]. Available: https://www.youtube.com/playlist?list=PLn8PRpmsu08ozoeoXgxPSBKLyd4YEHww8.

[6] M. Ulusoy, “Youtube,” MathWorks Inc., 2019. [Online]. Available: https://www.youtube.com/playlist?list=PLn8PRpmsu08ozoeoXgxPSBKLyd4YEHww8. [Accessed 2023].

[7] P. F. E. D. A. B. Miguel Cerdeira, “Researchgate,” July 2018. [Online]. Available: https://www.researchgate.net/publication/326274980_Reset_Controller_Design_Based_on_Error_Minimization_for_a_Lane_Change_Maneuver. [Accessed January 2024].

[8] J. Krejsa, “Ackermann mobile robot chassis with independent rear wheel drives,” in *Power Electronics and Motion Control Conference (EPE/PEMC)*, Brno, October 2010.

[9] Wikipedia, “Wikipedia,” September 2023. [Online]. Available: https://en.wikipedia.org/wiki/Ackermann_steering_geometry. [Accessed January 2024].

[10] H. R. S. C. W. W. Gang Liu, “The 3-DoF bicycle model with the simplified piecewise linear tire model,” in *Proceedings 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC)*, Shengyang, 2013.

[11] D. H. P. K. T. S. Adam Owczarkowski, “Dynamic modeling and simulation of a bicycle stabilized by LQR control,” in *2016 21st International Conference on Methods and Models in Automation and Robotics (MMAR)*, Miedzyzdroje, Poland, 2016.

[12] V. P. Mark Misin, “LPV MPC Control of an Autonomous Aerial Vehicle,” in *2020 28th Mediterranean Conference on Control and Automation (MED)*, France, 2020.

[13] B. B. R. S. R. M. D. D. S. Goran S. Vorotovic, «Determination of Cornering Stiffness,» *FME Transactions,* pp. 66-71, July 2013.

[14] O. S. Yurii Mariiash, “Implementation of model predictive control in a programmable logic controller,” in *Modeling, control and information technologies: Proceedings of VI International scientific and practical conference*, Ukraine, 2023.

[15] Q. L. S. W. L. G. Yonghua Huang, Dynamic modeling and analysis of a front-wheel drive bicycle robot moving on a slope, Hong Kong: IEEE International Conference on Automation and Logistics, 2010.

[16] B. S. Yanggu Zheng, “A Real-Time Nonlinear MPC for Extreme Lateral Stabilization of Passenger Vehicles,” in *2019 IEEE International Conference on Mechatronics (ICM)*, Ilmenau, Germany, 2019.

[17] M. P. Sorawuth Vatanashevanopakorn, “Steering control based balancing of a bicycle robot,” in *2011 IEEE International Conference on Robotics and Biomimetics*, Karon Beach, Thailand, 2011.

[18] Y. Q. Sisi Pan, “Optimal control via weighted congestion game with linear cost functions,” in *2017 36th Chinese Control Conference (CCC)*, Dalian, China, 2017.

[19] B. Özece, “MATLAB-Simulink Dersleri (Sistem Dinamiği Uygulamaları),” Udemy, 2021. [Online]. Available: https://www.udemy.com/course/matlab-simulink-dersleri-temelden-ileri-seviyeye/.

[20] Z. J. T. C. X. J. Lei Li, “Optimal Model Predictive Control for Path Tracking of Autonomous Vehicle,” in *2011 Third International Conference on Measuring Technology and Mechatronics Automation*, Shanghai, China, 2011.

[21] L. W. C. M. Guofei Xiang, “Trajectory Tracking Control of Transformer Inspection Robot Using Distributed Model Predictive Control,” in *College of Electrical Engineering, Sichuan University*, Chengdu, China, 2023.

[22] A. B. M. F. D. H. F. Borrelli, "An MPC/hybrid system approach to traction control," *IEEE Transactions on Control Systems Technology,* pp. 541-562, 24 April 2006.

