Holistic Vehicle Control Using Learning MPC

by

Chao Yu

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2023

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner: Ya-Jun Pan
Title: Professor
Department: Mechanical Engineering
University: Dalhousie University

Supervisor(s): Amir Khajepour
Title: Professor
Department: Mechanical and Mechatronics Engineering

Internal Member: Soo Jeon
Title: Associate Professor
Department: Mechanical and Mechatronics Engineering

Internal Member: Zhao Pan
Title: Assistant Professor
Department: Mechanical and Mechatronics Engineering

Internal-External Member: Jun Liu
Title: Associate Professor
Department: Applied Mathematics

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The field of vehicle stability control has witnessed significant advancements in recent years, and Holistic Vehicle Control (HVC) has emerged as a method to improve the safety and performance of modern automobiles. HVC aims to control a vehicle in a comprehensive manner, considering various factors that influence stability. Model Predictive Control (MPC) is a powerful control framework that has been adopted in various industries due to its ability to handle complex dynamics and constraints. Traditional MPC methods for vehicle stability control systems use a simplified physics-based vehicle model to forecast vehicle dynamics behaviors. These models may either provide a poor prediction for the nonlinear vehicle dynamics or result in unacceptable computational costs for real-time vehicle applications when the models become more complex. Besides, there are many modeling and parameter uncertainties, including road conditions, vehicle weight changes, tire dynamics, and road grades, etc., that the simplified physics-based vehicle model cannot capture in all practical conditions. These drawbacks lead to the deterioration of vehicle stability control performance.

In recent years, learning MPC schemes have been introduced to address these challenges of traditional MPC. They typically leverage different machine learning techniques to learn the system dynamics directly from data, allowing it to handle model uncertainty more effectively. Besides, they can adapt to changes by continuously updating the learned model using real-time data, ensuring that the controller remains effective even as the system evolves. However, there are some challenges for the existing learning MPC techniques. Firstly, learning-based control approaches often lack interpretability. Understanding and interpreting the learned models and their learning and prediction processes are crucial for safety critical systems such as vehicle stability systems. Secondly, existing learning MPC techniques rely solely on learned models, which might result in poor performance or instability if the model encounters scenarios that differ significantly from the training data. Thirdly, existing learning MPC techniques typically require large amounts of high-quality data for training accurate models, which can be expensive or impractical in the vehicle stability control domain.

To address these challenges, this thesis proposes a novel hybrid learning MPC approach for HVC. The main objective is to leverage the capabilities of machine learning algorithms to learn accurate and adaptive models of vehicle dynamics from data, enabling enhanced control strategies for improved stability and maneuverability. The hybrid learning MPC scheme maintains a traditional physics-based vehicle model and a data-based learning model. In the learned model, a variety of machine learning techniques can be used to predict vehicle dynamics based on learning from collected vehicle data. The performance of the developed hybrid learning MPC controller using torque vectoring (TV) as the actuator is evaluated through the Matlab/Simulink and CarSim co-simulation with a high-fidelity Chevy Equinox vehicle model under a series of harsh maneuvers. Extensive real-world experiments using a Chevy Equinox electric testing vehicle are conducted. Both simulation results and experimental results show that the developed hybrid learning MPC approach consistently outperforms existing MPC methods with better yaw rate tracking performance and smaller vehicle sideslip under various driving conditions.

Traditionally, the length of the prediction horizon in MPC is predetermined and remains constant throughout the control process. Fixed-prediction-horizon MPC may perform well under steady-state conditions or in scenarios where the system dynamics are relatively predictable and do not change significantly over time. However, it can be less effective when dealing with varying vehicle dynamics, road or tire condition uncertainties, or rapidly changing driving conditions. In this study, an adaptive-prediction-horizon MPC (adaptive MPC for short) is proposed and implemented to further improve the vehicle stability control performance. A prediction horizon adaptation strategy is introduced by adjusting the prediction interval in real-time based on observed vehicle behaviors, while keeping the number of predictions steps constant throughout the control process. This ensures that the control algorithm maintains a consistent view of the future and avoids introducing additional computational complexity associated with varying prediction steps. Again, the high-fidelity CarSim model was used in the simulation and the Equinox electric testing vehicle was used in the experimental study to verify the performance of the developed adaptive MPC for HVC. Both simulation and experimental results show that the developed adaptive MPC presents superior performance to track the designed yaw rate while maintaining the vehicle in a stable region compared to the fixed-prediction-horizon MPC.

The studies in this thesis contribute to advancing vehicle stability control by harnessing the power of machine learning and adaptive control strategies. These approaches enhance vehicles' control capability and lead to safer and more reliable vehicles on the road.

Acknowledgements

Foremost, I wish to express my profound gratitude to my dedicated supervisor, Prof. Amir Khajepour, whose exceptional guidance, unwavering support, and constant encouragement have been the cornerstones of my successful journey through my PhD program. Working under his mentorship has not only shaped my current academic prowess and professional vision but has also instilled in me a deep understanding of the ethics that underpin the field of engineering.

I would also like to extend my heartfelt appreciation to the entire community at the University of Waterloo and General Motors who have been instrumental in my academic pursuits. Their support and contributions have left an indelible mark on my journey, and I hold their involvement close to my heart. In addition, I am grateful for the invaluable technical guidance and insights provided by Prof. Ehsan Hashemi, Prof. Mohammad Pirani, Yubiao Zhang, Qingrong Zhao, Alireza Kasaiezadeh, Reza Hajiloo, Amin Habibnejad, Yukun Lu, and Jiaming Zhong during my PhD studies. Special recognition goes to the dedicated technicians at the Mechatronics Vehicle Systems Laboratory, specifically Jeff Graansma and Aaron Sherratt, whose efforts facilitated the successful execution of experimental tests central to my research. Their contributions have been pivotal to the development of my research.

Above all, my deepest gratitude goes to my cherished family, including my wife Ivy Wang, my parents, and my sister, for their unwavering support and unwavering encouragement throughout this arduous journey. I am acutely aware of the sacrifices they have made on my behalf, and I acknowledge that reaching this significant milestone would have been an insurmountable challenge without their unwavering presence in my life.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Motivation

Road traffic injuries cause considerable economic losses to individuals, their families, and nations. According to the statistics from World Health Organization, approximately 1.35 million people die each year in the world due to road traffic crashes [1]. In addition, road traffic injuries are the leading cause of death for children and young adults aged 5-29 years [1]. To reduce road accidents, various vehicle stability control systems, including ESC (Electronic Stability Control), ABS (Anti-lock Braking System), TCS (Traction Control System), etc., have been developed and employed on vehicles over the past three decades. Studies have shown that such techniques effectively reduce crashes [2, 3]. However, further studies and improvement of these systems should be encouraged to enhance vehicles' safety performance. From a technical perspective, these systems are usually activated when significant differences between actual and desired vehicle responses, such as side slip angle, are detected. However, in many cases, the vehicle has already reached the edge of instability by the time deviations in actual responses are measured by onboard sensors. Moreover, these situations frequently occur at high speeds or on slippery roads, where any delay in detection and proper control could result in potential accidents.

Based on the above discussion, active stability control systems need to predict dynamic vehicle behaviors to react proactively in critical situations. Model Predictive Control (MPC) has emerged as a solution to overcome the limitations of traditional vehicle stability control systems [4, 5]. In MPC, a physics-based model of the system is utilized to predict the vehicle's motion within a specific time frame, and optimal control actions are obtained by solving an optimization problem. However, MPC's effectiveness relies on having a sufficiently accurate model of the system. Vehicle stability control involves managing complex and highly nonlinear dynamics, encompassing vehicle motion, tire forces, and driver inputs. Traditional control approaches might struggle to accurately capture these intricate dynamics. In traditional MPC for vehicle stability control, a physics-based prediction model, like the single-track bicycle model, is commonly employed to forecast dynamic vehicle behaviors. Researchers have explored various vehicle dynamics models, ranging from a 2-DOF kinematic bicycle model to a more intricate 14-DOF full vehicle model, as the prediction model in MPC for vehicle stability control [6]. Using more complex physics-based models can enhance prediction accuracy, but it also introduces increased computational complexities and calibration workload. Additionally, the stability of a vehicle can be influenced by uncertain road conditions, such as variations in friction, surface roughness, or the presence of ice or water. These modeling and parameter uncertainties, including road conditions, vehicle weight changes, and tire dynamics, cannot always be precisely predicted in all situations. These uncertainties present challenges for accurate predictions and can impact the effectiveness of traditional MPC approaches.

To solve these problems, learning-based MPC, or learning MPC for short, has been proposed by some researchers [7-10]. The idea is to combine the physics-based model (the nominal model) with the learning model that can learn certain dynamics from data and thereby supplement the nominal model. Learning MPC leverages machine learning techniques to directly learn the system dynamics from data, allowing it to handle model uncertainty more effectively. Moreover, it can adapt to such changes by continuously updating the learned model using real-time data, ensuring that the controller remains effective even as the system evolves. In addition, it can learn and incorporate nonlinearities, enabling more accurate control of such systems by learning from historical data and identifying patterns, correlations, and optimal control strategies that might not be apparent through traditional modeling techniques alone. Figure 1.1 illustrates the concept of the learning MPC for vehicle control systems.

However, there are some challenges for the existing learn MPC techniques. First of all, while learning-based control approaches can offer improved performance, they often lack interpretability. Understanding and interpreting the learned models and their learning and prediction processes are crucial for safety-critical systems, such as vehicle stability control. Secondly, in existing learning-based control, relying solely on learned models might result in poor performance or instability if the model encounters scenarios that differ significantly from the training data. Thirdly, learning-based control requires large amounts of high-quality data for training accurate models. In the vehicle stability control domain, collecting such data can be expensive or impractical. On the other hand, traditional MPC utilizes a known mathematical model of the system, which provides insight into the underlying dynamics and control behavior. This interpretability can be crucial for safety, understanding system constraints, and debugging in critical applications. In this study, to address the drawbacks of model-based MPC and learning-based control approaches, a hybrid data/model-based MPC is proposed. It includes both a physics-based model and a learning data-based model to better predict the vehicle dynamics in the horizon time and further optimize the control actions.

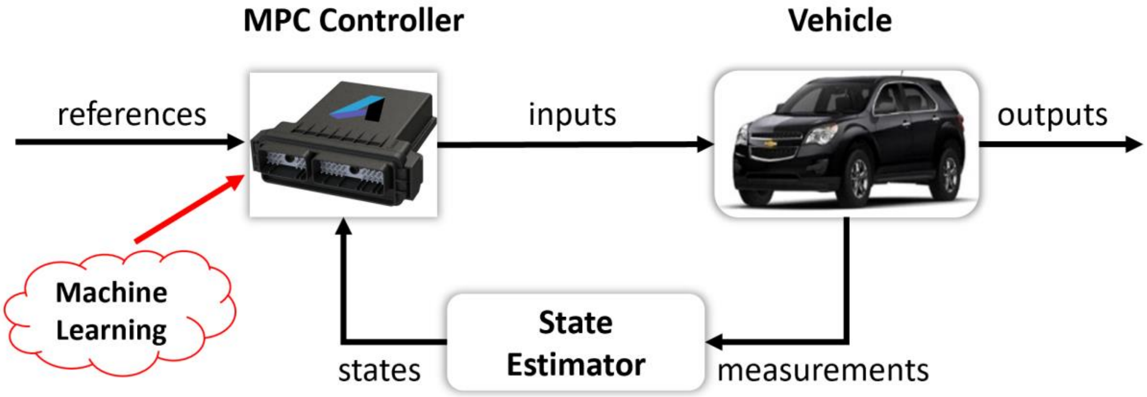


Figure 1.1: The concept of a learning MPC for vehicle control systems

1.2 Objectives

This study is to develop a hybrid physics-based and data-based MPC control technique for vehicle stability control systems. The hybrid learning MPC will benefit from deterministic and robust physics-based predictive control structures and take advantage of data-based systems and their learning abilities.

(1) **Improved Control Performance**: One of the primary objectives is to enhance the control performance of vehicle stability control systems. By combining data-driven techniques with known mathematical models, the proposed hybrid learning MPC aims to achieve superior stability, handling, and maneuverability of vehicles, leading to enhanced safety and improved overall performance than traditional MPC approaches.

(2) **Robustness to Uncertainties**: The hybrid learning MPC seeks to develop control strategies that can handle uncertainties and variations in vehicle dynamics, road conditions, and external disturbances. By incorporating both the known model and data-driven adaptation, it aims to provide robust control actions that can effectively deal with uncertain and dynamic driving environments. Some harsh driving scenarios such as double lane change on a slippery road condition will be applied to evaluate the performance of the proposed controller. In such conditions, the road condition is unknown to the controller and thus, maintaining the vehicle stability is a challenging task for a traditional MPC approach.

(3) **Interpretability and Explainability**: Many learning-based control approaches work like a black box which is hard to interpret and understand even by professionals. While the hybrid approach combines data-driven techniques with known models, it still strives to provide interpretability and

explainability. Understanding the learning process of the control algorithm is crucial for debugging and safety for vehicle stability control systems. This study has the objective to develop a hybrid learning MPC method that offers transparency and interpretable models to facilitate understanding and validation.

(4) **Online Learning Capability**: Vehicle dynamics and operating conditions can vary over time due to factors such as tire wear, aging, or changes in road conditions. It is not realistic to collect a training dataset that can cover all driving scenarios. This study aims to develop a hybrid learning MPC that has online learning capability instead of using a pre-trained learning model. Online learning allows the learning MPC to continuously adapt and update its models based on real-time data, ensuring accurate and up-to-date representation of the vehicle dynamics. Also, online learning enables the learning MPC to be more flexible and scalable. It can easily incorporate new data and adapt to changes in the system, allowing for the integration of different vehicle models, configurations, or control objectives without the need for retraining the entire model from scratch.

(5) **Real-Time Implementation**: The developed controller should be able to run in real-time for holistic vehicle control. It is aimed to develop control algorithms that can be implemented in real-time, enabling their practical application in vehicle stability control systems. Many learning-based control approaches can only be verified through simulation due to the challenges of high computation load caused by the introduction of machine learning. Efficient computation is an important objective to ensure that the control system can respond quickly to changing driving conditions and maintain stability in real-world scenarios.

By studying hybrid data/model-based MPC in vehicle stability control, it is aimed to achieve a balance between the benefits of model-based control (interpretability, robustness) and the advantages of data-driven techniques (adaptability, performance). Ultimately, the goal is to develop advanced control strategies that improve vehicle safety in a wide range of driving conditions.

1.3 Thesis Outline

This thesis is structured as follows. In the second chapter, it provides an overview of the existing research in the field regarding HVC systems and various MPC control algorithms with emphasis on learning MPC techniques. It discusses traditional physics model based MPC approaches and their limitations, presents the evolution of traditional MPC to adaptive MPC, and explores learning MPC techniques and their advantages. This chapter reviews relevant studies and advancements in various

MPC approaches in the area of vehicle stability control, highlighting the key findings, methodologies, and contributions of previous research. It establishes the foundation for this thesis by identifying the research gap and justifying the need for studying hybrid learning MPC in the context of vehicle stability control.

The third chapter introduces the background knowledge of this study. In this chapter, it describes the MPC control theory, and the basic structure of existing learning MPC techniques. This chapter also introduces Gaussian Process Regression (GPR) and its application in learning the system dynamics and uncertainties in learning MPC. It highlights the advantages of GPR in capturing nonlinearity, handling data-driven models, and providing uncertainty estimates. This chapter establishes the foundation for the subsequent research on applying GPR-based learning MPC in vehicle stability control.

The fourth chapter focuses on the development of a GPR-based hybrid learning MPC for vehicle yaw stability control. It outlines the steps involved in designing the controller, including the formulation of the objective function and constraints to optimize control actions for enhancing yaw stability. This chapter also presents the selection of relevant system variables in the learning model, the developed data management and model authentication strategy, the training of GPR models using collected data, and the integration of the GPR models into the MPC framework.

The fifth chapter discusses simulation and experimental results for the developed hybrid learning MPC. A high-fidelity CarSim model is developed and verified to represent the vehicle dynamics in the simulation. The designed hybrid learning MPC controller is implemented in Matlab and evaluated through the Matlab/Simulink CarSim co-simulation under a series of driving maneuvers. Experimental verification of the developed hybrid learning MPC on a Chevy Equinox electric testing vehicle is also presented.

In chapter six, an adaptive-prediction-horizon MPC is designed to further optimize the performance for HVC. The performance of the developed adaptive MPC controller is assessed using both simulation and experiments. Again, the CarSim vehicle model is used in Matlab/Simulink co-simulation and the Chevrolet Equinox electric testing vehicle is used for experimental evaluations. Simulation and experimental results under various driving maneuvers are presented to gauge the effectiveness and adaptability of the controller.

The last chapter summarizes the key findings and contributions of this study. It highlights the effectiveness of the proposed controller in improving yaw stability and handling performance based on simulation and experimental results. The chapter also discusses the limitations and challenges encountered during the research. It suggests potential future directions to handle these challenges.

Chapter 2
Literature Review

This chapter presents a comprehensive review of the literature on Holistic Vehicle Control (HVC) systems and various Model Predictive Control (MPC) schemes employed for achieving stability control of vehicles. By conducting this literature review, the chapter aims to provide an overview of the current state-of-the-art in HVC systems and MPC schemes, highlighting the key findings and trends in this area. It serves as a foundation for the subsequent sections of the study, enabling the development of novel MPC approaches and methodologies for HVC.

2.1 Vehicle Stability and Holistic Vehicle Control

Vehicle stability analysis is to find the conditions for stability of a vehicle, or to find regions inside which stability is inherently guaranteed [11-17]. Determining the stability boundary in a so-called phase plane is a prerequisite for the study of HVC [18-21]. Vehicle sideslip angle (β), lateral velocity (ν), and yaw rate (r) are three important variables to describe the status of vehicle stability, and the β-r phase plane or the ν-r phase plane as vehicle lateral stability criterion have been widely studied. For example, the authors in [21] studied the effects of the steering on the vehicle stability based on the single-track vehicle dynamics model. Steering input variations lead to substantial alterations in the phase portrait trajectories. Figure 2.1 illustrates the open-loop β-r dynamics of the research vehicle at a speed of 10 m/s and a road friction value of 0.55 for different steering angles. When the steering angles are set at 0°, 5°, and 10°, there is a single stable equilibrium point. Additionally, in each of these scenarios, two saddle equilibria can be observed. These points represent the yaw rate and sideslip angle that would occur during a right or left-handed drift maintained at the corresponding steering angle. For the given vehicle parameters, there is no stable equilibrium point when the steering angle is set to 15°. However, there is one unstable equilibrium point located on the line of maximum steady-state yaw rate.

The authors in [22] conducted a series of simulation studies to analyze the β-r phase plane characteristics under various vehicle speeds, road friction coefficient, and front-wheel steering angle. The results showed that the stable region on the β-r phase plane could be divided into two types. In normal driving conditions, the stability domain corresponded to the curve type, while it changed to the diamond type in critical driving conditions. The authors in [23] presented a study on controlling the lateral dynamics of an autonomous vehicle confronting a sudden obstacle. The controller was

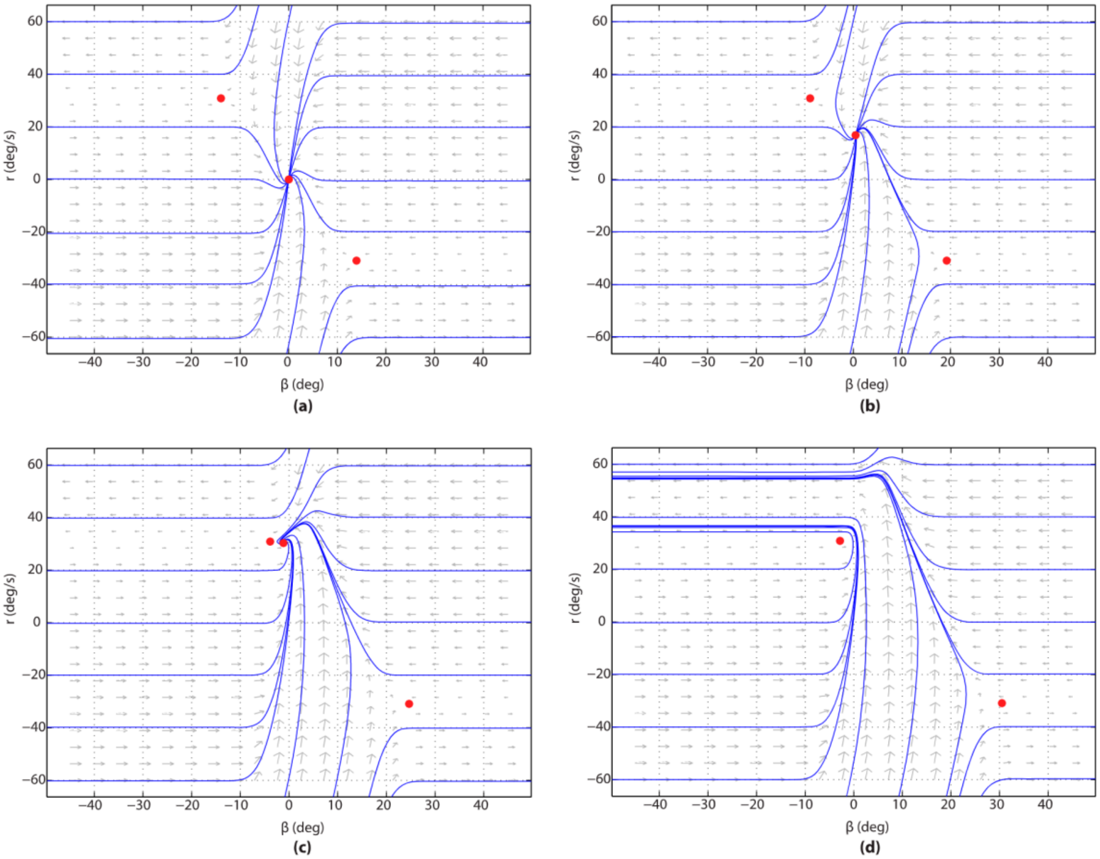


Figure 2.1: Yaw rate – sideslip phase portrait at (a) 0° (b) 5° (c) 10° (d) 15° steering angle [21].

prioritized as (1) collision avoidance, (2) vehicle stability, and (3) path tracking. The limits on yaw rate and lateral velocity were defined, and they created a parallelogram (the ν-r phase plane) in the state space. The area enclosed by this parallelogram was regarded as the secure and stable region for velocity states, lateral velocity, and yaw rate. This stable region could be incorporated into an MPC controller as a safety constraint to confine the projected states within the secure region and guarantee vehicle stability. The stable region inside the parallelogram was included within the controller as safe constraints to ensure vehicle stability.

To enhance vehicle stability, HVC is commonly used [24]. It involves coordinating and integrating various control systems within a vehicle to achieve optimal handling, traction, and stability in different driving conditions. The primary objective of HVC for vehicle lateral stability is to reduce the deviations of the actual yaw rate from the driver intended yaw rate while limiting the vehicle sideslip angle to prevent vehicle spin and preserve some yaw moment gain [24]. There are several types of systems proposed and developed for HVC [25]:

(1) Differential braking (DB): DB utilizes the anti-lock braking system (ABS) to apply hydraulic braking torque between the right and left wheels to control yaw moment. This approach employs a strategy where, in the event of vehicle oversteer, the wheels on the outer side of the vehicle are applied with braking force to lessen the yaw moment acting on the vehicle's center of gravity, thus regulating the sideslip angle. Conversely, if the vehicle understeers, the inner wheels are subjected to braking force to enhance the steering yaw moment and enhance the vehicle's maneuverability.

(2) Active front steering (AFS) or active rear steering (ARS): AFS and ARS modify the driver's steering angle input and add a correction steering angle to the wheels. An example is shown in Figure 2.2 that depicts the workflow of an AFS system incorporating variable gear ratio steering [26]. By determining an appropriate superposition angle between the AFS actuator and the driver's steering angle, the vehicle's yaw rate and sideslip angle can be regulated to enhance yaw stability. However, designing a controller for effective vehicle yaw enhancement with desirable tracking performance is highly challenging due to the high nonlinearity and uncertainties inherent in the vehicle-road system dynamics. The complex tire dynamic characteristics contribute to the system's high nonlinearity, while uncertainties arise from simplifications made in road-vehicle modeling, unpredictable environmental conditions, and unmeasurable parameters such as tire cornering stiffness coefficients and road surface adhesion coefficients. Addressing these nonlinearities and uncertainties typically requires a complex computational model, which increases computational complexity and impacts the controller's response time. Given that the controller's time interval is usually limited to a few milliseconds for each loop execution, the computational efficiency of the control algorithm significantly influences the controller's response time.

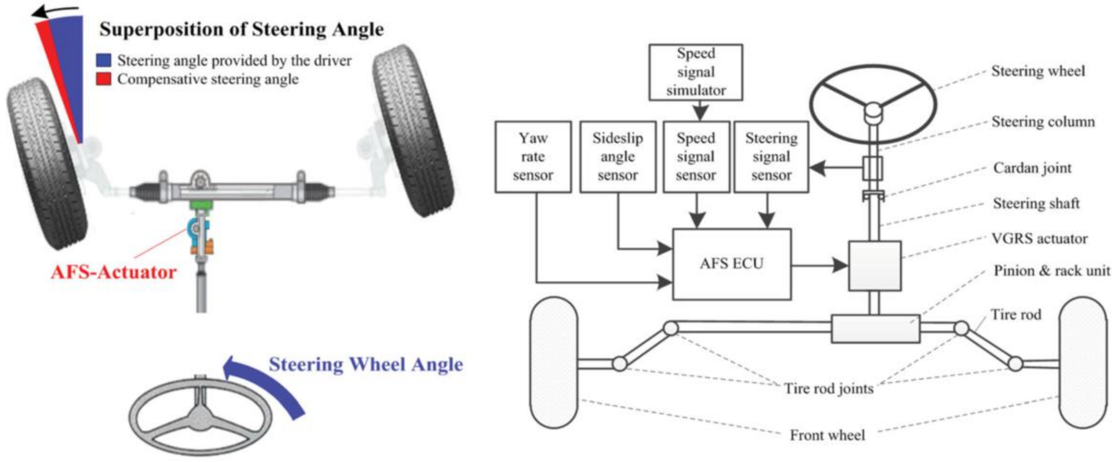


Figure 2.2: Functionality of an AFS system [26].

(3) Torque vectoring (TV): TV is an advanced technology employed in vehicle stability control systems to enhance a vehicle's handling and stability during cornering and challenging driving conditions. Instead of relying solely on traditional methods like braking individual wheels, torque vectoring actively adjusts the distribution of torque (rotational force) to each wheel, providing more precise control and improved performance. This technology is particularly useful in vehicles with all-wheel drive (AWD) or four-wheel drive (4WD) systems.

These stability control systems have received much attention from researchers in recent years. The authors in [27] studied the lateral stability of vehicles using AFS and DB. The controller could be configured to only use DB or AFS or use both DB and AFS. Experimental results on an electric Chevrolet Equinox showed that both methods were effective in controlling the vehicle's sideslip angle within an acceptable range in an acceleration-in-turn (AIT) maneuver on wet top sealed asphalt. The authors in [28] addressed the integrated longitudinal and lateral vehicle control problem using TV as the actuator. The experimental tests on an all-wheel-drive electric vehicle in double lane change (DLC), full-throttle launch (FTL), and AIT on slippery road conditions showed that the integrated longitudinal and lateral holistic vehicle control using TV has the capability in holding the vehicle stability in a safe zone. In comparison to DB, TV can provide stability control with less intrusive measures. Instead of applying the brakes to specific wheels, TV achieves stability adjustments through the distribution of torque. This can lead to a smoother and more natural driving experience. AFS systems involve sophisticated mechanical and electronic components, including actuators, sensors, and control units. The complexity of AFS system can lead to increased manufacturing and maintenance costs, which may impact the overall cost of the vehicle. In this study, TV is used as the actuator for vehicle stability control.

2.2 MPC Schemes for HVC

In the previous section, literature on vehicle stability analysis and HVC systems with different actuators was reviewed. The design of such a control system is an active research area. Extensive reviews on various mathematical models, such as PID control, Sliding Mode Control (SMC), and Linear Quadratic Regulator (LQR), were presented in [29-31]. PID and SMC control are not optimal in nature. While LQR does provide optimal control, it only works for unconstrained optimization problem which is a limitation for vehicle control as the vehicle dynamics are always bounded within the designed operating range. MPC on the other hand covers all the conclusions made under one control design and becomes a suitable control algorithm for vehicle stability control [32-35].

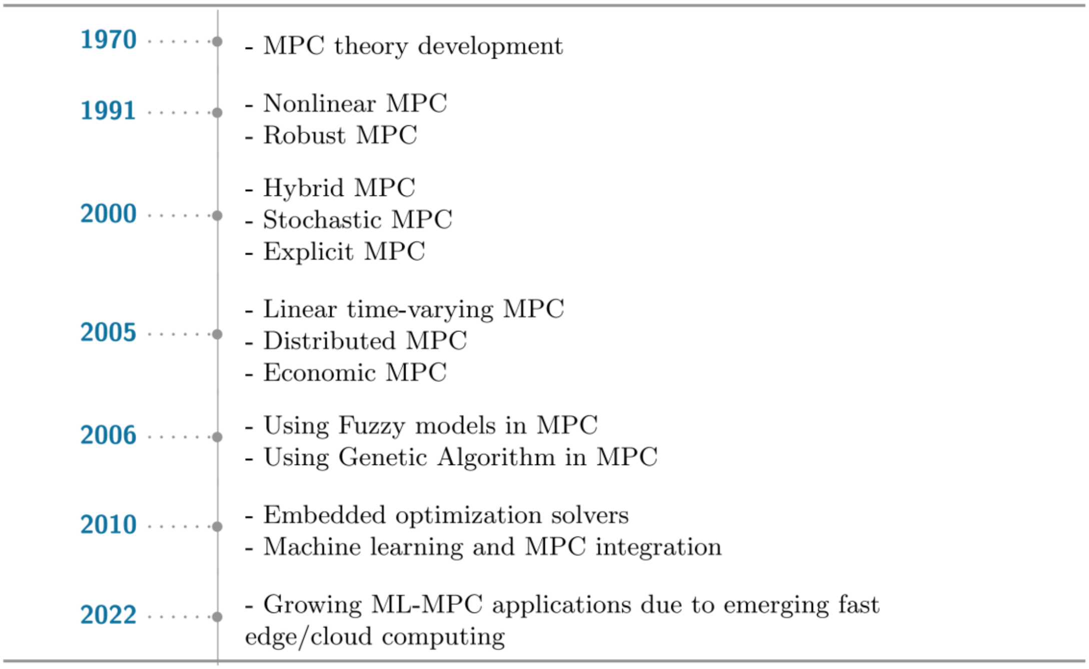

1970
- MPC theory development
1991
- Nonlinear MPC
- Robust MPC
2000
- Hybrid MPC
- Stochastic MPC
- Explicit MPC
2005
- Linear time-varying MPC
- Distributed MPC
- Economic MPC
2006
- Using Fuzzy models in MPC
- Using Genetic Algorithm in MPC
2010
- Embedded optimization solvers
- Machine learning and MPC integration
2022
- Growing ML-MPC applications due to emerging fast edge/cloud computing

Figure 2.3: Timeline of major theoretical developments of MPC [36].

A timeline of theoretical developments of MPC is shown in Figure 2.3. As it shows, MPC has evolved from linear to nonlinear formulations, incorporating machine learning to enhance adaptive capabilities, and leveraging cloud computing for real-time learning and decision-making. As MPC is an optimal control technique which is based on the designed prediction model, it can accommodate vehicle nonlinearities and Multi-Input–Multi-Output (MIMO) models in its design. For vehicle stability control systems, MPC schemes have been widely studied due to their capability of predicting vehicle behaviors using a model and explicitly considering the state and actuator constraints. There are variants of MPC schemes used in the study of vehicle stability control systems.

2.2.1 HVC Using Linear MPC

For linear MPC, the prediction is based on linear system models. In many cases, however, system models are nonlinear. In these cases, linear time-varying (LTV) MPC and linear parameter-varying (LPV) MPC based on successive online linearization of a nonlinear system model around the current operating point at each time step are widely used. For LTV-MPC, the model can also change over the prediction horizon at each sampling time. The authors in [37] presented a double linear model predictive control (DLMPC) structure for vehicle yaw stability control using AFS. The DLMPC structure consists of both linear MPC and LTV-MPC. The linear MPC part using a linear tire model controls the vehicle when the tire force is in the stable region, while the LTV-MPC part using the Pacejka tire model is activated and takes over the vehicle control when the tire force enters the unstable region. There are two advantages of the DLMPC structure: first, its computational burden is lower than nonlinear MPC by solving a convex quadratic problem; second, it avoids the problem of low utilization of the tire force caused by the constraints in a conventional linear MPC structure. Simulation results under DLC and the sine with dwell maneuver tests showed the effectiveness and feasibility of the DLMPC controller. Barbarisi et al. [38] proposed a vehicle dynamics control system based on LTV-MPC. The sampling time and prediction horizon were set to 0.25 seconds and 5 steps, respectively. The authors focused on controlling the lateral motion of the vehicle, assuming constant longitudinal dynamics. Although the controller successfully passed the standard ISO 19365:2016 Sine with Dwell test, it exhibited oscillatory behavior when operating near the constraint boundaries.

The authors in [27] proposed a linear MPC structure for vehicle lateral stability control using coordinated AFS and DB. The prediction model forecasts the prospective responses of the vehicle's lateral velocity, yaw rate, and tire slip angles over the prediction horizon, and then provides proactive control inputs to control the vehicle sideslip angle. Experimental results on an instrumented test vehicle showed satisfactory performance in various combinations of AFS and DB. Besides, the measured computational time of the controller using linear MPC was safely below the sample time of the controller. Choi and Choi [4] proposed an MPC controller for vehicle stability using an extended bicycle model incorporating AFS and DB. Their approach involved a prediction model that accounted for the lagged characteristics of actuator dynamics and tire forces, both modeled as first-order lag systems. The control action was determined by calculating the steering wheel angle and yaw moment correction. However, this approach resulted in increased computational time and performance degradation due to the need to solve two separate optimization problems. As a result, the original intention of having a single integrated controller for vehicle control was compromised.

2.2.2 HVC Using Nonlinear MPC

While linear MPC is effective for systems with relatively simple dynamics and linear relationships between inputs and outputs, it has limitations when dealing with highly nonlinear systems or when precise control of nonlinear behavior is required. This is where nonlinear MPC comes into play. Nonlinear MPC is characterized using nonlinear system models directly in the prediction. In [39], the authors developed a nonlinear MPC formulation for a DLC maneuver using a bicycle model as the system model, with the steering wheel angle as the control input. The nonlinear MPC controller successfully stabilized the vehicle at a speed of 7 m/s but failed to do so at 10 m/s. The authors concluded that integrating steering and braking control could enhance the controller's performance. Building on this, Falcone et al. [40] designed an nonlinear MPC-based control system with a prediction horizon of 1 second to optimize the combination of braking and steering for obstacle avoidance during a DLC maneuver. They employed a 10 Degree of Freedom (DoF) planar vehicle model as the prediction model, considering the vehicle's longitudinal and lateral velocity, heading angle, yaw rate, and global position coordinates as the first six DoF, while the remaining four DoF represented the individual dynamics of each wheel. The tire characteristics were modeled using a Pacejka model. The control inputs for this controller were the front steering angle and the brake torque values for each wheel. The controller successfully passed the test at a speed of 14 m/s. However, due to the increased number of model parameters, the controller's tuning was not optimal, resulting in high oscillations in the steering angle. Moreover, the controller's computational time was impractical for real-time applications, as it took approximately 15 minutes to complete a 12-second simulation. The authors in [41] studied HVC based on explicit nonlinear MPC and investigated the impact of the fidelity of the prediction model on controller performance. For explicit MPC, the solution is derived offline, stored in the controller, and then evaluated online with reduced computational complexity and delays. Controllers with different prediction models were assessed, including using a linear tire model or various nonlinear tire models. CarMaker simulation results showed that the developed controllers met the UN/ECE-R 13H performance requirements for the sine-with-dwell test. In addition, the explicit nonlinear MPC method showed significantly lower computational time than the implicit approach for the same controller formulation.

The authors in [42] proposed a nonlinear MPC method integrating AFS and an additional yaw moment for vehicle lateral stability control. The method was implemented on the field-programmable gate array (FPGA). The particle swarm optimization (PSO) algorithm was used to solve the nonlinear optimization problem, yielding parallel search capabilities. The results showed that the proposed nonlinear MPC method achieved better control performance than the direct yaw moment control method under various running conditions. Moreover, good robustness was achieved for the nonlinear MPC method regarding variations of the longitudinal velocity and tire-road friction coefficient within a suitable range. In the study conducted by Yi et al. [43], two MPC controllers were developed for collision avoidance. These controllers utilized a combination of steering and braking as the control action. The authors employed a nonlinear bicycle model for the design of the nonlinear MPC and linearized the model for the linear MPC. The nonlinear constraints were approximated into linear form. Instead of directly calculating brake torques, the MPC control action was the commanded longitudinal acceleration, which was then used to calculate the required brake torques using another logic. The controllers were tested in a single-lane-change maneuver at a speed of 70 km/h with a target lateral displacement of 2 m. The sampling time and prediction horizon were set at 0.06 seconds and 25 steps, respectively. Although the collision was avoided, both control strategies exhibited an overshot of approximately 35% in lateral position tracking, resulting in poor tracking performance. Additionally, the nonlinear MPC design was not real-time feasible, with a mean computation time ranging from 4 to 8 seconds.

Li et al. [44] presented a real-time NMPC-based controller for four-wheel independent motor-drive electric vehicles, aiming to enhance longitudinal and lateral stability under extreme driving conditions. A combined-slip tire model was used to develop a stability controller for low friction coefficient surfaces. The wheel slip ratios and slip angles were employed as virtual control inputs to achieve slip control, lateral stability control, and handling performance improvement, while considering multiple safety constraints. The control performance was evaluated through co-simulation with MATLAB/Simulink and CarSim, demonstrating efficient optimization and enhanced vehicle stability.

2.2.3 HVC Using Adaptive MPC

Both linear and nonlinear MPC rely on accurate predictive models, and when these models are not perfectly known or are subject to changes, their performance can degrade. Adaptive MPC is introduced to overcome these limitations by continuously updating the predictive model during operation. In [45], an adaptive MPC scheme was proposed for autonomous vehicles. The proposed controller utilized a holistic structure with a decoupled linear integrated model to ensure real-time performance. A weight adaptive mechanism was introduced to improve handling ability in various driving conditions, including extreme conditions. A multi-model adaptive law was designed to address uncertainties in tire cornering stiffness. Numerical results demonstrated the effectiveness and superiority of the proposed control method in both normal and extreme driving conditions. The holistic structure, weight adaptive mechanism, and multi-model adaptive law contribute to excellent performance and road safety.

Focusing on the problem of rollover instability in commercial vehicles under extreme steering conditions, Zhou et al. [46] used adaptive MPC with time-varying weights and constraints to control the roll stability of the vehicle. A layered control strategy was proposed to address the influence of wheel lifting on roll dynamics, incorporating engine torque limit, differential braking, and active front-wheel steering. The algorithm improved roll angle estimation accuracy and robustness, and incorporated driver behavior in roll stability control. Hardware-in-loop tests validated the effectiveness of the proposed algorithm in preventing vehicle rollover.

In [47], the authors presented an adaptive MPC scheme for improving yaw stability in four-wheel-independently-actuated electric vehicles. An autoregressive with exogenous input model was developed for vehicle prediction, and its time-varying parameters were identified using an unbiased estimation system based on instrumental variable method. The adaptive MPC scheme was proposed for direct yaw moment control, and a multi-objective optimization method was used for torque allocation. Simulation results demonstrated that the proposed scheme achieved smoother yaw rate signal, reduced computational time, and improved vehicle stability compared to conventional dynamics-model-based MPC.