[23] C. U. Dogruer, “Model Predictive Control Parameterized in Terms of Orthogonal Polynomials,” in *IEEE 11th International Conference on Systems and Control*, Sousse, Tunisia, 2023.

[24] T.-D. C. Chih-Keng Chen, “Modeling and Model Predictive Control for a Bicycle-Rider System,” in *2015 2nd International Conference on Information Science and Control Engineering*, Shanghai, China, 2015.

[25] M. B. T. Ai-Buraiki, Model predictive control design approach for autonomous bicycle kinematics stabiliza-tion, Palermo, Italy: 22nd Mediterranean Conference on Control and Automation, 2014.

[26] MathWorks, Inc., “Specify Cost Function for Nonlinear MPC,” [Online]. Available: https://www.mathworks.com/help/mpc/ug/specify-cost-function-for-nonlinear-mpc.html.

[27] The MathWorks, Inc., “Quadratic programming - MATLAB quadprog,” [Online]. Available: https://www.mathworks.com/help/optim/ug/quadprog.html?lang=en.

APPENDIX-A

MATLAB CODES

A-1 TRAJECTORY CREATOR FUNCTION:

```
%% hasankaantuna // 19015036 // trajectory creator script function

%%
function [psi_ref,X_ref,Y_ref]=trajectoryCreator(t, x_dot,trajectoryType)

if trajectoryType==1
    x=linspace(0,x_dot*t(end),length(t));
    y=9*tanh((t-t(end)/2));
else
    x=linspace(0,x_dot*t(end),length(t));
    y=-9*ones(1,length(t));
end

dx=x(2:end)-x(1:end-1);
dy=y(2:end)-y(1:end-1);

psi=zeros(1,length(x));
psiInt=psi;

psi(1)=atan2(dy(1),dx(1));
psi(2:end)=atan2(dy(:),dx(:));
dpsi=psi(2:end)-psi(1:end-1);

psiInt(1)=psi(1);
for i = 2:length(psiInt)
    if dpsi(i-1)<-pi
        psiInt(i)=psiInt(i-1)+(dpsi(i-1)+2*pi);
    elseif dpsi(i-1)>pi
        psiInt(i)=psiInt(i-1)+(dpsi(i-1)-2*pi);
    else
        psiInt(i)=psiInt(i-1)+dpsi(i-1);
    end
end


X_ref = [t' x'];
Y_ref = [t' y'];
psi_ref = [t' psiInt'];

end
```

A-2 MPC SETUP AND CALCULATOR OF DELTA(δ) FOR EACH SAMPLES

```
%% hasankaantuna // 19015036 // MPC script function

%%
function [Hdb,Fdbt,Cdb,Adb] = MPC_script(Ad,Bd,Cd,Dd,hz)

    Q=[100 0;0 1];
    S=[10 0;0 1];
    R=100;


A_aug=[Ad,Bd;zeros(length(Bd(1,:)),length(Ad(1,:))),eye(length(Bd(1,:)))];
    B_aug=[Bd;eye(length(Bd(1,:)))];
    C_aug=[Cd,zeros(length(Cd(:,1)),length(Bd(1,:)))];
    D_aug=Dd; % D_aug is not used because it is a zero matrix

    CQC=C_aug'*Q*C_aug;
    CSC=C_aug'*S*C_aug;
    QC=Q*C_aug;
    SC=S*C_aug;

    Qdb=zeros(length(CQC(:,1))*hz,length(CQC(1,:))*hz);
    Tdb=zeros(length(QC(:,1))*hz,length(QC(1,:))*hz);
    Rdb=zeros(length(R(:,1))*hz,length(R(1,:))*hz);
    Cdb=zeros(length(B_aug(:,1))*hz,length(B_aug(1,:))*hz);
    Adb=zeros(length(A_aug(:,1))*hz,length(A_aug(1,:)));

    for i = 1:hz
       if i == hz
           Qdb(1+length(CSC(:,1))*(i-
1):length(CSC(:,1))*i,1+length(CSC(1,:))*(i-1):length(CSC(1,:))*i)=CSC;
           Tdb(1+length(SC(:,1))*(i-
1):length(SC(:,1))*i,1+length(SC(1,:))*(i-1):length(SC(1,:))*i)=SC;
       else
           Qdb(1+length(CQC(:,1))*(i-
1):length(CQC(:,1))*i,1+length(CQC(1,:))*(i-1):length(CQC(1,:))*i)=CQC;
           Tdb(1+length(QC(:,1))*(i-
1):length(QC(:,1))*i,1+length(QC(1,:))*(i-1):length(QC(1,:))*i)=QC;
       end

       Rdb(1+length(R(:,1))*(i-1):length(R(:,1))*i,1+length(R(1,:))*(i-
1):length(R(1,:))*i)=R;

       for j = 1:hz
           if j<=i
               Cdb(1+length(B_aug(:,1))*(i-
1):length(B_aug(:,1))*i,1+length(B_aug(1,:))*(j-
1):length(B_aug(1,:))*j)=A_aug^(i-j)*B_aug;
           end
       end
       Adb(1+length(A_aug(:,1))*(i-
1):length(A_aug(:,1))*i,1:length(A_aug(1,:)))=A_aug^(i);
    end
    Hdb=Cdb'*Qdb*Cdb+Rdb;
    Fdbt=[Adb'*Qdb*Cdb;-Tdb*Cdb];
end
```

A-3 MAIN SCRIPT FILE TO RUN THE SYSTEM THAT INCLUDES VEHICLE STATE SPACE LTI MODEL AND DISCRITIZER

```
%% hasankaantuna // 19015036 // main script


%% Defining the constant values
% Constants
m=1575;
Iz=2875;
Caf=19000;
Car=33000;
lf=1.2;
lr=1.6;
Ts=0.1;
hz =15;
cz=5;
x_dot=15;
numberControlledStates=2;
velReassing=0;
velReassingSetSpeed=x_dot;
trajectoryType=2;
totalT=40;

%% Calling reference creator script

t = 0:Ts:totalT;
%  [psi_ref,X_ref,Y_ref]=trajectoryCreator(t,x_dot,trajectoryType);
% psi_ref=[t,psi_ref(:,2)];
% Y_ref=[t,Y_refNew(:,2)];
% x=linspace(0,x_dot*t(end),length(t));
% X_ref=[t x'];
sim_length=length(t);

refSignals=zeros(length(X_ref(:,2))*numberControlledStates,1);

k=1;
for i = 1:numberControlledStates:length(refSignals)
    refSignals(i)=psi_ref(k,2);
    refSignals(i+1)=Y_ref(k,2);
    k=k+1;
end

clear i k

%% Initial state definitions

y_dot=0;
psi=psi_ref(1,2);
psi_dot=0;

if trajectoryType==2
    Y=0;
else
    Y=Y_ref(1,2);
end

states=[y_dot,psi,psi_dot,Y];
```

```
statesTotal=zeros(length(t),length(states));
statesTotal(1,:)=states;

%% Initiate the controller - simulation loops

delta=0;
dutotal=zeros(length(t),1);
deltaTotal=zeros(length(t),1);
deltaTotal(1,:)=delta;

psi = states(2);
Y=states(4);

%% Definitions of State Space Matrices

scriptA=[-(2*Caf+2*Car)/(m*x_dot) 0 (-x_dot-(2*Caf*lf-2*Car*lr)/(m*x_dot)) 0;
...
    0 0 1 0; ...
    -(2*lf*Caf-2*lr*Car)/(Iz*x_dot) 0 -(2*lf^2*Caf+2*lr^2*Car)/(Iz*x_dot) 0;
...
    1 x_dot 0 0];
scriptB=[2*Caf/m;0;2*lf*Caf/Iz;0];
C= [0 1 0 0;0 0 0 1];
D=[0; 0];

%% Definition of Discrete State Space Model

Ad=eye(length(scriptA(1,:)))+Ts*scriptA;
Bd=Ts*scriptB;
Cd=C;
Dd=D;

%%
[Hdb,Fdbt,Cdb,Adc]=MPC_script(Ad,Bd,Cd,Dd,hz);
l=1;
k=1;
o=hz;
for i =1:sim_length-1


    % Generating the current state and the reference vector
    x_aug_t=[states';delta];

    k=k+numberControlledStates;

    if k+numberControlledStates*hz-1 <= length(refSignals)
        r=refSignals(k:k+numberControlledStates*hz-1);
    else
        r=refSignals(k:length(refSignals));
        hz=hz-1;
    end

    if hz<o
        [Hdb,Fdbt,Cdb,Adc]=MPC_script(Ad,Bd,Cd,Dd,hz);
    end

        % Calling the optimizer (quadprog) // Cost function in quadprog:
min(du)*1/2*du'Hdb*du+f'du
```

```
        ft=[x_aug_t',r']*Fdbt;
        if l<=cz
        % Call the solver
        options = optimset('Display', 'on');
        [du,fval]=quadprog(Hdb,ft,[],[],[],[],[],[],[],options);
        dutotal(i+1,:)=delta+du(1);
        % Update the real inputs

            l=l+1;
            delta=du(1);
            if delta < -pi/6
                delta=-pi/6;
            elseif delta>pi/6
                delta=pi/6;
            else
                delta=delta;
            end
            deltaTotal(i+1,:)=delta;