Lin et al. [48] proposed an adaptive MPC for path tracking in changing working conditions that incorporated estimators for tire cornering stiffness and road friction coefficient, as well as a control parameter selection module. The estimated parameters were used to update the vehicle model and road adhesion constraints, improving path tracking accuracy and stability. Simulation results demonstrated the effectiveness of the proposed controller in adapting to changing working conditions, particularly speed and road conditions.

In [49], an adaptive MPC strategy was proposed for path following in four-wheel independent drive automated vehicles. The strategy included an estimator for real-time estimation of tire cornering stiffness, a modified tube-based MPC method for path following under disturbances, and a torque distribution algorithm for generating compensation yaw moment. Simulation and experimental results demonstrated that the proposed strategy improved path following accuracy by 41.6% and 60% compared to traditional MPC and LQR methods, respectively. The strategy also ensured vehicle stability during severe maneuvers.

Liang et al. [50] proposed a novel control approach that combines two adaptive MPC techniques to improve the longitudinal and lateral motion control of a vehicle. The scenario adaptive MPC automatically decided the local motion behavior (e.g., driving along the global path, car-following, lane-change) to ensure road safety and desired velocity requirements. The weight adaptive MPC tracked the global path and conducted lane-change maneuvers based on the scenario adaptive MPC command. Case studies demonstrated the effectiveness of the proposed algorithm in complex driving conditions. However, the algorithm was not personalized for different driving styles and expanded to handle off-road and crossroad scenarios.

The authors in [51] developed an adaptive MPC for traction control. In the research, the prediction model consisting of wheels and vehicle longitudinal dynamics was discretized using a semi-implicit Euler approximation. The maximum coefficient of friction was estimated by a sensitivity-based joint Unscented Kalman Filter (UKF) with a highly nonlinear two-track vehicle model representing longitudinal and lateral as well as roll and pitch dynamics. Experiments were carried out with a Volkswagen Golf GTE plug-in hybrid electric vehicle at full throttle maneuvers on low friction road conditions. The results showed that the simultaneous estimation of the maximum coefficient of friction using UKF and the adaptive control of the wheel slip resulted in significantly improved traction control performance. The authors in [52] designed an adaptive MPC for speed control of heavy-duty vehicles by coordinating the use of compression brakes and friction brakes on downhill slopes. A Recursive Least Square (RLS) scheme with forgetting was applied to the controller to update the estimates of vehicle mass and road grade at each sampling time. Simulation results showed that the speed tracking performance was improved noticeably, and the oscillatory behavior of speed control observed in regular MPC was reduced.

While adaptive MPC has been an active area of research, based on the literature review, the focus has primarily been on adapting model parameters and updating the control model in real-time based on online data. However, there is relatively less research on adapting the prediction horizon for MPC. The prediction horizon in MPC refers to the length of the future time steps over which the controller predicts the system's behavior and calculates the optimal control actions. An adaptive prediction horizon would involve dynamically adjusting the length of this prediction window based on real-time data and system conditions. In this study, this will be investigated in the context of vehicle stability control.

2.3 Learning MPC

Traditional MPC schemes discussed above are of no learning capability or limited learning capability (e.g., adaptive MPC), and they are facing challenges to modeling emerging complex systems, especially those with uncertainties. In recent years, successes of machine learning technologies in many application domains [53-59], as well as the availability of increased sensing and computational capabilities in modern control systems [60-67], have raised a growing interest in data-driven learning-

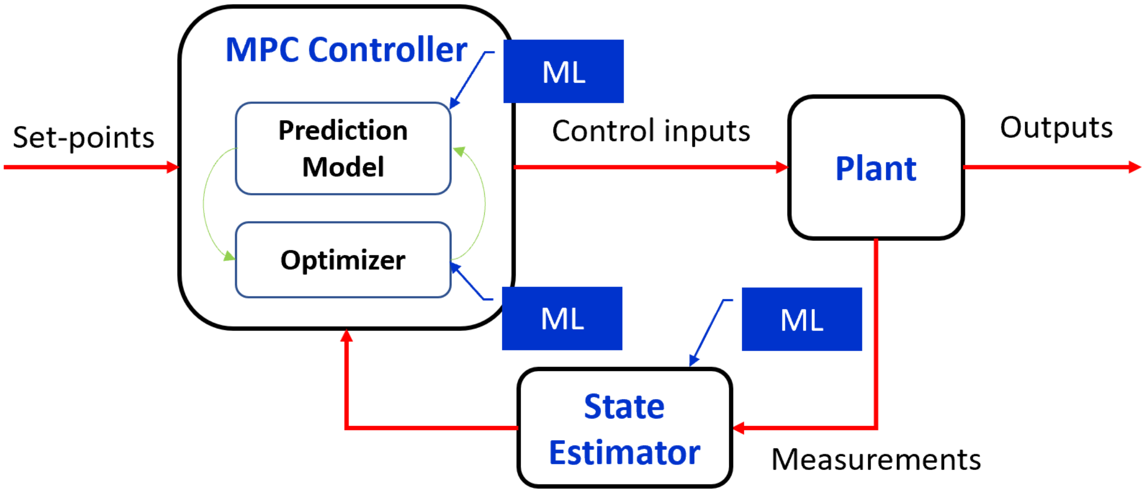


Figure 2.4: Integration of machine learning in MPC

based control techniques [68]. However, learning by itself cannot ensure safety and stability. The learning MPC scheme, which integrates machine learning techniques into the robust MPC framework, is becoming a hot spot of research in automatic control areas. There are diverse research areas depending on where the machine learning (ML) takes place in the MPC structure. They include learning the prediction model, learning control laws, utilizing learning in state estimation, and utilizing MPC for safe learning, shown as Figure 2.4.

2.3.1 Learning the Prediction Model

The performance of traditional MPC relies heavily on sufficiently accurate prediction models which typically use physical principles as representations of the system dynamics. In some systems, obtaining an accurate and computationally efficient model for prediction may be challenging due to the complexity of the underlying dynamics. Learning the prediction model using machine learning techniques allows for approximating complex system behaviors and making accurate predictions without the need for explicit model equations. Many real-world systems exhibit nonlinear behavior and uncertainties that are difficult to capture with traditional analytical models. Learning-based approaches can effectively handle such nonlinearity and uncertainty, providing more accurate and robust predictions for control. There are some studies focusing on learning the prediction model.

In [69], the authors focused on the development of a learning-based nonlinear MPC approach that emphasized the use of data-based models, rather than fixed physical models. The proposed approach utilized nonlinear prediction models based on the Nonlinear Set Membership (NSM) learning method and considered an online learning setting where the learning model is updated repeatedly during the controller runtime. The advantages of using these data-based models were highlighted for robust stability, recursive feasibility, and improved control performance. By utilizing these models, the approach achieved better closed-loop performance while ensuring stability and feasibility, even in the presence of estimation errors.

The authors in [8] proposed a provably safe and robust learning MPC framework. The main insight of this learning MPC structure is that safety and performance can be decoupled by maintaining two models of the system. The first one is an approximate model with bounds on its uncertainty, and the second one is an updated model by statistical methods. With theoretical proof, performance is improved by solving an optimization problem using the second model with learning capabilities, as well as the safety and robustness can be guaranteed by checking whether the same control inputs keep the first model with bounds stable when it is subject to uncertainties. In [10], the authors implemented this learning MPC for an Heating, Ventilation, and Air Conditioning (HVAC) system, and experiments showed that the learning MPC method saved an estimated 30%-70% of energy. In [70], they presented the robustness of this learning MPC structure to mis-learning through experiments using a quadrotor to catch a ball thrown with an unknown trajectory.

The authors in [71, 72] presented a learning-based nonlinear MPC algorithm for an autonomous mobile robot to reduce path-tracking errors over repeated traverses along a reference path. The proposed learning MPC algorithm used a simple robot model as a priori and a learned disturbance model to capture the uncertainties. Gaussian process regress (GPR) was used to model disturbances based on experience collected during previous traversals as a function of system states, control inputs, and other variables. Two experiments on two significantly different robots demonstrated the system's ability to handle unmodelled terrain and robot dynamics and also to interpolate and extrapolate from learned disturbances. The experimental results showed that the learning MPC approach was flexible and effective at reducing path-tracking errors based on learned experience, as well as robust and safe by applying restricted constraints to the mean predicted sequence.

Here are some areas of research where learning MPC is being explored in the context of vehicle control. The authors in [73] proposed a two-layer learning MPC method for yaw stability control, addressing the model-vehicle mismatch challenge. The upper layer used Gaussian Process regression to compensate for model mismatch, while the lower layer distributed braking torque based on a new weight distribution rule. Simulation results showed that the proposed method improved stability tracking and reduces energy consumption compared to traditional methods. Kabzan et al. [74] presented a learning-based MPC approach for autonomous racing, using GPR to enhance a simple nominal model and improve racing performance. Constraints were dynamically tightened based on residual uncertainty to ensure safe driving behavior. Data points for GPR predictions were selected online using an information gain criterion, enabling continuous learning. Testing on a full-size AMZ driverless car showed a 10% reduction in lap times while maintaining safety.

2.3.2 Learning the Control Laws

While the prediction model is a key element for MPC, other elements in the control systems, such as the cost function and constraints, also play an important role in the control performance. There are some studies focusing on learning control laws. In [75], the authors proposed a framework to augment an existing MPC design with active learning that is achieved by employing a user-defined learning cost function with the aim to improve model knowledge and reduce uncertainty through model adaptation. The framework is applicable to various nonlinear MPC designs and ensures desired performance bounds for the resulting closed loop. The practicality of the proposed approach is demonstrated using a numerical example.

Since a closed-form expression of the control performance as a function of controller parameters of MPC is typically not available, in a traditional MPC, selecting such controller parameters is nontrivial and usually accomplished by trial and error. This makes such a problem ideal for the application of learning techniques. The authors in [76] presented a new approach to tune the parameters in MPC using machine learning techniques. In their study, data-driven Bayesian optimization was used to tune the prediction horizon in MPC. Numerical simulation for a pendulum on a cart showed that using this approach, the prediction horizon can be optimized from data, and satisfactory controllers can be designed with no knowledge of the system dynamics.

The authors in [77, 78] proposed a reinforcement learning (RL) approach to select control parameters in MPC for robot control. Finite Action-Set Learning Automata (FALA), a theme of the RL techniques, was used to tune the weighting parameters in the MPC cost function. Experiments were conducted on both ground and aerial robots using this approach, and the results showed that the overall number of parameters that needed to be tuned in MPC was reduced.

2.3.3 Using MPC for Safe Learning

While machine learning techniques have shown great potential in many areas, the transfer of machine learning techniques into real-world applications, especially safety-critical systems, is challenged due to that most of them cannot ensure that safety constraints under physical limitations are met. MPC techniques can be used to address such a problem due to its robust structure that explicitly considers constraints for states and control inputs. The main idea of this line of research is to decouple the optimization of the cost function, which can be solved using machine learning techniques, from the requirement of constraint satisfaction, which is addressed using MPC. As pure learning-based algorithms, such as RL, do not directly consider hard system constraints on states and outputs during exploration of different control actions or policies, MPC is utilized to enforce these constraints. Currently, the implementation of this method has been limited to simulation studies, including chemical process and automotive. Although this method has not yet been widely used, it has the potential to address the safety issues associated with learning-based controllers, particularly in safety-critical industries.

Previous research has discussed the use of machine learning to enhance MPC performance or compensate for its drawbacks. This section presents a combination of MPC and machine learning, in which MPC facilitates learning-based controllers to ensure constraint satisfaction. Despite the potential of learning-based techniques in optimizing control laws, most of these techniques cannot guarantee hard safety constraints, which are often critical for engineering applications. As violations of constraints, such as plant states and output constraints, can occur during learning iterations, MPC is integrated with learning-based approaches to address this limitation [68].

Figure 2.5 illustrates the process of combining MPC and machine learning to safely learn a learning-based controller. In this figure, $u_{L,k}$ denotes the plant input from the learning-based control at time k. The learning process ensures constraint satisfaction and achieves this by either rejecting $u_{L,k}$ or modifying it to minimize the difference between u_k and $u_{L,k}$. A safety filter, $\pi^S\left(x_k, u_{L,k}\right)$, is then obtained through an MPC framework to generate an optimal safe trajectory of u_k.

RL is a commonly used machine learning technique for safe learning, as it enables the control action to be learned directly from experiments. However, most RL methods do not explicitly consider state

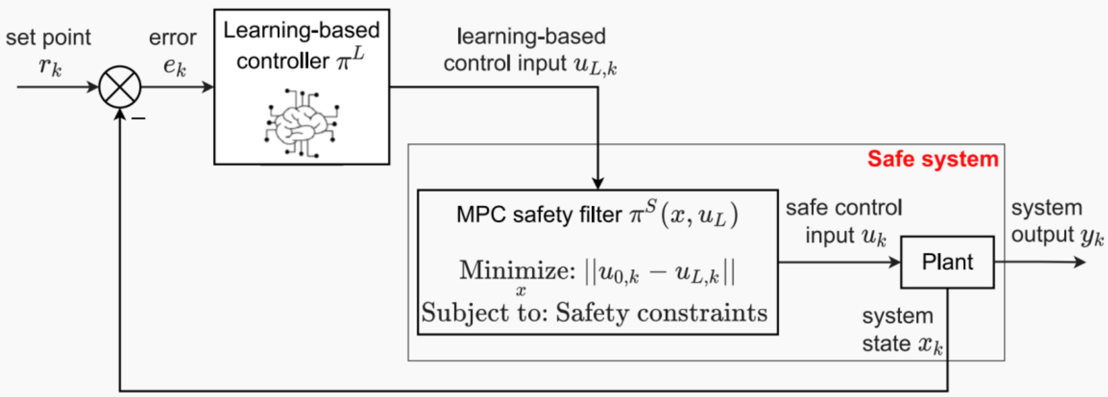

set point
r_k
error
e_k
Learning-based controller π^L
learning-based control input $u_{L,k}$
Safe system
MPC safety filter $\pi^S(x, u_L)$
Minimize: $\min_x ||u_{0,k} - u_{L,k}||$
Subject to: Safety constraints
safe control input u_k
Plant
system output y_k
system state x_k
−

Figure 2.5: Schematic of MPC for a safe learning controller algorithm [36]

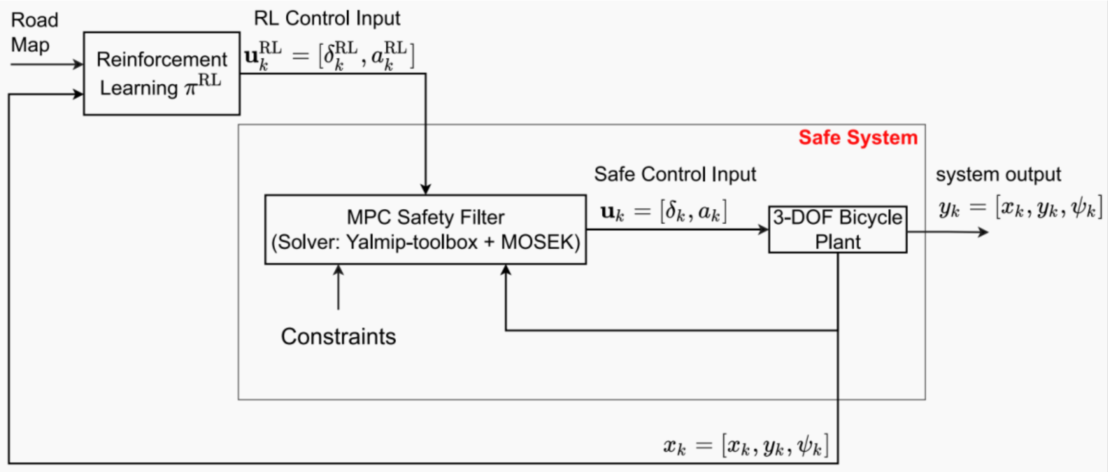

Road Map
Reinforcement Learning π^{RL}
RL Control Input
$\mathbf{u}_k^{\mathrm{RL}} = [\delta_k^{\mathrm{RL}}, a_k^{\mathrm{RL}}]$
Safe System
Safe Control Input
$\mathbf{u}_k = [\delta_k, a_k]$
MPC Safety Filter
(Solver: Yalmip-toolbox + MOSEK)
3-DOF Bicycle Plant
system output
$y_k = [x_k, y_k, \psi_k]$
Constraints
$x_k = [x_k, y_k, \psi_k]$

Figure 2.6: Example of MPC for safe learning controller for RL [79]

and input constraints, which has led to different approaches being used to address this issue [79]. For instance, in an autonomous vehicle application, a probabilistic MPC-based safe learning filter was implemented for RL-based control [79]. The control of autonomous vehicles on a narrow road was based on an MPC-based filter and RL-based controller. Figure 2.6 shows the block diagram of the MPC-based safe filter used to control double-lane change on the narrow road. The RL-generated steering angle and acceleration, u_k^{RL}, were modified to safe steering and acceleration, uk. The authors in [79, 80] introduced a model predictive safety filter into an RL controller for nonlinear systems. The concept in the study is that the RL controller provides a control input request, which is processed by the model predictive safety filter and then applied to the real system. If the control input request from the RL controller is deemed unsafe, the model predictive safety filter will modify the control input request to ensure safety. With this model predictive safety filter, a constrained dynamical system can be turned into an unconstrained safe system, to which any RL algorithm can be applied. The method and its design procedure were illustrated by enhancing a simple RL algorithm with safety certificates using a numerical car simulation.

Based on the literature review, existing learning MPC research often focuses on using machine learning in various areas within the MPC framework, such as learning prediction models, control laws, or optimizing specific control tasks. There is a potential gap regarding the development of hybrid methods that effectively combine traditional model-based approaches with data-driven methods to leverage the strengths of the two methods in a unified framework.

2.4 Summary

This chapter presents a review of recent studies on vehicle stability control systems using various Model Predictive Control (MPC) methods. Adaptive MPC has been used in vehicle control, where the control system dynamically adjusts its parameters to accommodate changes in the system or operating conditions. It involves updating model parameters, control gains, or constraints based on real-time data and feedback, allowing the controller to adapt to variations in the vehicle's dynamics, road conditions, or driver behavior. One area where there is a gap for further study in adaptive MPC is the adaptation of the prediction horizon. The prediction horizon in MPC represents the length of the future time steps over which the controller predicts the system's behavior to optimize control actions. Traditionally, the prediction horizon is set to a fixed value, but in dynamic environments or when the system's behavior changes significantly, an adaptive prediction horizon may provide significant advantages. While traditional MPC has been widely employed in HVC, it faces challenges in modeling complex systems with uncertainties due to limited learning capabilities. Learning MPC, which integrates machine learning techniques with MPC, is gaining attention as it can further improve control performance by learning from data. Existing learning MPC research often focuses on using machine learning in various areas within the MPC framework. There is few research regarding the development of hybrid methods that effectively combine traditional model-based approaches with data-driven methods to leverage the strengths of the two methods in a unified framework.

Chapter 3
Background

In this chapter, the background for the proposed hybrid learning MPC approach is being presented. The chapter has been structured as follows: Firstly, MPC theory is introduced. Then, a general learning MPC is presented. Nonparametric machine learning and kernel regression techniques, as well as Gaussian process regression, are formulated.

3.1 Model Predictive Control Theory

Feedback controllers (also called closed-loop controllers) compare a reference with a measured variable determining a suitable value for the manipulated variable on the basis of the resulting deviation, shown as Figure 3.1. The PID controller is renowned as the most widely recognized feedback controller in industrial applications due to its exceptional significance and extensive utilization [81]. However, finding a suitable parametrization, particularly for nonlinear or time-variant systems, can often be challenging, despite the existence of various setup rules [82].