            deltaInc=deltaTotal(i+1,:)-deltaTotal(i,:);
            deltaInc2=deltaTotal(i+1,:)+deltaTotal(i,:);
            if pi/18>deltaInc && deltaInc>-pi/18
                delta=delta;
            elseif pi/1<deltaInc
                delta=deltaTotal(i,:)+pi/18;
            elseif deltaInc<-pi/18
                delta=deltaTotal(i,:)-pi/18;
            end

            deltaTotal(i+1,:)=delta;
            if l==o
                l=1;
            end
        elseif l==o
            l=1;
            delta=delta;
            deltaTotal(i+1,:)=delta;
        else
        l=l+1;
        delta=delta;
        deltaTotal(i+1,:)=delta;
        end
        deltaTotal(i+1,:)=delta;


    % Assign the states

    y_dot=states(1);
    psi = states(2);
    psi_dot = states(3);

    % The nonlinear equation describing the dynamics of the vehicle
        dx(1,1)=-(2*Caf+2*Car)/(m*x_dot)*y_dot+(-x_dot-(2*Caf*lf-
2*Car*lr)/(m*x_dot))*psi_dot+2*Caf/m*delta;
        dx(2,1)=psi_dot;
        dx(3,1)=-(2*lf*Caf-2*lr*Car)/(Iz*x_dot)*y_dot-
(2*lf^2*Caf+2*lr^2*Car)/(Iz*x_dot)*psi_dot+2*lf*Caf/Iz*delta;
        dx(4,1)=sin(psi)*x_dot+cos(psi)*y_dot;

    % Simulate the new states
    T = (Ts)*(i-1):(Ts)/30:Ts*(i-1)+(Ts);
    [T,x]=ode45(@(t,x) dx,T,states);
    states=x(end,:);
    statesTotal(i+1,:)=states;
    if velReassing==1
        if states(4)-Y_ref(i,2)>0.5
            x_dot=x_dot-x_dot*5/100;
            X_ref(i+1,2)=X_ref(i,2)+(x_dot*Ts);

            scriptA=[-(2*Caf+2*Car)/(m*x_dot) 0 (-x_dot-(2*Caf*lf-
2*Car*lr)/(m*x_dot)) 0; ...
                0 0 1 0; ...
                -(2*lf*Caf-2*lr*Car)/(Iz*x_dot) 0 -
(2*lf^2*Caf+2*lr^2*Car)/(Iz*x_dot) 0; ...
                1 x_dot 0 0];
            scriptB=[2*Caf/m;0;2*lf*Caf/Iz;0];
            C= [0 1 0 0;0 0 0 1];
            D=[0; 0];

            Ad=eye(length(scriptA(1,:)))+Ts*scriptA;
            Bd=Ts*scriptB;
            Cd=C;
            Dd=D;
            [Hdb,Fdbt,Cdb,Adc]=MPC_script(Ad,Bd,Cd,Dd,hz);
        else
            if x_dot+x_dot*15/100<velReassingSetSpeed
                x_dot=x_dot+x_dot*15/100;
            X_ref(i+1,2)=X_ref(i,2)+(x_dot*Ts);
            scriptA=[-(2*Caf+2*Car)/(m*x_dot) 0 (-x_dot-(2*Caf*lf-
2*Car*lr)/(m*x_dot)) 0; ...
                0 0 1 0; ...
                -(2*lf*Caf-2*lr*Car)/(Iz*x_dot) 0 -
(2*lf^2*Caf+2*lr^2*Car)/(Iz*x_dot) 0; ...
                1 x_dot 0 0];
            scriptB=[2*Caf/m;0;2*lf*Caf/Iz;0];
            C= [0 1 0 0;0 0 0 1];
            D=[0; 0];

            Ad=eye(length(scriptA(1,:)))+Ts*scriptA;
            Bd=Ts*scriptB;
            Cd=C;
            Dd=D;
            [Hdb,Fdbt,Cdb,Adc]=MPC_script(Ad,Bd,Cd,Dd,hz);
            else
                x_dot=velReassingSetSpeed;
            end
        end
    end
    x_dotTotal(i)=x_dot;
end
    x_dotTotal(i+1)=x_dot;
```

A-4 PLOTTING AND ANALYSIS SCRIPTS

```
%% hasankaantuna // 190150636 // plotterScript

%% Plotting the position

figure;
subplot(2,1,1)
plot(t(1:sim_length),Y_ref(1:sim_length,2),'b','LineWidth',2)
title('Lateral Displacement // Discrete Time MPC Result // Cz=5,
Hz=20','FontSize',16)
xlabel('Time [s]','FontSize',12)
ylabel('Lateral Displacement [m]','FontSize',12)
legend({'y ref','y actual'},'Location','southeast','FontSize',12)
hold on

plot(t(1:sim_length),statesTotal(1:sim_length,4),'r','LineWidth',2)
grid on
legend({'y ref','y actual'},'Location','southeast','FontSize',12)
xlim([0 40])
%%%%%%%%%%%%%%%%%%%%%%%

subplot(2,1,2)
plot(CT_lateralPos,'LineWidth',2)
title('Lateral Displacement // Continious Time MPC Result // Cz=5,
Hz=20','FontSize',16)
xlabel('Time [s]','FontSize',12)
ylabel('Lateral Displacement [m]','FontSize',12)
legend({'y ref','y actual'},'Location','southeast','FontSize',12)
xlim([0 40])
grid on

%% Plotting the Yaw

figure;
subplot(2,1,1)
plot(t(1:sim_length),psi_ref(:,2),'b','LineWidth',2)
title('Yaw Angle // Discrete Time MPC Result // Cz=5, Hz=20','FontSize',16)
xlabel('Time [s]','FontSize',12)
ylabel('Yaw Angle [rad]','FontSize',12)
legend({'yaw ref','yaw actual'},'Location','northeast','FontSize',12)
hold on

plot(t(1:sim_length),statesTotal(1:sim_length,2),'r','LineWidth',2)
grid on
legend({'yaw ref','yaw actual'},'Location','northeast','FontSize',12)
xlim([0 40])
%%%%%%%%%%%%%%%%%%%%%%%

subplot(2,1,2)
plot(CT_yawAngle,'LineWidth',2)
title('Yaw Angle // Continious Time MPC Result // Cz=5, Hz=20','FontSize',16)
xlabel('Time [s]','FontSize',12)
ylabel('Yaw Angle [rad]','FontSize',12)
legend({'yaw ref','yaw actual'},'Location','northeast','FontSize',12)
xlim([0 40])
grid on

%% Plotting Control Inputs
```

```
figure;
subplot(3,1,1)
plot(t(1:sim_length),deltaTotal,'b','LineWidth',2)
title('Steering Angle // Discrete Time MPC Result // Cz=5,
Hz=20','FontSize',16)
xlabel('Time [s]','FontSize',12)
ylabel('Steering Angle [rad]','FontSize',12)
legend({'steering angle'},'Location','northeast','FontSize',12)
grid on
xlim([0 40])

%%%%%%%%%%%%%%%%%%%%%%%

subplot(3,1,2)
plot(t(1:sim_length),CT_delta.Data,'LineWidth',2)
title('Steering Angle // Continious Time MPC Result // Cz=5,
Hz=20','FontSize',16)
xlabel('Time [s]','FontSize',12)
ylabel('Steering Angle [rad]','FontSize',12)
legend({'yaw ref','yaw actual'},'Location','northeast','FontSize',12)