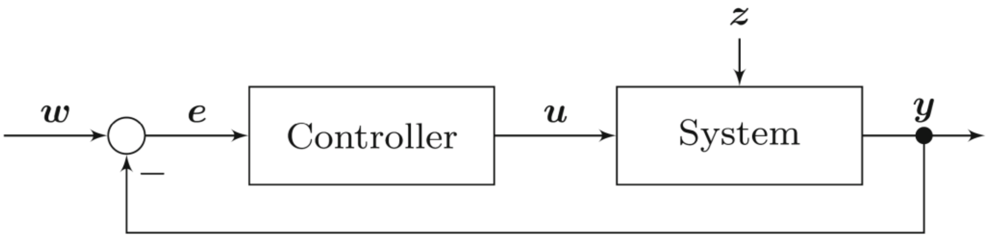


Figure 3.1: Block diagram of a classical feedback control loop (e.g., PID control) [83]

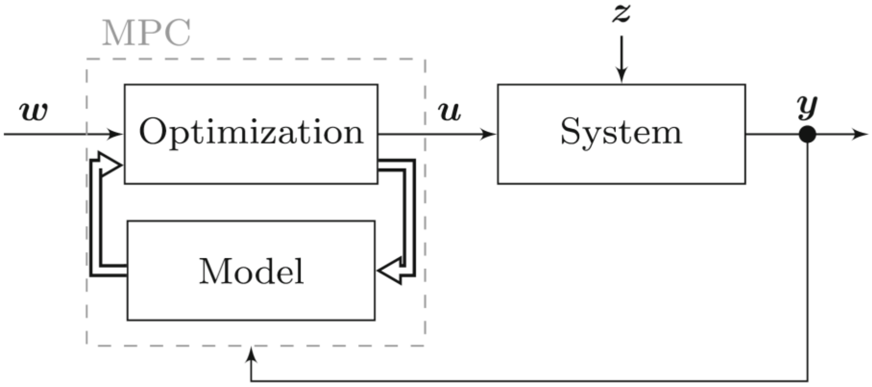


Figure 3.2: Simplified block diagram of a MPC-based control loop [83]

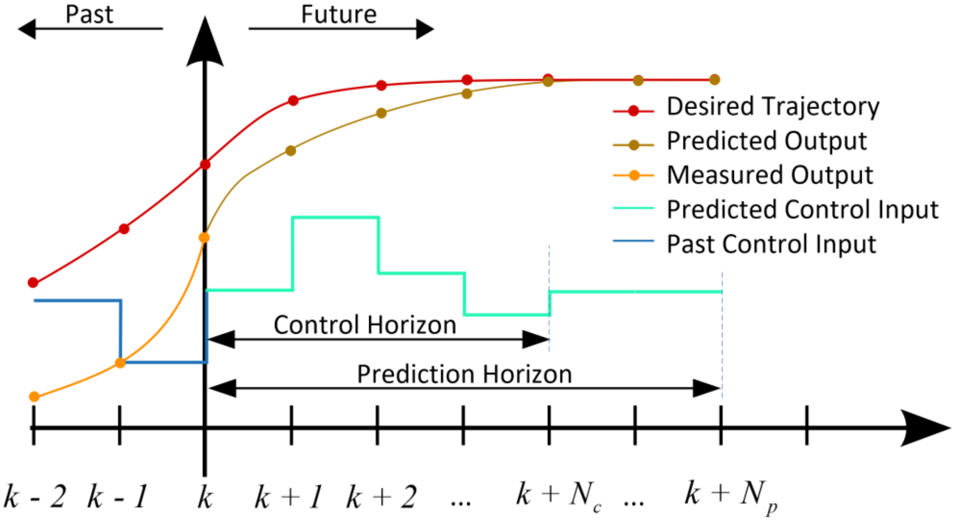


Figure 3.3: Illustration of model predictive control

MPC was introduced in the 1980s by several researchers [84, 85]. It operates by repeatedly optimizing a mathematical system model in real-time [86]. Using this model, MPC predicts the future behavior of the system and incorporates it into the optimization process to determine the optimal trajectory of the manipulated variable, as shown in Figure 3.2. While MPC offers an intuitive parameterization by adjusting a process model, it requires higher computational effort compared to classical controllers. The method's ability to anticipate behavior and consider hard constraints adds significant value in controlling real systems. As computational power continues to advance and complex process models become more accessible for various systems, MPC now enables the control of previously unimaginable systems.

The philosophy of MPC can be described using Figure 3.3. At each time step, the current system state variables are either measured or estimated. Based on these state variables, the prediction model generates the predicted output behaviors of the system (the brown line) within the time window, known as the prediction horizon. By solving an optimization problem, the MPC controller generates the optimal sequence of control inputs (the green line) within the time window, referred to as the control horizon. This optimal sequence of control inputs is typically a balanced solution to a predefined cost function, considering factors such as the tracking error between the desired trajectory (the red line) and the predicted output, the magnitude of control effort, and the smoothness of control inputs. Only the first step of the optimal control input sequence is implemented and applied to the system. Then, at the next sample time step, the system state variables are obtained again, and a new optimal sequence of control inputs is determined. This process is repeated at each time step to achieve continuous control.

In order to investigate the MPC problem, a general formulation of a linear time varying dynamic system is considered:

$$\mathbf{x} = A\mathbf{x} + B\mathbf{u} \tag{3.1}$$

where $\mathbf{x}$ and $\mathbf{u}$ are the system state and the control input of the system, respectively; A and B are the state matrix and the control input matrix, respectively. Discretization of the continuous system model at sampling time t will result in:

$$\mathbf{x}_{t+1} = \bar{A}\mathbf{x}_t + \bar{B}\mathbf{u}_t \tag{3.2}$$

where $\bar{A}$ and $\bar{B}$ are the discretized state matrix and the discretized control input matrix at sampling time t, respectively. Then, the following quadratic cost function over a time horizon N_p is defined as:

$$\begin{aligned}
& J = \min \sum_{k=1}^{N_p} \left\| \mathbf{x}(t+k|t) - \mathbf{x}_{des}(t+k|t) \right\|_L^2 + \sum_{k=0}^{N_c-1} \left\| \mathbf{u}(t+k|t) \right\|_R^2 \\
& \text{s.t. } \mathbf{x}_{k+1|t} = \bar{A}\mathbf{x}_{k|t} + \bar{B}\mathbf{u}_{k|t} \\
& \qquad \mathbf{x}_{k|t} \in \mathbf{X} \; k = t+1, \cdots, t+N_p \\
& \qquad \mathbf{u}_{k|t} \in \mathbf{U} \; k = t+1, \cdots, t+N_c-1
\end{aligned} \tag{3.3}$$

where L and R are the weight matrix of the state tracking error and the control effort, respectively; $\mathbf{x}_{des}(t+k|t)$ is the desired state trajectory, and $\mathbf{x}(t+k|t)$ is the predicted state trajectory by applying the control sequence $\mathbf{u}_{k|t}$ to the system; $\mathbf{X}$ and $\mathbf{U}$ denote the state and control input constraints, respectively.

3.2 A General Learning MPC Structure

Learning MPC leverages machine learning techniques to learn the system dynamics directly from data, allowing it to handle model uncertainty more effectively. Learning MPC was initially introduced in [21] and later used in mainly robotics applications in [7, 71]. To describe learning MPC mathematically, one must define three discrete-time models of the system dynamics. The first one is the true system dynamics, defined as

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t + g\left(\mathbf{x}_t, \mathbf{u}_t\right) \tag{3.4}$$

where t denotes time, $\mathbf{x}$ and $\mathbf{u}$ are the system state and the control input of the true system, respectively; A and B are the state matrix and the control input matrix, respectively; g represents unmodelled dynamics or disturbance.

The second model is the nominal system dynamics model, defined as

$$\bar{\mathbf{x}}_{t+1} = A\bar{\mathbf{x}}_t + B\bar{\mathbf{u}}_t \tag{3.5}$$

where $\bar{\mathbf{x}}_t$ and $\bar{\mathbf{u}}_t$ are the system state and the control input of the nominal model, respectively.

The third model is the learned system dynamics model with the formulation as

$$\tilde{\mathbf{x}}_{t+1} = A\tilde{\mathbf{x}}_t + B\hat{\mathbf{u}}_t + \mathrm{O}_t\left(\tilde{\mathbf{x}}_t, \hat{\mathbf{u}}_t\right) \tag{3.6}$$

where $\tilde{\mathbf{x}}_t$ and $\hat{\mathbf{u}}_t$ are the system state and the control input of the learned model, respectively; O_t is an oracle that provides corrections to the nominal model by learning the unmodelled dynamics g online.

With these three models, the general structure of learning MPC is defined as [8]:

$$\begin{aligned}
&V_t\left(\mathbf{x}_t\right) = \min_{c,\theta} J_t\left(\tilde{\mathbf{x}}_t, ..., \tilde{\mathbf{x}}_{t+N}, \hat{\mathbf{u}}_t, ..., \hat{\mathbf{u}}_{t+N-1}\right) \\
&s.t. \\
&\begin{cases}
\tilde{\mathbf{x}}_t = \mathbf{x}_t,\ \bar{\mathbf{x}}_t = \mathbf{x}_t \\
\tilde{\mathbf{x}}_{t+i+1} = A\tilde{\mathbf{x}}_{t+i} + B\hat{\mathbf{u}}_{t+i} + \mathrm{O}_t\left(\tilde{\mathbf{x}}_{t+i}, \hat{\mathbf{u}}_{t+i}\right) \\
\bar{\mathbf{x}}_{t+i+1} = A\bar{\mathbf{x}}_{t+i} + B\hat{\mathbf{u}}_{t+i} \\
\hat{\mathbf{u}}_{t+i} = K\bar{\mathbf{x}}_{t+i} + c_{t+i} \\
\bar{\mathbf{x}}_{t+i+1} \in \mathrm{X} \,!\ \mathrm{R}_{i+1},\ \hat{\mathbf{u}}_{t+i} \in \mathrm{U} \,!\ K\mathrm{R}_i \\
\bar{\mathbf{x}}_{t+N} \in \Omega \,!\ \mathrm{R}_N
\end{cases}
\end{aligned} \tag{3.7}$$

where Ω is the constructed invariant set, in which any trajectory of the system with initial condition chosen from this set and with control remains within the set for any sequence of bounded disturbance while satisfying constraints on the state and input, and K is the feedback gain used to compute Ω.

As can be seen from the above mathematical structure, learning MPC maintains two models: a nominal model $\bar{\mathbf{x}}_{t+1} = A\bar{\mathbf{x}}_t + B\bar{\mathbf{u}}_t$ with bounds on system states and control inputs and a learned model

$\tilde{\mathbf{x}}_{t+1} = A\tilde{\mathbf{x}}_t + B\hat{\mathbf{u}}_t + \mathrm{O}_t\left(\tilde{\mathbf{x}}_t, \hat{\mathbf{u}}_t\right)$ with an oracle O_t that can be updated by statistical methods. Its performance is optimized by choosing control inputs $\hat{\mathbf{u}}_{t+i}$ that minimize a cost function subject to the learned model (rather than the nominal model), while its safety and robustness are ensured by checking whether the same control inputs $\hat{\mathbf{u}}_{t+i}$ maintain the nominal model stable when it is subject to uncertainties. This innovative decoupled architecture handles the performance and safety criteria separately using theories from reachability analysis [87] and from robust MPC [88]. One of the other advantages associated with this decoupled structure is that this learning MPC is robust to mis-learning, which means that even if the oracle O_t is poorly designed, the robustness is still provably maintained. It is argued in [8] that the control performance of learning MPC is typically superior to that of linear MPC, and as good as that of nonlinear MPC. However, unlike nonlinear MPC that usually requires the expertise of designers to conduct accurate modeling of the nonlinearities of the system and imposes a huge computational burden on real-time controllers, learning MPC can use a simple system model and leave the work of capturing the modeling errors and uncertainties to statistical tools. In general, learning MPC can (a) handle state and input constraints, (b) optimize system performance for a cost function, (c) take advantage of statistical tools to learn model uncertainties, and (d) be provably convergent. These properties of learning MPC are proved both theoretically [8] and experimentally [7, 70].

3.3 Gaussian Process Learning Model

3.3.1 Nonparametric Machine Learning

A machine learning algorithm can be classified as either parametric or nonparametric. Parametric algorithms have a fixed number of parameters. A parametric machine learning algorithm is usually computationally faster, but it makes stronger assumptions about the data distribution; the algorithm may work well if the assumptions turn out to be correct, but it may perform poorly if the assumptions are wrong. In contrast, nonparametric algorithms use a flexible number of parameters. Nonparametric machine learning algorithms are often computationally intractable for large datasets, but they usually achieve better performance due to fewer assumptions about the data [89].

As a classical machine learning technique, regression is used to find a function representing a set of data points as closely as possible. Generally, regression methods can also be divided into two categories, parametric regression, and nonparametric regression. In parametric regression models, the mapping function $\mathrm{y}\left(\mathbf{x}, \mathbf{w}\right)$ from input vector $\mathbf{x}$ to output y is governed by a weight vector $\mathbf{w}$ obtained using a set of training data during the learning phase. The training data is then discarded, and during the prediction phase, the predictions for new inputs are purely based on the learned weight vector $\mathbf{w}$ [90]. For nonparametric regression methods, the training data are stored first and then used during the prediction phase.

Nonparametric regression methods are memory-based methods that involve storing the entire training set to make predictions for future data points, typically requiring a metric to be defined that measures the similarity of any two vectors in the input space [90]. A kernel is a weighting function that assigns weights to the neighboring data points based on their distances to the query point where its output needs to be predicted. Kernel functions are widely used in machine learning because they allow distance-based similarity measures that go far beyond the standard Euclidean distance.

A simple and easy-to-implement nonparametric regression method is the so-called K-Nearest-Neighbor (KNN) algorithm. The KNN method uses those observations in the training dataset closest to the query point in the input space to make the prediction for the query point. In KNN, an estimation of the regression function is calculated as

$$\hat{f}\left(x_0\right) = \mathrm{Ave}\left(y_i \middle| x_i \in N_K\left(x_0\right)\right) \tag{3.8}$$

where $N_k\left(x_0\right)$ is the set of K points nearest to the query point x_0 and Ave denotes the average function.

Instead of giving all the neighborhood points an equal weight in the KNN method, the Nadaraya-Watson regression (NWR) method, also known as kernel regression, assigns weights that die off smoothly with distance from the query point [91]. In NWR, the kernel-weighted average is calculated as

$$\hat{f}\left(x_0\right) = \frac{\sum_{i=1}^{N} k\left(x_0, x_i\right) y_i}{\sum_{i=1}^{N} k\left(x_0, x_i\right)} \tag{3.9}$$

where k is the kernel function that will be discussed in the next section.

The performances using KNN and NWR are illustrated in Figure 3.4. In this figure, 100 pairs x_i, y_i are generated at random from the blue curve with Gaussian errors. The red point represents the fitted constant $\hat{f}(x_0)$, and the red circles indicate those observations contributing to the fit at a given target point x_0. The solid yellow region shows the weights assigned to observations. In the left panel, the green curve is the result of applying a 30-nearest-neighbor running-mean smoother to the KNN predictions. This smoothing technique helps visualize the overall trend in the fitted values as the target point varies along the x-axis. However, it can be observed that the green curve using KNN at different values appears bumpy. This is due to the discrete nature of KNN, where the predicted values are obtained by averaging the target variable values of the k-nearest neighbors, resulting in local fluctuations. In contrast, the right panel shows the performance using NWR. As the target point is moved from left to right, points enter the neighborhood initially with weight zero. Then, their contribution gradually increases as the target point gets closer to each data point within its neighborhood. The green curve in the right panel represents the kernel-weighted average obtained using NWR. It can be seen that the fitted function using NWR is continuous and quite smooth. This is a result of the continuous weighting scheme in NWR, where weights of nearby observations are assigned based on a kernel function, allowing for a smooth transition between neighboring target points.

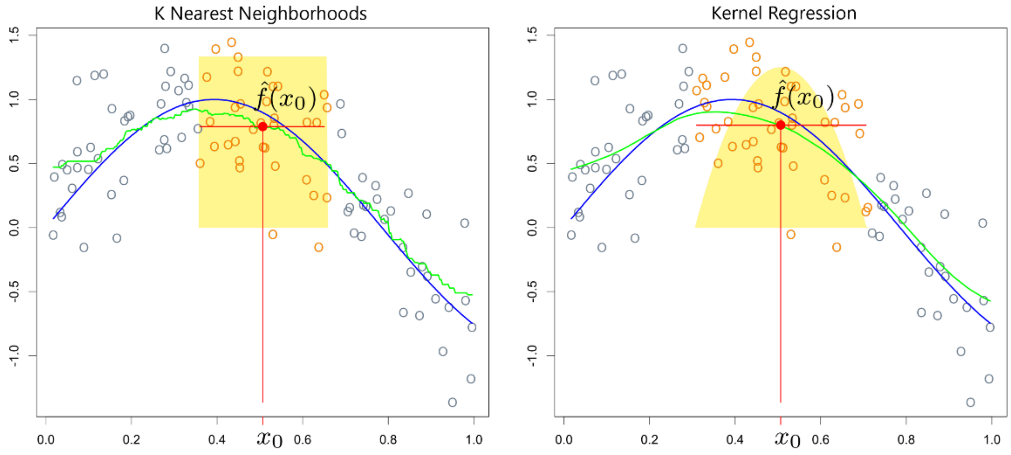


Figure 3.4: Illustration of kernel regression [91]

3.3.2 Kernel Function

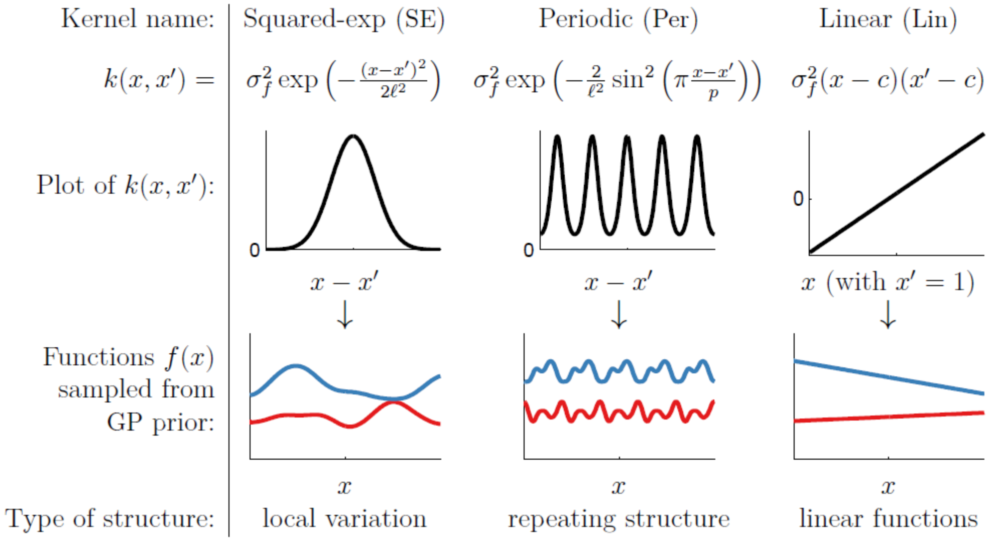


Figure 3.5: Examples of basic kernel functions [92].

Kernel functions are fundamental components of various machine learning algorithms. These functions play a crucial role in capturing the underlying relationships and patterns in data by defining the similarity or distance measure between data points. Kernel functions provide a flexible and efficient way to transform the input space into a higher-dimensional feature space, where complex relationships can be more easily modeled. They allow algorithms to implicitly operate in this feature space without explicitly computing the transformed feature vectors, thereby avoiding the computational burden of explicit feature expansion. The primary purpose of a kernel function is to measure the similarity or dissimilarity between pairs of data points. It takes two inputs and returns a similarity measure or a measure of how much the points "look alike." The choice of kernel function depends on the characteristics of the data and the learning task at hand. Basic kernel functions include squared exponential kernel, and periodic kernel, and linear kernel, shown as Figure 3.5.

Squared Exponential (SE) Kernel: The SE kernel, also known as the Gaussian kernel or Radial Basis Function (RBF) kernel, is one of the most commonly used kernel functions in machine learning algorithms. It is known for its ability to capture smooth and continuous relationships between variables. The SE kernel is a popular choice for capturing non-linear relationships. It assigns a similarity measure based on the distance between data points, with closer points having higher similarity. The kernel includes a bandwidth parameter that controls the kernel's smoothness and flexibility.

Periodic Kernel: The Periodic kernel is a specialized kernel function commonly used to model data that exhibits periodic patterns or periodicity in its underlying structure. It is particularly useful when dealing with time series or cyclic data where the periodicity is an essential characteristic.

Linear Kernel: The linear kernel computes the dot product between two vectors, capturing linear relationships between features. It is widely used when the data is believed to be linearly separable.

3.3.3 Gaussian Process Regression

Gaussian process regression (GPR) is a powerful nonparametric regression in machine learning and has been widely studied in many different fields [71, 72, 93]. A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution [93]. A Gaussian process is a generalization of the Gaussian distribution, whereas a Gaussian distribution describes the probability distribution of random variables that are scalars or vectors, a Gaussian process governs the probability distribution over possible functions that fit a set of data points, which can be loosely thought of as a long vector, and each entry in the vector specifying the function value at a particular input [93]. Figure 3.6 is a visual representation of a Gaussian process model of a one-dimensional test function $f(\theta)$ over the input θ, in which the dashed line represents the target function, the red dots are exsiting data with observation noise, the blue line shows the mean prediction of the Gaussian process regression with the shaded 2σ credible region indicating the corresponding uncertainty [94].

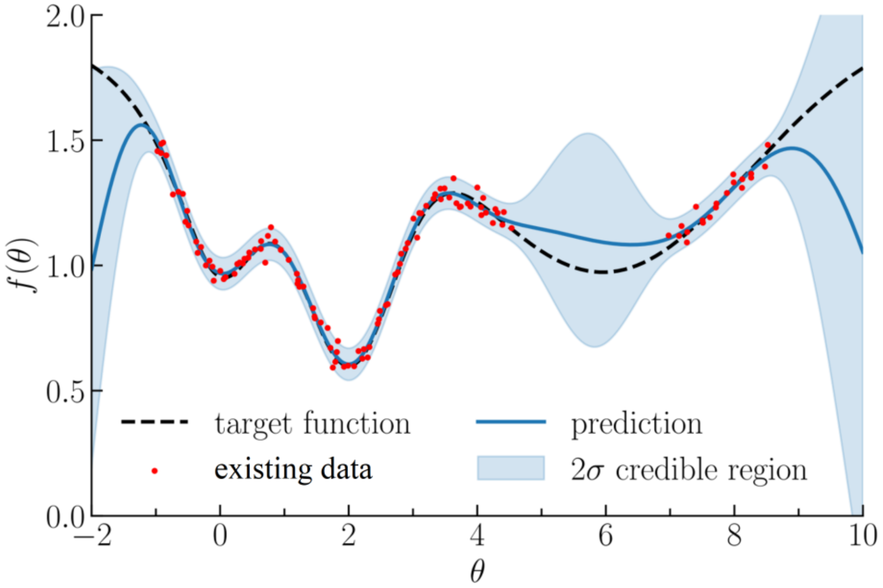


Figure 3.6: Illustration of Gaussian process regression in one dimension [94]

Compared with other nonparametric regression methods, GPR has several benefits, including: (1) working well on relatively small datasets, and (2) being able to explicitly deal with measurement noise, and (3) being mathematically explainable, as well as (4) having the ability to provide uncertainty measurements on the predictions, which makes it suitable to be used in the learning MPC for vehicle control systems.

Now it is time to formulate the standard Gaussian process model for regression. The regression function modeled by a Gaussian process is given as [95]

$$p\left(\mathbf{f}\,|\mathbf{X}\right) \sim \mathcal{N}\left(\boldsymbol{\mu},\mathbf{K}\right) \tag{3.10}$$

where $\mathbf{X}=\left[\mathbf{x}_1,...,\mathbf{x}_n\right]$ are the n observed data points, $\mathbf{f}=\left[f\left(\mathbf{x}_1\right),...,f\left(\mathbf{x}_n\right)\right]$ are the function outputs of these data points, $\boldsymbol{\mu}=\left[\mu\left(\mathbf{x}_1\right),...,\mu\left(\mathbf{x}_n\right)\right]$ represents the mean function, and $\mathbf{K}$ represents the covariance function.

With some observed data points, the predicted outputs at new test points $\mathbf{X}_*$ are denoted as $\mathbf{f}_*$. Then, the joint distribution is also Gaussian process and expressed as

$$\begin{bmatrix}\mathbf{f}\\ \mathbf{f}_*\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}\\ \boldsymbol{\mu}_*\end{bmatrix},\begin{bmatrix}\mathbf{K} & \mathbf{K}_*\\ \mathbf{K}_*^{\mathrm{T}} & \mathbf{K}_{**}\end{bmatrix}\right) \tag{3.11}$$

where $\mathbf{K}=k\left(\mathbf{X},\mathbf{X}\right)$, $\mathbf{K}_*=k\left(\mathbf{X},\mathbf{X}_*\right)$, and $\mathbf{K}_{**}=k\left(\mathbf{X}_*,\mathbf{X}_*\right)$ are covariance matrixes calculated using a kernel function $k\left(x,x'\right)$.

From the joint Gaussian distribution, the conditional distribution of $\mathbf{f}_*$ can be obtained using the marginal and conditional distributions theorem. The result is expressed as

$$\mathbf{f}_*\left|\mathbf{f},\mathbf{X},\mathbf{X}_*\right. \sim \mathcal{N}\left(\mathbf{K}_*^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{f},\mathbf{K}_{**}-\mathbf{K}_*^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{K}_*\right) \tag{3.12}$$

In practice, the measured function values are noisy, which can be defined as

$$y=f\left(\mathbf{x}\right)+\varepsilon \tag{3.13}$$

where $\varepsilon \sim \mathcal{N}\left(0,\sigma_n^2\right)$ is Gaussian noise with zero mean and variance σ_n^2.

Now, for the test point $\mathbf{X}_*$, the prediction for its output is given as

$$\mathbf{y}_* \left| \mathbf{y}, \mathbf{X}, \mathbf{X}_* \right. \sim \mathrm{N}\left(\text{mean}, \text{cov}\right) \tag{3.14}$$

where the mean function and the covariance function are expressed as

$$\text{mean} = \mathbf{K}_*^{\mathrm{T}} \left(\mathbf{K} + \sigma_n^2 I\right)^{-1} \mathbf{y} \tag{3.15}$$

$$\text{cov} = \mathbf{K}_{**} - \mathbf{K}_*^{\mathrm{T}} \left(\mathbf{K} + \sigma_n^2 I\right)^{-1} \mathbf{K}_* \tag{3.16}$$

The uncertainty for the predicted mean value is commonly expressed using the 2σ credible region, also named the 95% credible interval (CI) that is calculated as

$$95\%\ \text{CI} = 1.96 \times \sqrt{\text{cov}} \tag{3.17}$$

To calculate the covariance matrixes $\mathbf{K}$, $\mathbf{K}_*$, and $\mathbf{K}_{**}$, a kernel function $k\left(x, x'\right)$ is to be used. The most widely used kernel function is the squared exponential (SE) kernel function, also called the Gaussian kernel, which is defined as

$$k\left(x, x'\right) = \sigma_f{}^2 \exp\left(-\frac{1}{2l}\left(x - x'\right)^T \left(x - x'\right)\right) \tag{3.18}$$

where $\sigma_f{}^2$ and l are hyperparameters. The vertical scale $\sigma_f{}^2$ describes how vertically the function can span and determines the average distance of the function away from its mean. The horizontal scale l determines the length of the 'wiggles' in the function, and a larger l generates a smoother function while a smaller l gave a wigglier function.

The SE kernel is the default kernel for the Gaussian process due to its nice properties. First, it is universal and can be integrated into most functions. Secondly, compared with other kernels, it has only two hyperparameters $\sigma_f{}^2$ and l that need to be optimized.

3.3.4 Hyperparameters for GPR

Hyperparameters are all those parameters of a machine learning model which are not updated during the learning and are used to configure either the model or the algorithm used to lower the cost function. The hyperparameters $\sigma_f{}^2$ and l in the SE kernel function need to be finely tuned in order

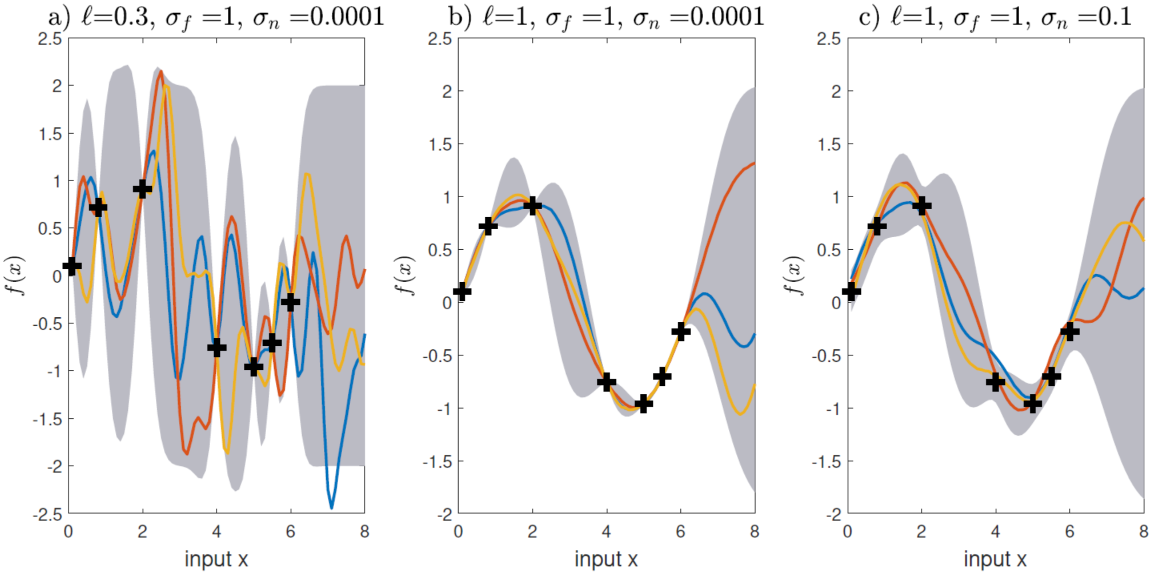


Figure 3.7: GPR predictions with different sets of hyperparameters [96].

to use the GPR algorithm effectively. As seen in Figure 3.7, the prediction quality of GPR is highly dependent on the hyperparameters. Generally, the typical algorithms used for hyperparameters tuning in a machine learning model can also be used here for GPR.

The simplest way to select hyperparameters is by manual search, in other words, trial and error. In this method, users try different hyperparameters values based on guessing or domain knowledge and repeat this process until they obtain a satisfactory result [97]. Manually tuning hyperparameters typically needs sufficient time to get good results, but it will help beginners better understand the effect of hyperparameters on the results. Thus, this method will be used in the early phase of this research.

Another way to tune hyperparameters is to use search-based methods. For the search-based methods, a search space must be defined beforehand, setting bounds for all the hyperparameters, and adding some prior knowledge on them. In the grid search algorithm, the search space of each hyperparameter is discretized, and the total search space is discretized as the Cartesian products of them. Then, the algorithm launches learning for each of the hyperparameter configurations and chooses the best at the end. Random search is a variation of the grid search algorithm, which randomly samples the search space instead of discretizing it with a Cartesian grid. The number of trials must be set to end the searching process. Compared with grid search, the random search usually results in more precisely the optima of hyperparameters. Both grid search and random search are easy to implement; however, there is no guarantee of finding a local minimum unless the search space is thoroughly sampled.

The method of Bayesian optimization can also be used to optimize hyperparameters. Fundamentally, Bayesian optimization is a sequential model-based approach to solving the problem of finding a global maximizer (or minimizer) of an unknown objective function [98]. Bayesian optimization builds a probabilistic proxy model (e.g., Gaussian process) for the objective function, using outcomes of past experiments as training data [99]. The entire optimization approach is guided by an acquisition function, which defines the following points (i.e., set of hyperparameters) to be evaluated. Bayesian optimization is well suited when the function evaluations are expensive, making grid or exhaustive search-based methods impractical [100].

In this study, the hyperparameters are selected based on the maximization of the marginal likelihood as given by [93]

$$p\left(y|X,\theta\right)=\int p\left(y|X,\theta\right)p\left(\mathbf{f}\,|X,\theta\right)d\mathbf{f} \tag{3.19}$$

where θ denotes the hyperparameter vector.

This can be done by maximizing the log marginal likelihood. The hyperparameter vector $\theta=[\ell_1,\cdots,\ell_d,\sigma^2]$ can then be obtained as

$$\theta_{\text{opt}}=\arg\min_{\theta}\left(-\log p\left(y|X,\theta\right)\right) \tag{3.20}$$

3.3.5 Feature Selection for GPR

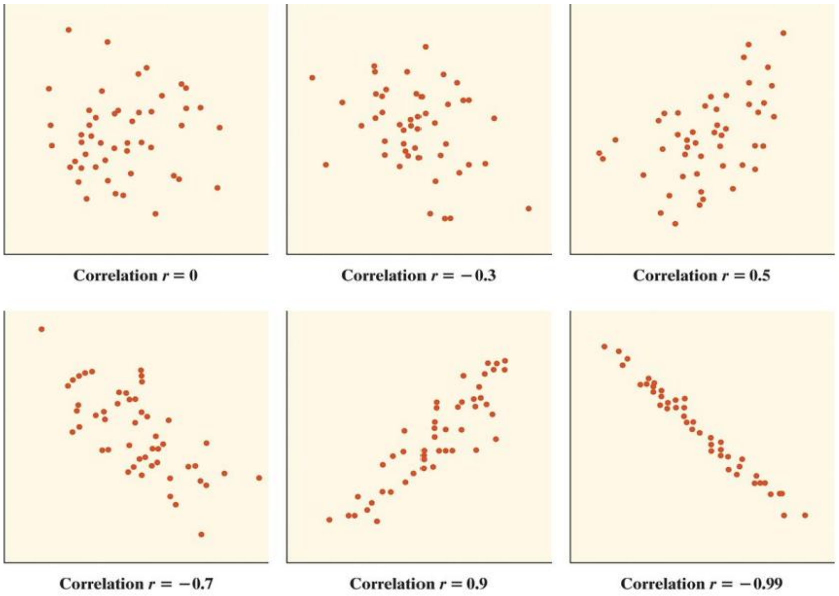
Feature selection in GPR refers to the process of identifying a subset of input features that are most relevant to the prediction task. Feature selection is an important step in GPR to identify the most relevant features that contribute to the prediction task. By selecting informative features, model interpretability can be improved, computational complexity can be reduced, and prediction accuracy can be potentially enhanced. To select features in GPR, the pairwise correlations between the features and the target variable should be calculated. Features with high absolute correlation values are likely to be more relevant.

In this study, the Pearson correlation coefficient (PCC) method is used to identify important variables for predictions. PCC was initially introduced by Karl Pearson in 1895 as a measure of the correlation between two sets of data in statistics [101]. By definition, PCC is the covariance of the two variables divided by the product of their standard deviations, described as

$$P_{xy}=\frac{\sum_i \left(x_i-\overline{x}\right)\left(y_i-\overline{y}\right)}{\sqrt{\sum_i \left(x_i-\overline{x}\right)^2}\sqrt{\sum_i \left(y_i-\overline{y}\right)^2}} \tag{3.21}$$

where $\overline{x}$ and $\overline{y}$ are the mean values of the variables x and y, respectively.

Thus, PCC is essentially a normalized measurement of the covariance, such that the result always has a value between -1 and 1. PCC equals a value of 1 if the two variables are perfectly positively related, a value of -1 if they are perfectly negatively related, and a value of 0 if they do not have any relation. Figure 3.8 nicely illustrates PCC through scatterplots. A PCC matrix is a table showing correlation coefficients for multiple variables. Each cell in the table shows the correlation between the corresponding two variables. A PCC matrix is symmetrical, with the same correlation shown above the main diagonal and those below the main diagonal. It's important to note that PCC only captures linear relationships between variables. If the relationship between the features and the target variable is non-linear, PCC may not be sufficient for feature selection. In such cases, it might be necessary to consider other techniques such as mutual information, nonlinear correlation measures, or model-based feature selection methods.



Correlation $r = 0$

Correlation $r = -0.3$

Correlation $r = 0.5$

Correlation $r = -0.7$

Correlation $r = 0.9$

Correlation $r = -0.99$

Figure 3.8: Pearson correlations visualized as scatterplots [102]

3.4 Summary

This chapter provides an overview of the background and foundational concepts related to GPR-based hybrid learning Model Predictive Control (MPC). First, the fundamental theory of MPC is presented. MPC is a control technique that utilizes an optimization framework to solve a dynamic optimization problem repeatedly over a finite time horizon. The primary objective is to determine control actions that optimize a given performance criterion while satisfying system dynamics and constraints. Then, this chapter introduced the concept of learning MPC, which combines the principles of MPC with machine learning techniques. Unlike a traditional MPC that typically uses one physics-based prediction model, learning MPC maintains two models, a physics-based nominal system dynamics model with bounds on system states and control inputs, as well as a data-based learned system dynamics model with an oracle that can be updated by statistical methods. With the decoupled structure of learning MPC, performance and robust safety are guaranteed. These properties of learning MPC are important for developing an evolving vehicle stability control system. Learning MPC aims to enhance control performance by leveraging historical data and adaptive learning algorithms. By incorporating machine learning models, it becomes possible to capture complex system dynamics and improve predictive capabilities. To enable flexible and adaptive modeling in learning MPC, nonparametric machine learning techniques are employed. Nonparametric methods do not assume a fixed functional form but instead estimate the underlying relationship between input and output variables from data. This chapter provides an overview of nonparametric approaches and their advantages in learning MPC applications. Compared with parametric machine learning methods, one of the advantages of nonparametric machine learning techniques lies in its online learning capability, which is attractive for vehicle control systems that usually face uncertainties during practical operations. As a powerful nonparametric regression technique, GPR is explained through the formulation, where a prior distribution over functions is defined by a mean function and a covariance (kernel) function. GPR offers a flexible framework for modeling complex relationships and estimating uncertainties. Besides, GPR works well on small datasets, which is important for real-time applications and can provide uncertainty measurements on its predictions. These unique characteristics make GPR an ideal choice to be used in learning MPC for vehicle stability control systems. The choice of kernel function significantly influences the behavior and assumptions made by GPR models. This chapter also discusses various kernel functions commonly used in GPR, such as the squared exponential kernel and the linear kernel. Each kernel captures different properties of the underlying data, enabling the modeling of diverse system dynamics. Optimizing the hyperparameters of GPR models is crucial for achieving optimal performance. This chapter discusses techniques for hyperparameter optimization, which involve tuning parameters such as the kernel hyperparameters and noise variance. Approaches like trial and error, grid search, and Bayesian optimization are introduced to find the best hyperparameter configuration. Finally, a feature selection method for GPR was presented. The feature selection method plays a vital role in GPR-based learning MPC by identifying the most relevant input variables. This chapter explores the use of Pearson Correlation Coefficient (PCC) as the feature selection method. PCC measures the linear correlation between features and the target variable, assisting in the identification of influential features.

Chapter 4
Proposed Hybrid Learning MPC for Holistic Vehicle Control

In this chapter, a hybrid learning MPC controller for holistic vehicle control (HVC) is developed. This chapter is structured as follows. First, the control architecture of the hybrid learning MPC for HVC is introduced. Then, the physics-based nominal model and the data-driven learned model are designed. Next, the data collection and management method for the learned model is presented, and an authentication strategy is proposed to evaluate the models. Finally, the cost function and constraints are defined in the hybrid learning MPC for HVC.

4.1 Structure of Hybrid Learning MPC for HVC

The proposed hybrid learning MPC structure combines traditional vehicle dynamic models with data-driven approaches to enhance control performance and adaptability. As Figure 4.1 shows, in this general hybrid learning MPC structure, the control strategy incorporates two distinct control paths to optimize the control actions. The first path is the physics-based control path, which relies on a nominal model of the system derived from first principles or physics-based equations. This model represents the fundamental dynamics and characteristics of the vehicle system, providing a reliable baseline for control decisions. The second path is the data-based control path, where machine learning techniques, such as GPR, are employed to learn from historical data and real-time measurements. This data-driven learned model captures the system's behavior from observed data, enabling the controller to adapt and improve its predictions based on the actual performance of the vehicle. The hybrid nature of this approach allows the controller to combine the strengths of both paths to achieve enhanced control performance. The physics-based control path provides stability and robustness through its knowledge of system dynamics, while the data-based control path offers adaptability and responsiveness to changes in the environment or system behavior. By continuously updating and refining the learned model with new data, the hybrid learning MPC can continuously improve its control decisions, making it well-suited for handling complex and uncertain scenarios encountered in holistic vehicle control applications. The proposed hybrid learning MPC controller consists of the following key components:

Vehicle Dynamic Model: A traditional physics-based dynamic model is incorporated into the MPC framework to capture the underlying physics and dynamics of the vehicle system. This model includes equations that describe the vehicle's motion, such as longitudinal and lateral dynamics, tire characteristics, and vehicle constraints.