xlim([0 40])
grid on

subplot(3,1,3)
plot(t(1:sim_length),deltaTotal,t(1:sim_length),CT_delta.Data,'r','LineWidth'
,2)
title('Steering Angle // Comparison of Discr. and Cont. // Cz=5,
Hz=20','FontSize',16)
xlabel('Time [s]','FontSize',12)
ylabel('Steering Angle [rad]','FontSize',12)
legend({'Steering(discrete)','Steering(continious)'},'Location','northeast','
FontSize',12)
xlim([0 40])
grid on


%% hasankaantuna // 190150636 // plotterScript2

%% Discrete MPC Results

figure;
subplot(3,1,1)
plot(t(1:sim_length),deltaTotal,'b','LineWidth',2)
title('Steering Angle // Discrete Time MPC Result // Cz=5,
Hz=20','FontSize',16)
xlabel('Time [s]','FontSize',12)
ylabel('Steering Angle [rad]','FontSize',12)
legend({'steering angle'},'Location','northeast','FontSize',12)
grid on
xlim([0 40])

%%%%%%%%%%%%%%%%%%%%%%%

subplot(3,1,2)
plot(t(1:sim_length),psi_ref(1:sim_length,2),'--b','LineWidth',2)
hold on
plot(t(1:sim_length),statesTotal(1:sim_length,2),'r','LineWidth',1)
title('Yaw Angle // Discrete Time MPC Result // Cz=5, Hz=20','FontSize',16)
```

```
xlabel('Time [s]','FontSize',12)
ylabel('Steering Angle [rad]','FontSize',12)
legend({'yaw ref','yaw actual'},'Location','northeast','FontSize',12)
xlim([0 40])
grid on


%%%%%%%%%%%%%%%%%%%%%%%

subplot(3,1,3)
plot(t(1:sim_length),Y_ref(1:sim_length,2),'--b','LineWidth',2)
hold on
plot(t(1:sim_length),statesTotal(1:sim_length,4),'r','LineWidth',1)
title('Lateral Displacement // Discrete Time MPC Result // Cz=5,
Hz=20','FontSize',16)
xlabel('Time [s]','FontSize',12)
ylabel('Steering Angle [rad]','FontSize',12)
legend({'y ref','y actual'},'Location','southeast','FontSize',12)
xlim([0 40])
grid on

%% Continuous MPC Results

figure;
subplot(3,1,1)
plot(t(1:sim_length),CT_delta.Data,'b','LineWidth',2)
title('Steering Angle // Continuous Time MPC Result // Cz=5,
Hz=20','FontSize',16)
xlabel('Time [s]','FontSize',12)
ylabel('Steering Angle [rad]','FontSize',12)
legend({'steering angle'},'Location','northeast','FontSize',12)
grid on
xlim([0 40])

%%%%%%%%%%%%%%%%%%%%%%%

subplot(3,1,2)
hold on
plot(CT_yawAngle,'LineWidth',2)
title('Yaw Angle // Continuous Time MPC Result // Cz=5, Hz=20','FontSize',16)
xlabel('Time [s]','FontSize',12)
ylabel('Steering Angle [rad]','FontSize',12)
legend({'yaw ref','yaw actual'},'Location','northeast','FontSize',12)
xlim([0 40])
grid on


%%%%%%%%%%%%%%%%%%%%%%%