Data Acquisition and Management: Real-time data from various sensors, such as the wheel speed sensor and the Inertial Measurement Unit (IMU) is acquired to provide accurate and up-to-date information about the vehicle's state. Data management involves various tasks, such as data preprocessing, feature engineering, and data selection for prediction.

Data-driven Vehicle Modeling: In this component, data-driven machine learning algorithms are employed to learn the dynamics of the vehicle. These techniques utilize the collected sensor data to estimate the unknown or time-varying dynamics of the vehicle, allowing for accurate representation of the vehicle's behavior. In this study, GPR is used in this module.

Model Authentication: This module focuses on evaluating the credibility and reliability of the machine learning model using appropriate evaluation metrics. One of the key advantages of GPR is its ability to provide uncertainty estimates. This module evaluates the uncertainty estimates generated by the GPR model and ensures they align with the expected level of uncertainty in the problem domain. High uncertainty regions should correspond to driving scenarios with sparse data when a traditional physics-based model will be used in the prediction.

Optimization Control Module: The hybrid learning MPC algorithm utilizes the vehicle dynamic model, the acquired sensor data, and the learned model parameters to predict the future behavior of the vehicle and optimize control actions. This module solves an optimization problem over a finite time horizon to determine the optimal control inputs that minimize a defined cost function while satisfying constraints.

Control Actuation Module: Based on the optimized control inputs from the hybrid learning MPC algorithm, control actions are generated and translated into actuators, such as TV, DB, or AFS.

By implementing the hybrid learning MPC structure in the context of HVC, the control architecture is visualized in Figure 4.2. The hybrid learning MPC controller receives the desired yaw rate, which is computed by a Driver Command Interpreter (DCI) module. The DCI module derives the desired vehicle responses based on the driver's inputs, such as the steering angle and the drive torque, along with vehicle states obtained from onboard sensors or estimation modules. The hybrid learning MPC controller processes this information and calculates the optimal adjustments for the drive torque. These adjustments are then integrated with the driver's requests, resulting in modified drive torque

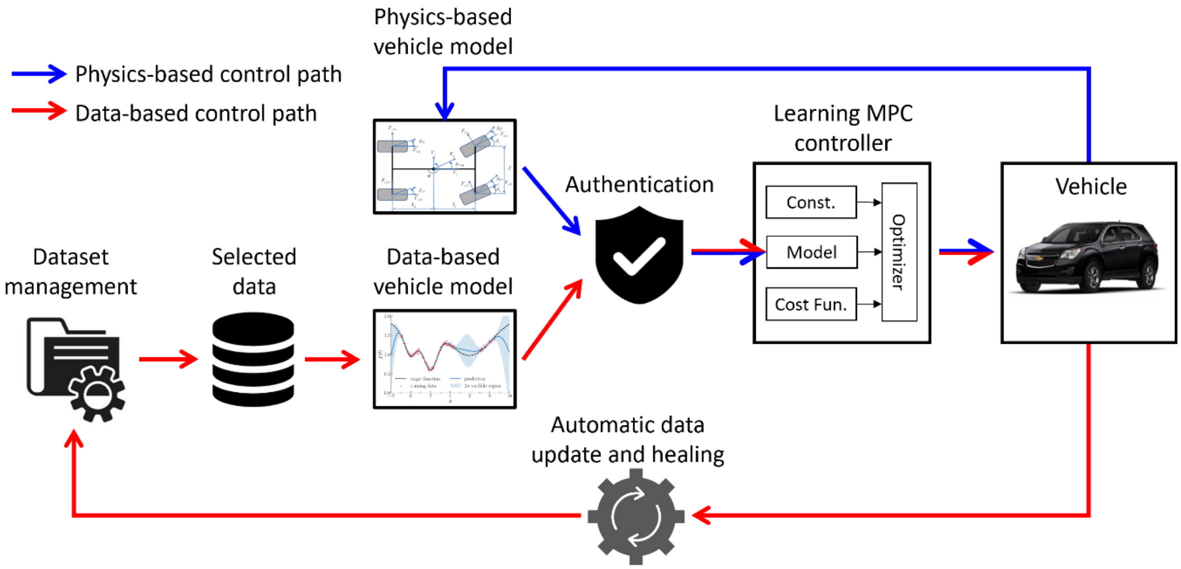


Figure 4.1: The general structure of the proposed hybrid learning MPC.

commands. Finally, adjusted drive torque commands are applied to the vehicle, leading to refined control actions that work collaboratively with the driver's inputs to achieve HVC.

In the hybrid learning MPC controller, a combination of a physics-based vehicle model and a learning-based model is utilized to forecast vehicle responses for predictive control, distinguishing it from the conventional MPC approach that solely relies on a physics-based model for all conditions. The learning-based model, unlike the physics-based model, can leverage a wide range of machine learning techniques and a vast amount of valuable data gathered from the vehicle to accurately model vehicle and tire dynamics, especially capturing nonlinear behaviors beyond the capabilities of the physics-based model. However, the performance of the learning-based vehicle model is highly contingent on the size and effective utilization of the training dataset. To address this, a data management module is incorporated, responsible for collecting, updating, and selecting data using statistical tools to enhance the learned model's performance. With two vehicle dynamics models in the hybrid learning MPC controller, a model authentication module is introduced to evaluate the accuracy of both models and select the more reliable one for use in the constrained optimization problem. The authentication process ensures that the most dependable model is employed to enhance control effectiveness. In the subsequent sections, detailed discussions on these modules will be provided.

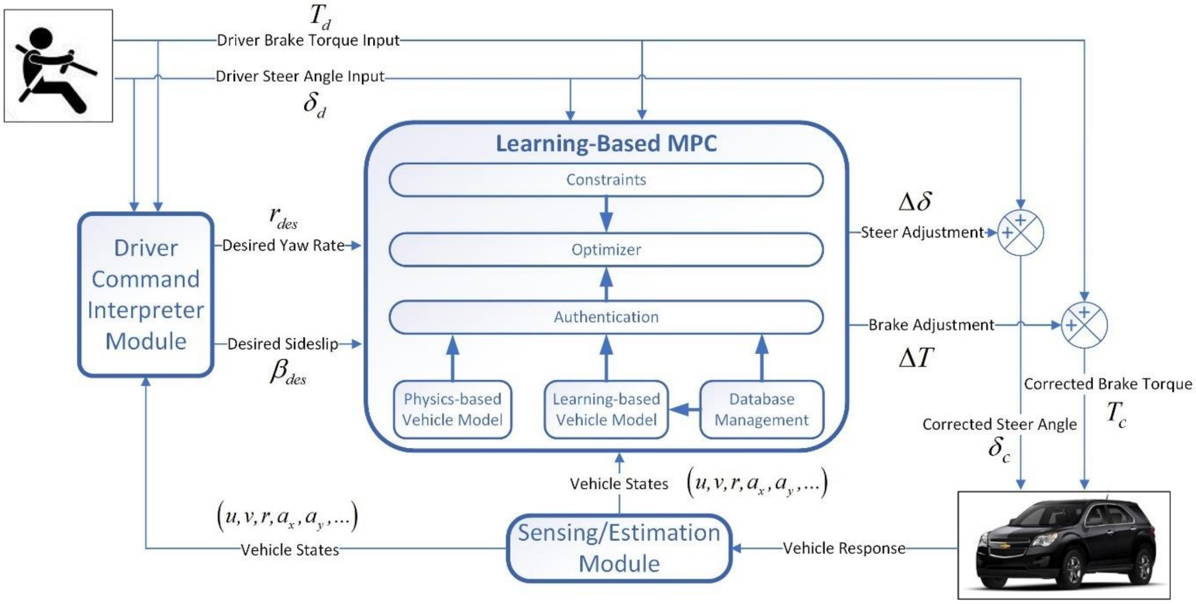


Figure 4.2: The control architecture of the hybrid learning MPC for HVC

4.2 Desired Vehicle Responses

The ideal yaw rate refers to the target rate of rotation around the vertical axis (z-axis) of a vehicle. In this study, the theoretically ideal yaw rate for the vehicle is calculated as [25],

$$r^* = \frac{u}{L + K_{us} u^2} \delta_f \tag{4.1}$$

where L is the wheelbase of the vehicle, δ_f is the front steering angle, u is the longitudinal velocity, and K_{us} denotes the understeer coefficient.

The understeer coefficient K_{us} in Equation (4.1) is obtained by

$$K_{us} = \frac{2m\left(l_r C_{\alpha r} - l_f C_{\alpha f}\right)}{L C_{\alpha f} C_{\alpha r}} \tag{4.2}$$

where m denotes the mass of the vehicle; $C_{\alpha f}$ and $C_{\alpha r}$ are the cornering stiffness for each front and rear tire, respectively; l_f and l_r are the distance to the vehicle's center of gravity (CG) for each front and rear tire, respectively.

Inserting Equations (4.2) into Equation (4.1) gives

$$r^* = \frac{u}{L + \frac{mu^2 \left(C_{\alpha r} l_r - C_{\alpha f} l_f \right)}{2C_{\alpha f} C_{\alpha r} L}} \delta_f \tag{4.3}$$

In practice, r^* cannot always be obtained due to the limitation of lateral acceleration. Therefore, the desired yaw rate is given as

$$r_{des} = \operatorname{sign}\left(r^*\right) \min\left(\left|r^*\right|, \left|\frac{a_{y\max}}{u}\right| \right) \tag{4.4}$$

where $a_{y\max}$ denotes the maximum lateral acceleration of the vehicle, which depends on road conditions.

4.3 Physics-based Vehicle Model

4.3.1 Vehicle and Tire Dynamics

In vehicle control applications, having a suitable and accurate model is crucial for achieving effective control performance while keeping computational costs manageable [103]. There are two commonly used simplified vehicle models: the kinematic bicycle model and the dynamic bicycle model. The kinematic bicycle model provides a relatively straightforward representation of vehicle behavior, making it computationally less expensive compared to the dynamic bicycle model. However, the kinematic bicycle model may not capture all the essential dynamics, particularly when dealing with high-speed cornering and large lateral forces that lead to significant side slip angles in the vehicle's tires [104]. On the other hand, the dynamic bicycle model is capable of achieving smaller prediction errors, especially under conditions where the vehicle is driven at high speeds and experiences larger tire side slip angles, which are typical scenarios in the study of vehicle stability control performance. Therefore, given the focus of this thesis on vehicle stability control, where harsh maneuvers with considerable tire side slip angles are often encountered, the dynamic bicycle model is chosen as the

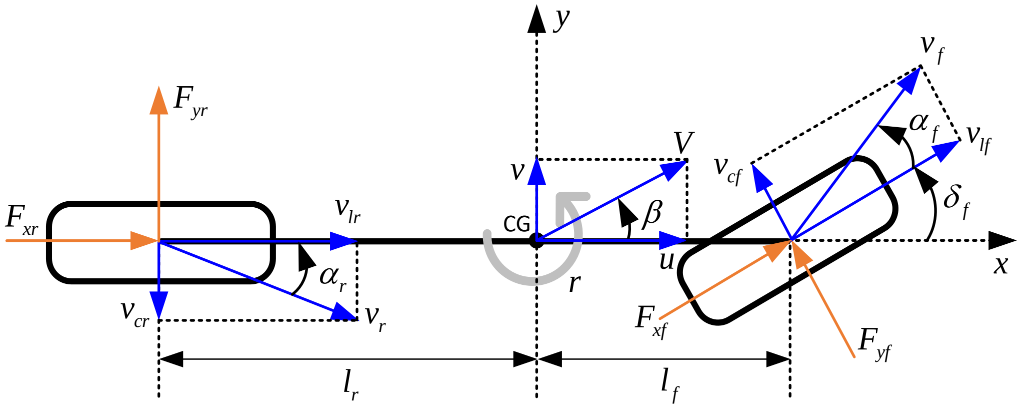


Figure 4.3: Dynamic bicycle model

appropriate model for more accurate and reliable control analysis. Its ability to capture the complex vehicle dynamics at play during such maneuvers ensures that the control strategy developed based on this model can effectively handle challenging stability control scenarios and contribute to improved safety and performance in vehicle control systems.

The dynamic bicycle model is illustrated in Figure 4.3. Using Newton's second law for motion, the derivatives of lateral velocity and yaw rate are calculated as

$$\dot{v} = \frac{F_{yf} + F_{yr}}{m} - ur \tag{4.5}$$

$$\dot{r} = \frac{F_{yf} l_f - F_{yr} l_r + M_c}{I_z} \tag{4.6}$$

where F_{yf} and F_{yr} denote the front and rear lateral tire forces, respectively; I_z is the vehicle's moment of inertia; M_c represents the adjusted yaw moment.

The Fiala tire model (also known as the brush model) presented by Pacejka [105] is used to represent the lateral tire forces as

$$F_y = \begin{cases} C_\alpha \tan\alpha - \dfrac{C_\alpha{}^2}{3\mu F_z}\left|\tan\alpha\right|\tan\alpha + \dfrac{C_\alpha{}^3}{27\mu^2 F_z{}^2}\tan^3\alpha & \left|\alpha\right| \le \arctan(\dfrac{3\mu F_z}{C_\alpha}) \\ \mu F_z \operatorname{sgn}\alpha & \text{otherwise} \end{cases} \tag{4.7}$$

where C_α is the nominal tire cornering stiffness; F_z is the normal load on tires; μ is the road friction coefficient; α is the slip angle of tires.

Based on the assumptions that the vehicle is only front steering, the front tire slip angle and the rear tire slip angle are calculated as

$$\alpha_f = \frac{v + l_f r}{u} - \delta_f \tag{4.8}$$

and

$$\alpha_r = \frac{v - l_r r}{u} \tag{4.9}$$

The nonlinearity of the lateral tire force makes the vehicle dynamics model nonlinear. The nonlinear tire force is linearized at each time step as an affine function of the current tire slip angle to reduce the computational complexity.

As shown in Figure 4.4, the approximated lateral tire force is given as

$$F_y = \bar{F}_y - \bar{C}_\alpha \left(\alpha - \bar{\alpha} \right) \tag{4.10}$$

where $\bar{C}_\alpha = \frac{\partial F_y}{\partial \alpha}\Big|_{\alpha=\bar{\alpha}}$ is the equivalent tire cornering stiffness at the current tire slip angle $\bar{\alpha}$.

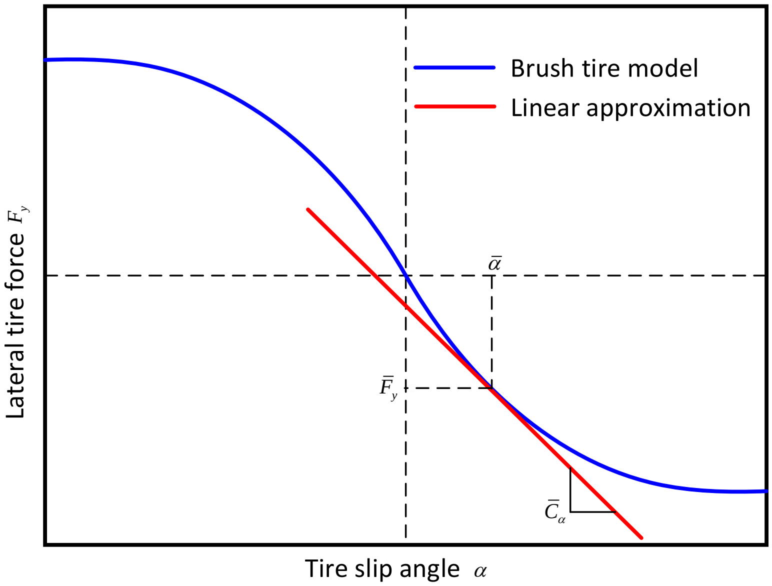


Figure 4.4: Brush tire model with affine approximation

4.3.2 Nominal Prediction Model

The above dynamic vehicle model can be written in the format of a state-space equation as

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} + \mathbf{Dw} + \mathbf{d} \\ \mathbf{y} = \mathbf{Cx} \end{cases} \tag{4.11}$$

where $\mathbf{x} = \begin{bmatrix} v & r \end{bmatrix}^T$ denotes the system states vector; $\mathbf{u} = \begin{bmatrix} M_c \end{bmatrix}^T$ denotes the control input vector including the yaw moment adjustment M_c; $\mathbf{w} = \begin{bmatrix} \delta_d \end{bmatrix}$ denotes the uncontrolled input vector including the driver steering angle δ_d; $\mathbf{y} = \begin{bmatrix} \beta & r \end{bmatrix}^T$ denotes the system output vector; $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$, and $\mathbf{d}$ are matrixes which are defined as

$$\mathbf{A} = \begin{bmatrix} -\dfrac{\bar{C}_{\alpha f} + \bar{C}_{\alpha r}}{mu} & \dfrac{\bar{C}_{\alpha r} l_r - \bar{C}_{\alpha f} l_f}{mu} - u \\ \dfrac{\bar{C}_{\alpha r} l_r - \bar{C}_{\alpha f} l_f}{I_z u} & -\dfrac{\bar{C}_{\alpha f} l_f^{\,2} + \bar{C}_{\alpha r} l_r^{\,2}}{I_z u} \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} 0 & \dfrac{\bar{C}_{\alpha f}}{m} \\ \dfrac{1}{I_z} & \dfrac{\bar{C}_{\alpha f} l_f}{I_z} \end{bmatrix},$$

$$\mathbf{D} = \begin{bmatrix} \dfrac{\bar{C}_{\alpha f}}{m} \\ \dfrac{\bar{C}_{\alpha f} l_f}{I_z} \end{bmatrix}, \ \mathbf{C} = \begin{bmatrix} 1/u & 0 \\ 0 & 1 \end{bmatrix}, \text{ and } \mathbf{d} = \begin{bmatrix} \dfrac{\bar{F}_f + \bar{F}_r + \bar{C}_{\alpha f}\bar{\alpha}_f + \bar{C}_{\alpha r}\bar{\alpha}_r}{m} \\ \dfrac{\bar{F}_f l_f - \bar{F}_r l_r + \bar{C}_{\alpha f}\bar{\alpha}_f l_f - \bar{C}_{\alpha r}\bar{\alpha}_r l_r}{I_z} \end{bmatrix}.$$

This continuous model needs to be discretized before it can be used for designing the controller. In this thesis, Euler approximation is used to perform the discretization as

$$\begin{cases} \mathbf{A}^k = I + T * \mathbf{A} \\ \mathbf{B}^k = T * \mathbf{B} \\ \mathbf{D}^k = T * \mathbf{D} \\ \mathbf{C}^k = \mathbf{C} \\ \mathbf{d}^k = T * \mathbf{d} \end{cases} \tag{4.12}$$

where T denotes the sampling time of the controller.

Therefore, the discrete representation of the nominal prediction model can be written as

$$\begin{cases} \bar{\mathbf{x}}(k+1) = \mathbf{A}^k \bar{\mathbf{x}}(k) + \mathbf{B}^k \mathbf{u}(k) + \mathbf{D}^k \mathbf{w}(k) + \mathbf{d}^k \\ \bar{\mathbf{y}}(k+1) = \mathbf{C}^k \bar{\mathbf{x}}(k+1) \end{cases} \tag{4.13}$$

By adding a learning module $\mathbf{O}^{k}$ to the nominal equations, the discrete representation of the learned prediction model can be written as

$$\begin{cases} \tilde{\mathbf{x}}(k+1) = \mathbf{A}^k\tilde{\mathbf{x}}(k) + \mathbf{B}^k\mathbf{u}(k) + \mathbf{D}^k\mathbf{w}(k) + \mathbf{d}^k + \mathbf{O}^{k} \\ \tilde{\mathbf{y}}(k+1) = \mathbf{C}^k\tilde{\mathbf{x}}(k+1) \end{cases} \tag{4.14}$$

With the definitions of $\mathbf{X}(k) = \left[\tilde{\mathbf{x}}(k) \quad \mathbf{u}(k-1)\right]^T$ and $\Delta\mathbf{u}(k) = \mathbf{u}(k) - \mathbf{u}(k-1)$, the incremental mode of the above state-space equation is obtained as

$$\begin{cases} \mathbf{X}(k+1) = \mathbf{A}_k\mathbf{X}(k) + \mathbf{B}_k\Delta\mathbf{u}(k) + \mathbf{D}_k\mathbf{w}(k) + \mathbf{d}_k + \mathbf{O}_k \\ \tilde{\mathbf{y}}(k+1) = \mathbf{C}_k\mathbf{X}(k+1) \end{cases} \tag{4.15}$$

where the expanded matrixes $\mathbf{A}_k$, $\mathbf{B}_k$, $\mathbf{D}_k$, $\mathbf{d}_k$, $\mathbf{O}_k$, and $\mathbf{C}_k$ are defined as

$$\mathbf{A}_k = \begin{bmatrix} \mathbf{A}^k & \mathbf{B}^k \\ \mathbf{0} & \mathbf{I} \end{bmatrix},\ \mathbf{B}_k = \begin{bmatrix} \mathbf{B}^k \\ \mathbf{I} \end{bmatrix},\ \mathbf{D}_k = \begin{bmatrix} \mathbf{D}^k \\ \mathbf{0} \end{bmatrix},\ \mathbf{d}_k = \begin{bmatrix} \mathbf{d}^k \\ \mathbf{0} \end{bmatrix},\ \mathbf{O}_k = \begin{bmatrix} \mathbf{O}^{k} \\ \mathbf{0} \end{bmatrix}, \mathbf{C}_k = \begin{bmatrix} \mathbf{C}^k & \mathbf{0} \end{bmatrix}.$$

It is assumed that the steering angle from the driver maintains a constant value within the prediction horizon. Therefore, the predicted system states within the prediction horizon can be obtained as

$$\begin{aligned} \begin{bmatrix} \mathbf{X}(k+1) \\ \mathbf{X}(k+2) \\ \vdots \\ \mathbf{X}(k+N_p) \end{bmatrix} &= \begin{bmatrix} \mathbf{A}_k \\ \mathbf{A}_k{}^2 \\ \vdots \\ \mathbf{A}_k{}^{N_p} \end{bmatrix} \mathbf{X}(k) + \begin{bmatrix} \mathbf{B}_k & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}_k\mathbf{B}_k & \mathbf{B}_k & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_k{}^{N_p-1}\mathbf{B}_k & \mathbf{A}_k{}^{N_p-2}\mathbf{B}_k & \cdots & \mathbf{B}_k \end{bmatrix} \begin{bmatrix} \Delta\mathbf{u}(k) \\ \Delta\mathbf{u}(k+1) \\ \vdots \\ \Delta\mathbf{u}(k+N_p-1) \end{bmatrix} + \\ &\begin{bmatrix} \mathbf{D}_k & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}_k\mathbf{D}_k & \mathbf{D}_k & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_k{}^{N_p-1}\mathbf{D}_k & \mathbf{A}_k{}^{N_p-2}\mathbf{D}_k & \cdots & \mathbf{D}_k \end{bmatrix} \begin{bmatrix} \mathbf{w}(k) \\ \mathbf{w}(k) \\ \vdots \\ \mathbf{w}(k) \end{bmatrix} + \begin{bmatrix} \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}_k & \mathbf{I} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_k{}^{N_p-1} & \mathbf{A}_k{}^{N_p-2} & \cdots & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{d}_k + \mathbf{O}_k \\ \mathbf{d}_{k+1} + \mathbf{O}_{k+1} \\ \vdots \\ \mathbf{d}_{k+N_p-1} + \mathbf{O}_{k+N_p-1} \end{bmatrix} \end{aligned} \tag{4.16}$$

And the predicted system output can be obtained as

$$\begin{bmatrix} \mathbf{Y}(k+1) \\ \mathbf{Y}(k+2) \\ \vdots \\ \mathbf{Y}(k+N_p) \end{bmatrix} = \begin{bmatrix} \mathbf{C}_k\mathbf{X}(k+1) \\ \mathbf{C}_k\mathbf{X}(k+2) \\ \vdots \\ \mathbf{C}_k\mathbf{X}(k+N_p) \end{bmatrix} \tag{4.17}$$

Equation (4.17) and (4.18) can be written in the compact form as

$$\begin{cases} \vec{\mathbf{X}} = \mathbf{S}_x \mathbf{X}(k) + \mathbf{S}_u \Delta\vec{\mathbf{u}} + \mathbf{S}_w \vec{\mathbf{w}} + \mathbf{d}^k + \mathbf{O}^k \\ \vec{\mathbf{Y}} = \mathbf{C}^k \vec{\mathbf{X}} \end{cases} \tag{4.18}$$

4.4 Learning-based Vehicle Model

4.4.1 Variable Importance Analysis

In a regression model, some input variables have larger influences on the outputs, whereas others have a smaller influence on the outputs. The analysis of variable importance can be useful for determining where simulation and experimental efforts should be focused; it is not necessary to spend a lot of effort for calculating or measuring variables that do not have a significant effect on the concerned output responses. In this study, the important variables for the prediction of lateral velocity and yaw rate are identified using both the vehicle dynamics equations and the measure of the Pearson correlation coefficient (PCC).

The vehicle dynamics equations are mathematical models that describe the motion and behavior of a vehicle. These equations are derived from the principles of physics and mechanics, considering various forces and moments acting on the vehicle during motion. By analyzing these equations, parameters that have direct or indirect effects on the lateral velocity and yaw rate can be determined. Based on the vehicle dynamics discussed above, the lateral and yaw dynamics of a vehicle can be obtained using a linear tire model as

$$\dot{v} = -\frac{\bar{C}_{\alpha f} + \bar{C}_{\alpha r}}{mu} v + \left(\frac{\bar{C}_{\alpha r} l_r - \bar{C}_{\alpha f} l_f}{mu} - u \right) r + \frac{\bar{C}_{\alpha f}}{m} \delta_f \tag{4.19}$$

$$\dot{r} = \frac{\bar{C}_{\alpha r} l_r - \bar{C}_{\alpha f} l_f}{I_z u} v - \frac{\bar{C}_{\alpha r} l_r^2 + \bar{C}_{\alpha f} l_f^2}{I_z u} r + \frac{\bar{C}_{\alpha f} l_f}{I_z} \delta_f \tag{4.20}$$

This continuous model can be discretized with the Euler approximation as

$$v^{k+1} = \left(1 - \frac{\bar{C}_{\alpha f} + \bar{C}_{\alpha r}}{mu^k} T \right) v^k + \left(\frac{\bar{C}_{\alpha r} l_r - \bar{C}_{\alpha f} l_f}{mu^k} - u^k \right) T r^k + \frac{\bar{C}_{\alpha f}}{m} T \delta_f^{\ k} \tag{4.21}$$

$$r^{k+1} = \frac{\bar{C}_{\alpha r} l_r - \bar{C}_{\alpha f} l_f}{I_z u^k} T v^k + \left(1 - \frac{\bar{C}_{\alpha r} l_r^2 + \bar{C}_{\alpha f} l_f^2}{I_z u^k} T \right) r^k + \frac{\bar{C}_{\alpha f} l_f}{I_z} T \delta_f^{\ k} \tag{4.22}$$

Another approach to predict the lateral dynamics is to use the following kinematic model of the vehicle [106]:

$$\dot{v} = -ur + a_y \tag{4.23}$$

This continuous model can also be discretized with the Euler approximation as

$$v^{k+1} = v^k - u^k r^k T + a_y{}^k T \tag{4.24}$$

Based on these mathematical equations, it can be noticed that steering angle, vehicle speed, lateral acceleration, historical lateral velocity, and historical yaw rate have direct or indirect effects on the lateral velocity and yaw rate.

The variables analyzed for the predictions of v^k and r^k using the PCC method include steering angle δ^{k-1}, steering rate $\dot{\delta}^{k-1}$, vehicle speed u^{k-1}, lateral velocity v^{k-1}, yaw rate r^{k-1}, longitudinal acceleration $a_x{}^{k-1}$, and lateral acceleration $a_y{}^{k-1}$. The PCC matrices for lateral velocity and yaw rate are calculated based on the previously collected dataset on the targeted vehicle and expressed in the form of heatmaps, shown in Figure 4.5. The value of 1.00 in the main diagonal going from the top left to the bottom right shows that each variable always perfectly correlates with itself. It can be found in the figure that steering angle δ^{k-1}, vehicle speed u^{k-1}, historical lateral velocity v^{k-1}, historical yaw rate r^{k-1}, and lateral acceleration $a_y{}^{k-1}$ have relatively strong correlations with lateral velocity v^k and yaw rate r^k.

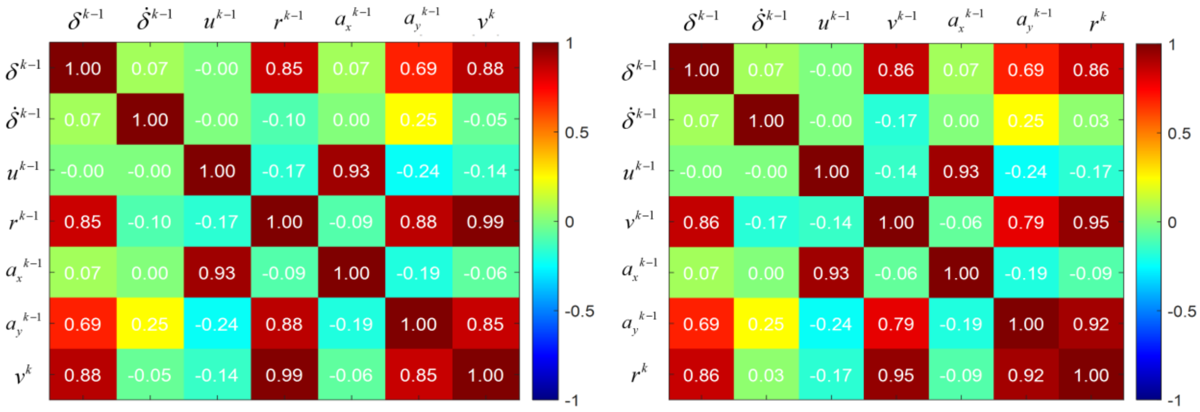
Based on the above mathematical analysis and PCC analysis, the variables that are selected to predict yaw rate and lateral velocity include: steering angle, vehicle speed, lateral acceleration, historical lateral velocity, and historical yaw rate.

4.4.2 Data Collection for Training

A dataset with input features and output responses of the vehicle is collected for training GPR. As discussed in the previous subsection, the input features for v and r at time step k are selected as

$$\mathbf{a}_v^k = \left(\delta^k, u^k, v^k, r^k, a_y^k\right) \tag{4.25}$$

$$\mathbf{a}_r^k = \left(\delta^k, u^k, v^k, r^k, a_y^k\right) \tag{4.26}$$



(a) PCC matrix for lateral velocity

(b) PCC matrix for yaw rate

Figure 4.5: PCC matrix for lateral velocity and yaw rate.

In order to predict the vehicle responses that are one step ahead of the current states, the observed experience data in a training trip are paired as $\left(\mathbf{a}_v^{k-1}, v^k\right)$ and $\left(\mathbf{a}_r^{k-1}, r^k\right)$ for v and r, respectively. Observations for all sampling times in a trip are collected and organized into two sets of data pairs for v and r as

$$\mathrm{D}_v^j = \left\{\left(\mathbf{a}_v^0, v^1\right), \ldots, \left(\mathbf{a}_v^{n-1}, v^n\right)\right\} \tag{4.27}$$

$$\mathrm{D}_r^j = \left\{\left(\mathbf{a}_r^0, r^1\right), \ldots, \left(\mathbf{a}_r^{n-1}, r^n\right)\right\} \tag{4.28}$$

where n denotes the number of time-steps during the trip j.

After j trips, multiple subsets have been collected. They are combined into two supersets for v and r as

$$\mathrm{D}_v = \left\{\mathrm{D}_v^1, \ \ldots, \mathrm{D}_v^j\right\} \tag{4.29}$$

$$\mathrm{D}_r = \left\{\mathrm{D}_r^1, \ \ldots, \mathrm{D}_r^j\right\} \tag{4.30}$$

These supersets collected offline are stored in the controller as the training dataset for the online prediction of v and r. Besides, they are automatically updated online to capture the latest information.

4.4.3 Data Management for Prediction

The most prominent drawback of GPR is that it suffers from a cubic time complexity $\mathrm{O}\left(n^3\right)$ because of the inversion of the $n \times n$ kernel matrix, where n denotes the number of observations [107]. As n becomes large, the prediction computation can become exceedingly expensive. To address the computational complexity issue with large datasets, an effective approach is to choose a smaller subset of observations from the total n observations and then apply the GPR model to these m points for predictions. This reduced set is referred to as the active set or inducing input set. This approximation technique is known as the Subset-of-Data (SoD) method. By employing the SoD method, the computational burden is significantly alleviated while still providing reasonably accurate estimates and predictions for the GPR model. With a subset comprising m data points, the time complexity of GPR is reduced from $\mathrm{O}\left(n^3\right)$ to $\mathrm{O}\left(m^3\right)$.

In the SoD method for GPR, the goal is to select a smaller subset of data points from the original dataset to reduce computational complexity while maintaining the model's accuracy. There are various strategies for selecting the subset of data points.

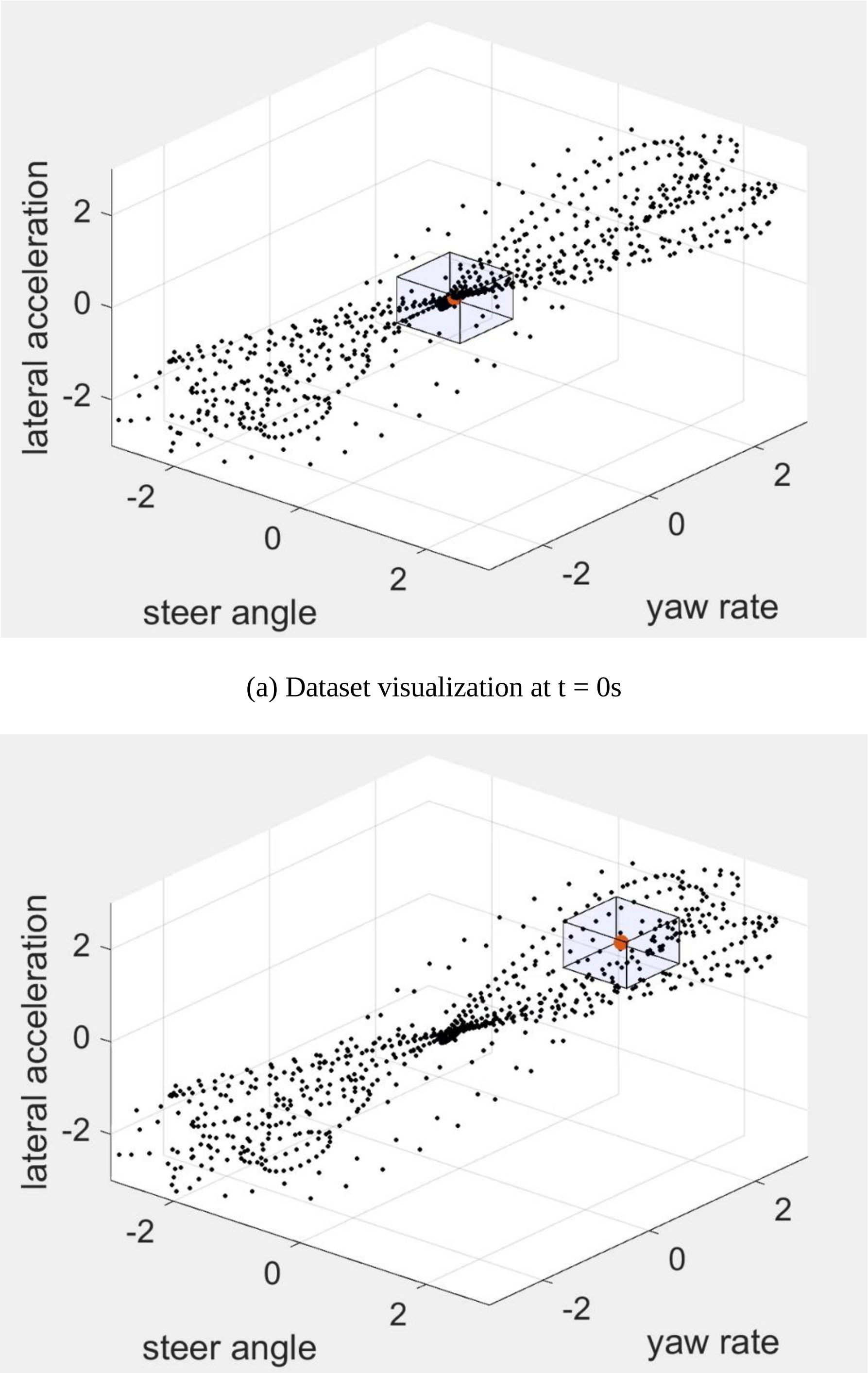
(a) Random Subset Selection: This method involves randomly selecting a subset of data points from the original dataset. The simplicity of this approach is a key advantage, as it is easy to implement and computationally efficient. However, because the selection is random, there is no guarantee that the chosen subset will provide the best representation of the data distribution. In some cases, important patterns or characteristics of the data might be missed due to random chance.

(b) K-means Clustering: In this method, the original dataset is partitioned into K clusters using the K-means clustering algorithm. Each cluster contains a group of data points that are similar to each other in terms of their features. Then, representative points (centroids) are chosen from each cluster to form the subset. By selecting centroids, this approach can offer a more structured representation of the data and potentially better coverage of the data distribution compared to random selection. However, there is a trade-off with K-means clustering. As the dataset size increases, the computational cost of the clustering algorithm grows significantly, which can become a limiting factor, especially for large datasets.

The approach in the study involves selecting a subset of data points using a bounding box. This bounding box is a rectangular region that encapsulates the original dataset. The bounds of the box are determined through calibration for each feature (dimension), likely based on the maximum and minimum values of the features in the dataset. Within the bounding box, a fixed number of representative points are chosen to form the subset. These representative points can be selected either randomly or using some specific strategies. The selection of the number of points in the subset is a trade-off between achieving good model performance and managing computational cost. Selecting a smaller number of points reduces computational complexity, but it might not fully capture the data distribution and could potentially lead to a loss of model accuracy. On the other hand, selecting a larger number of points might improve accuracy but comes with increased computational cost. In the study, the 6 nearest data points from the testing point within the bounding box are selected. This means that when a new data point needs to be predicted, a bounding box around that point is formed, and then the 6 data points (if available) from the original dataset that are closest to the testing point within that box are selected. The idea is to use the information from these nearby points to make predictions for the new data point. Figure 4.6 illustrates an example of this process. In this visual representation, a three-dimensional feature space is employed to depict the dataset's characteristics. The axes correspond to the lateral acceleration, steering angle, and yaw rate parameters. Each black dot within the visualization signifies a data point that has been collected and recorded. The red dot represents the current operational state known as the query point. The bounding box is superimposed around the query point. This bounding box effectively delineates the region of interest within the feature space. Within this limited region of interest, a subset of data points is selected and designated as training points. While this approach might be a simple and effective starting point, there are more robust data selection strategies that can be studied in the future.

In the context of online management of data in GPR, when dealing with new data points, a common strategy is to handle them differently based on their similarity to existing data points. Specifically, if a new data point is sufficiently different from the existing ones, it can be added to the dataset without removing any old points. On the other hand, if the new data point is close to one or more existing points, the nearest existing point(s) will be replaced to maintain a diverse and representative subset. This process can be summarized as follows:

Checking for Data Similarity: When a new data point becomes available, its similarity to the existing data points in the dataset is assessed. This similarity can be measured using a distance metric, such as Euclidean distance, depending on the feature space.