subplot(3,1,3)
hold on
plot(CT_lateralPos,'LineWidth',2)
title('Lateral Displacement // Continuous Time MPC Result // Cz=5,
Hz=20','FontSize',16)
xlabel('Time [s]','FontSize',12)
ylabel('Steering Angle [rad]','FontSize',12)
legend({'y ref','y actual'},'Location','southeast','FontSize',12)
xlim([0 40])
grid on
```

```
%% hasankaantuna // 190150636 // plotterScript3

%% Discrete MPC vs Continuous MPC Results

figure;
subplot(3,1,1)
stairs(t(1:sim_length),deltaTotal,'r','LineWidth',2.5)
hold on
stairs(t(1:sim_length),CT_delta.Data,'k','LineWidth',2.5)
title('Steering Angle // Discrt. Time MPC vs Cont. Time MPC Result // Cz=5,
Hz=20','FontSize',16)
xlabel('Time [s]','FontSize',12)
ylabel('Steering Angle [rad]','FontSize',12)
legend({'Discrt. Steer. Angl.','Cont. Steer.
Angl.'},'Location','northeast','FontSize',12)
grid on
xlim([0 40])

%%%%%%%%%%%%%%%%%%%%%%%

subplot(3,1,2)
plot(t(1:sim_length),psi_ref(1:sim_length,2),'--b','LineWidth',2)
hold on
plot(t(1:sim_length),statesTotal(1:sim_length,2),'r','LineWidth',2)
hold on
plot(t(1:sim_length),CT_yawAngle.Data(:,1),'k','LineWidth',2)
title('Yaw Angle // Discrt. Time MPC vs Cont. Time MPC Result // Cz=5,
Hz=20','FontSize',16)
xlabel('Time [s]','FontSize',12)
ylabel('Yaw Angle [rad]','FontSize',12)
legend({'Ref Yaw','Discrt. Yaw','Cont.
Yaw'},'Location','northeast','FontSize',12)
xlim([0 40])
grid on


%%%%%%%%%%%%%%%%%%%%%%%

subplot(3,1,3)
plot(t(1:sim_length),Y_ref(1:sim_length,2),'--b','LineWidth',2)
hold on
plot(t(1:sim_length),statesTotal(1:sim_length,4),'r','LineWidth',2)
hold on
plot(t(1:sim_length),CT_lateralPos.Data(:,2),'k','LineWidth',2)
title('Lateral Displacement // Discrt. Time MPC vs Cont. Time MPC Result //
Cz=5, Hz=20','FontSize',16)
xlabel('Time [s]','FontSize',12)
ylabel('Lateral Displacement [meters]','FontSize',12)
legend({'Ref Lat. Disp.','Discrt. Lat. Disp.','Cont. Lat.
Disp.'},'Location','southeast','FontSize',12)
xlim([0 40])
grid on
```

RESUME

1. STUDENT PERSONAL INFORMATION

Name SURNAME	: Hasan Kaan TUNA
Birthplace and Date	: Kocasinan 29.10.2000
Foreign Language	: English
E-mail	: kaan.tuna@std.yildiz.edu.tr

EDUCATION STATUS

Degree	**Field**	**School/University**	**Graduation Year**
High School	Science	Sınav Koleji	2018

JOB EXPERIENCES

Year	**Firm**	**Duty**
2021	Nokta Makina	Automation Dept. Intern
2023	Ford Otomotiv Sanayi A.Ş.	Powertrain Control and Calibration Intern