(a) Dataset visualization at t = 0s

(b) Dataset visualization at t = 10s

Figure 4.6: Illustration of the data selection method.

Different Data Point: If the new data point is significantly different from all the existing data points, it is considered a "different" data point. In this case, the new data point can be added to the dataset without removing any old points. By doing so, the dataset gradually grows to represent a broader range of the underlying data distribution.

Close to Existing Data: If the new data point is close to one or more existing data points, it is considered similar to the existing data, possibly representing redundant or overlapping information. In this case, one or more of the existing data points can be replaced by the new data point to maintain a compact yet representative subset.

Adding and Removing: The new data point replaces the nearest existing data point(s) in the dataset. The choice of how many data points to replace depends on the specific strategy and the desired size of the dataset. For example, if you want to maintain a fixed-size dataset, you can replace the nearest point only. If you have a flexible dataset size, you can replace multiple nearest points.

This approach ensures that the dataset adapts to new information efficiently. It retains diverse and essential data points while avoiding excessive redundancy. By continuously monitoring the data points' similarity and updating the dataset accordingly, the GPR model can maintain its accuracy and generalization capabilities over time as the underlying data distribution evolves.

4.5 Model Authentication

As discussed above, the learning MPC structure maintains two models: (1) the physics-based nominal vehicle dynamics model and (2) the data-driven learned vehicle dynamics model. The nominal physics model is typically robust by complying with the laws of physics but usually lacks the nonlinear and unmodelled behavior of the vehicle. The learned model can evolve as more experience data is observed, but its performance highly depends on the training or data availability. One of the beauties of the learning MPC method is that it provides the flexibility to select the proper one that has higher accuracy and robustness at each control interval.

In this study, a model authentication strategy is created to evaluate the credibility of both models and choose the proper one to be used for the predictions at each control interval. The model authentication strategy mainly considers the necessity to use the learned model, the reliability of the learned model, and the accuracy of the learned model. As discussed in the background chapter, GPR provides not only point predictions but also uncertainty estimates in the form of credible intervals. Credible intervals represent a range of values within which the true value of the output is likely to lie with a certain confidence level. GPR model's uncertainty quantification can be evaluated using the width of credible intervals. Wider credible intervals indicate higher uncertainty, while narrower credible intervals suggest a well-performing prediction. In this study, the GPR model will only be used to enhance the prediction when the credible interval of GPR is larger than a predefined value, which means the GPR model has enough data or confidence.

4.6 Cost Function and Constraints

The purpose of solving a typical tracking problem is to find an optimal sequence of control inputs to minimize a predefined cost function with constraints. Considerations in the cost function typically include: (1) the deviation between the desired system responses and the measured or estimated system responses, (2) the effort of control actions, and (3) the chatter of control actions. The cost function in this study is defined as

$$J = \sum_{i=1}^{N_p} \left(\left\| \mathbf{Y}(k+i) - \mathbf{Y}_{des}(k+i) \right\|_{\mathbf{Q}}^2 + \left\| \mathbf{u}(k+i) \right\|_{\mathbf{R}}^2 + \left\| \Delta\mathbf{u}(k+i) \right\|_{\mathbf{P}}^2 \right) \tag{4.31}$$

where $\mathbf{Y}_{des}$ is the desired vehicle responses for yaw rate. The first term in the cost function is the tracking error of the concerned vehicle responses, the second term is the values of control inputs, i.e., torque adjustment T^{FL} and T^{FR}, and the last term is the rate of change of the control inputs. $\mathbf{Q}$, $\mathbf{R}$, and $\mathbf{P}$ are weighting matrices reflecting the relative importance of the three terms in the cost function.

The constraints for torque adjustment result from the physical limitations that the propulsion motors can apply to the wheels. The constraints for torque adjustment for the front left motor T^{FL} and the front right motor T^{FR} are defined as

$$T_{\min}^{FL} \le T^{FL} \le T_{\max}^{FL} \tag{4.32}$$

$$T_{\min}^{FR} \le T^{FR} \le T_{\max}^{FR} \tag{4.33}$$

4.7 Summary

This chapter introduces a novel approach to holistic vehicle control using a hybrid learning Model Predictive Control (MPC) controller. The hybrid learning MPC scheme leverages two prediction models: a physics-based nominal vehicle dynamics model and a data-driven learned vehicle dynamics model. The nominal model incorporates a dynamic bicycle model and a brush tire model to capture the vehicle and tire dynamics. The learned model employs Gaussian Process Regression (GPR) to model the uncertainty that the nominal model does not capture. To enhance the accuracy of predictions for yaw rate, essential variables were identified using vehicle dynamics equations and Pearson correlation coefficient (PCC). Based on mathematical analysis and PCC analysis, the variables that are selected to predict yaw rate and lateral velocity include: steering angle, vehicle speed, lateral acceleration, historical lateral velocity, and historical yaw rate. To facilitate model selection for predictive control, the chapter outlines data collection, updating, and management strategies. A data management strategy was developed to retain diverse and essential data points while avoiding excessive redundancy. By continuously monitoring the data points' similarity and updating the dataset accordingly, the GPR model can maintain its accuracy and generalization capabilities over time as the underlying data distribution evolves. Additionally, a model authentication strategy was devised to assess the accuracy and reliability of both models. Based on the evaluation results, the controller can choose either the nominal or learned model for optimal predictive control. Finally, the chapter elaborates on defining the cost function and constraints for HVC. An evaluation of the developed learning MPC controller will be presented in the next chapter.

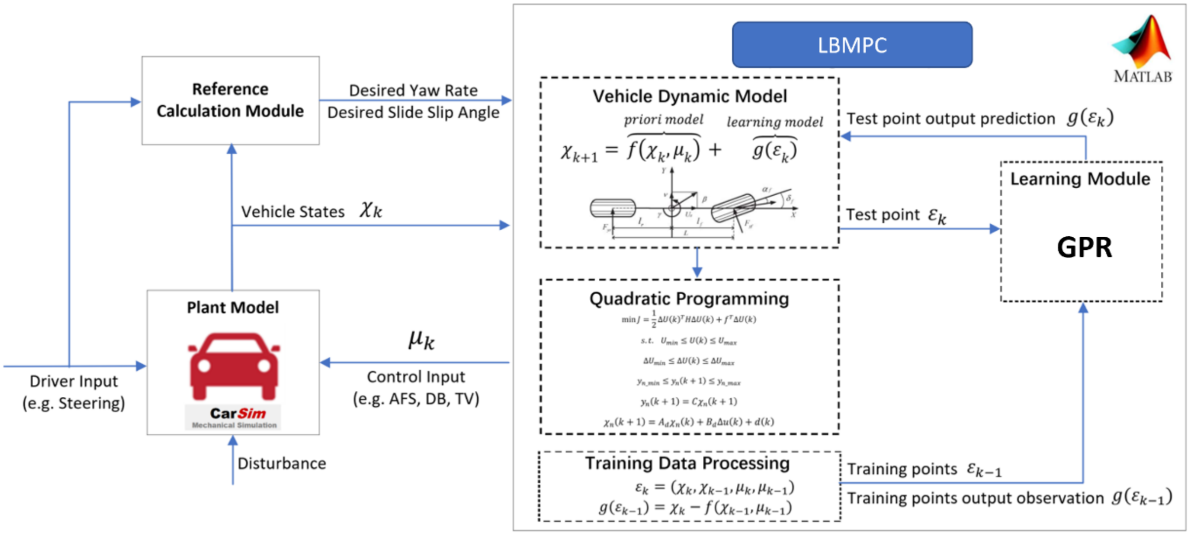
Chapter 5

Simulation and Experimental Verification of Learning MPC

In this chapter, the performance of the developed hybrid learning MPC controller is evaluated through simulation and experiments. A high-fidelity CarSim model is used to represent the vehicle dynamics in the simulation. The designed learning MPC controller is implemented in Matlab and evaluated through the Matlab/Simulink CarSim co-simulation under a series of driving maneuvers. Experimental verification of the developed hybrid learning MPC is also presented.

5.1 Simulation Model and Verification

The Matlab/Simulink CarSim co-simulation environment is built to evaluate the designed hybrid learning MPC controller, shown as Figure 5.1. To simulate the vehicle dynamics accurately, a high-fidelity CarSim model was developed specifically for the Chevrolet Equinox, shown as Figure 5.2. The main parameters of the vehicle are specified in Table 5.1. These parameters include characteristics such as vehicle mass, dimensions, and tire properties. This model considers various factors that influence the vehicle's behavior, including tire dynamics, suspension characteristics, and mass distribution. To verify the accuracy of the CarSim model, experimental data was collected in a double lane change (DLC) maneuver performed on an Equinox testing vehicle.



LBMPC
MATLAB
Reference Calculation Module
Desired Yaw Rate
Desired Slide Slip Angle
Vehicle Dynamic Model
priori model
learning model
$\chi_{k+1} = \overbrace{f(\chi_k, \mu_k)}^{\text{priori model}} + \overbrace{g(\varepsilon_k)}^{\text{learning model}}$
Test point output prediction $g(\varepsilon_k)$
Learning Module
GPR
Test point ε_k
Vehicle States χ_k
Quadratic Programming
$\min J = \frac{1}{2}\Delta U(k)^T H \Delta U(k) + f^T \Delta U(k)$
$s.t.\ \ U_{min} \le U(k) \le U_{max}$
$\Delta U_{min} \le \Delta U(k) \le \Delta U_{max}$
$y_{n\_min} \le y_n(k+1) \le y_{n\_max}$
$y_n(k+1) = C\chi_n(k+1)$
$\chi_n(k+1) = A_d\chi_n(k) + B_d\Delta u(k) + d(k)$
Plant Model
μ_k
Control Input (e.g. AFS, DB, TV)
Driver Input (e.g. Steering)
CarSim
Mechanical Simulation
Disturbance
Training Data Processing
$\varepsilon_k = (\chi_k, \chi_{k-1}, \mu_k, \mu_{k-1})$
$g(\varepsilon_{k-1}) = \chi_k - f(\chi_{k-1}, \mu_{k-1})$
Training points ε_{k-1}
Training points output observation $g(\varepsilon_{k-1})$

Figure 5.1: Matlab/Simulink CarSim co-simulation structure for HVC

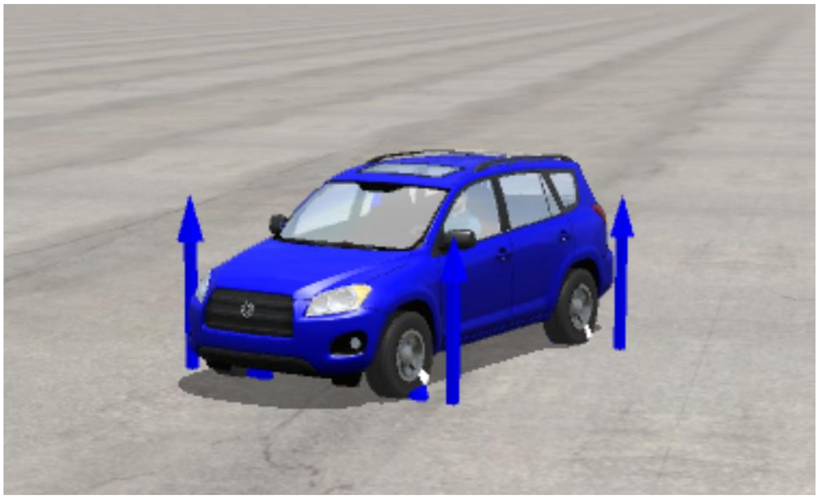


Figure 5.2: Chevrolet Equinox vehicle model in CarSim.

Table 5.1: Specification of the vehicle in CarSim.

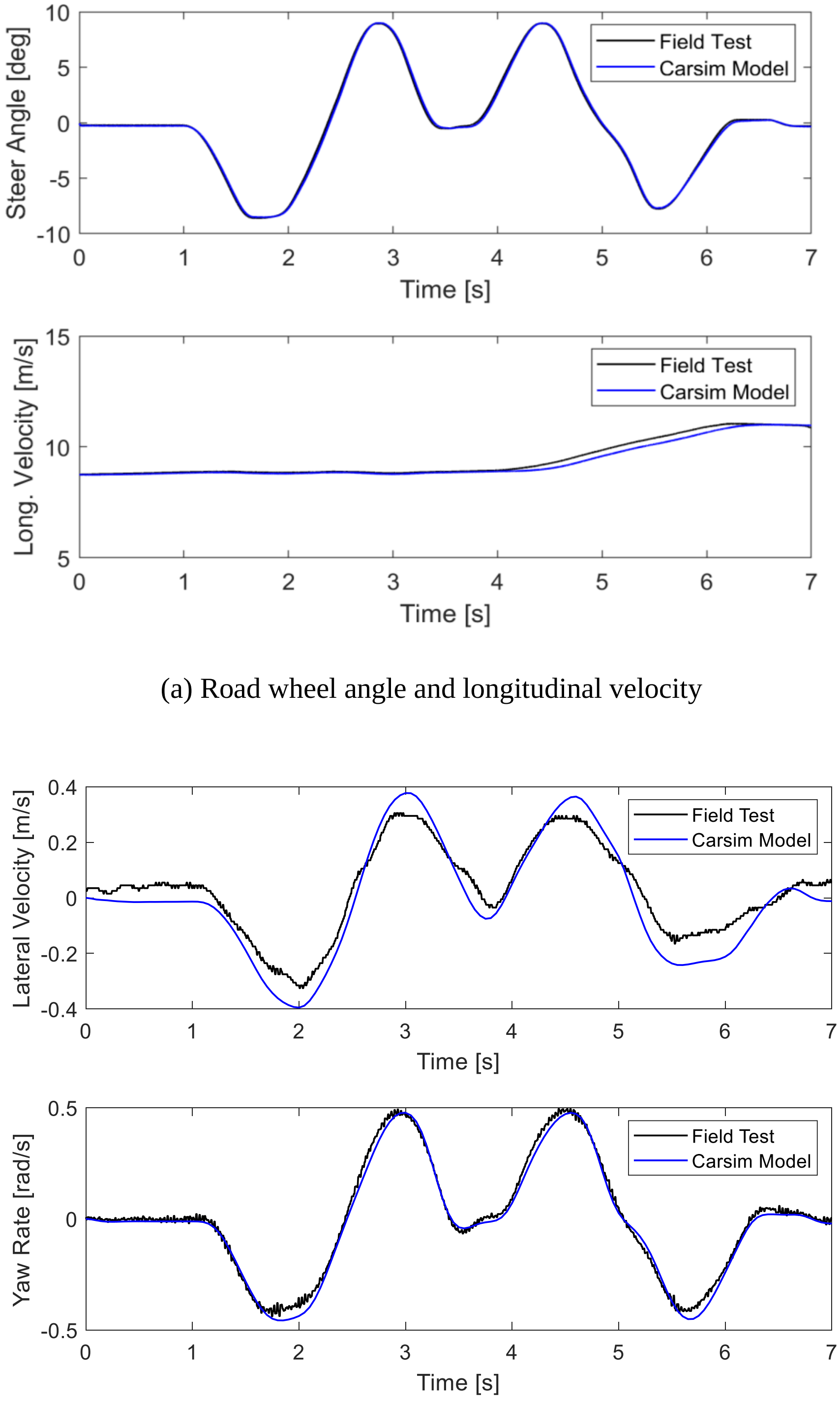
Parameter	Unit	Value	Description
m	[kg]	2257	Vehicle mass
L_{wb}	[m]	3.14	Wheelbase
l_f	[m]	1.33	CG distance to front axle
l_r	[m]	1.81	CG distance to rear axle
H_{CG}	[m]	0.78	CG hight
R_e	[m]	0.368	Effective radius of the tires
l_s	[m]	1.725	Front and rear track width
I_z	[kg.m^2]	3525	Vehicle yaw moment of inertia
$C_{\alpha f}$	[N/rad]	152343	Front tires cornering stiffness
$C_{\alpha r}$	[N/rad]	121943	Rear tires cornering stiffness

Figure 5.3 illustrates a comparison between experimental data collected from the Equinox testing vehicle and simulation results obtained from the CarSim vehicle model. The yaw rate is measured through an IMU (Inertial Measurement Unit), and the lateral velocity is measured by GNSS (Global Navigation Satellite System). The comparison is based on a DLC maneuver performed with a speed range from 8 m/s to 12 m/s on a dry road. In the figure, the road wheel steering angle and longitudinal velocity profiles applied during both the field test and the CarSim simulation are shown. To verify the

accuracy of the CarSim model, the speed and steering angle profiles are kept almost the same as those used in the real-world field test. The responses of lateral velocity and yaw rate from the CarSim model are compared with those from the real testing vehicle. The comparison demonstrates an acceptable fit for both lateral velocity and yaw rate, indicating that the CarSim model accurately captures the vehicle's dynamic behavior during the DLC maneuver. The relatively large difference observed in lateral velocity between the CarSim vehicle model, and the real testing vehicle can be attributed to several factors. One major factor is the variations in road conditions and tire parameters between the simulated environment and the real-world testing scenario. In the CarSim model, the road conditions and tire parameters are typically based on theoretical assumptions or simplified data. On the other hand, in real-world testing, road surfaces can be highly variable, with different levels of roughness, friction, and irregularities that can significantly affect the vehicle's lateral dynamics. Additionally, accurately measuring and modeling the tire behavior and properties, such as tire slip angles, tire stiffness, and tire-road interaction, are challenging tasks in field tests due to practical limitations. Furthermore, measurement errors in both the experimental data collection and the simulation can contribute to the observed differences of lateral velocity. These errors may arise from sensor inaccuracies of GNSS. Despite this difference, the agreement between the experimental and simulated results is sufficient for the purposes of subsequent simulations and evaluations conducted in this study. Overall, the developed CarSim vehicle model provides a satisfactory level of accuracy, allowing it to be used effectively in further simulations and evaluations.

5.2 Flick Maneuver Simulation

In this simulation, a flick maneuver on a wet surface is used to evaluate the performance of the developed hybrid learning MPC for HVC. Figure 5.4 illustrates the steering input and the speed of the vehicle in the flick maneuver, which is commonly employed to induce a vehicle to drift without using acceleration or braking. The road surface is slippery, with a friction coefficient ($\mu = 0.5$). In this driving scenario, the vehicle speed is maintained at the same speed of 60 km/h. The driver steers to the right and immediately counter steers while maintaining the position of the steering wheel. The main goal of this driving scenario was to evaluate the performance using the developed hybrid learning MPC control theory compared to the case using a conventional MPC and the case without MPC control.



(a) Road wheel angle and longitudinal velocity

(b) Lateral velocity and yaw rate response

Figure 5.3: CarSim model verification under a DLC maneuver.

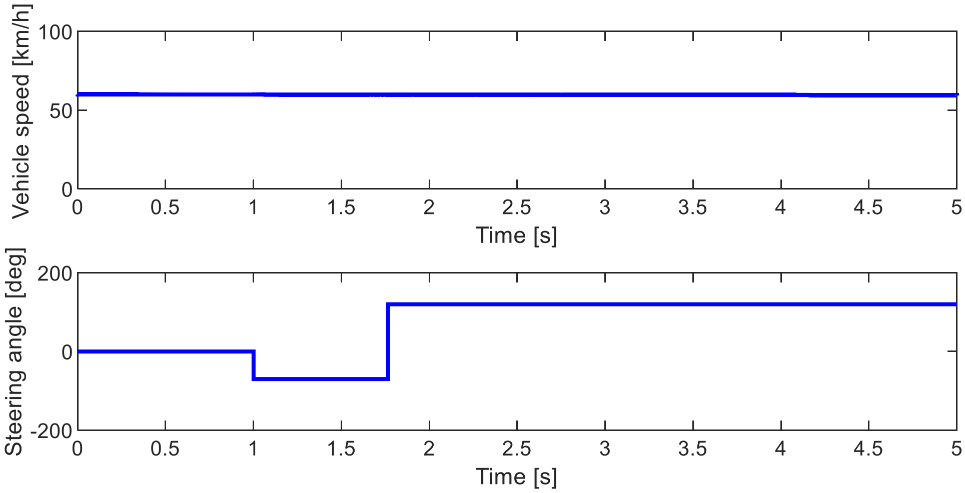


Figure 5.4: Vehicle speed and steering wheel input in flick maneuver on wet surface.

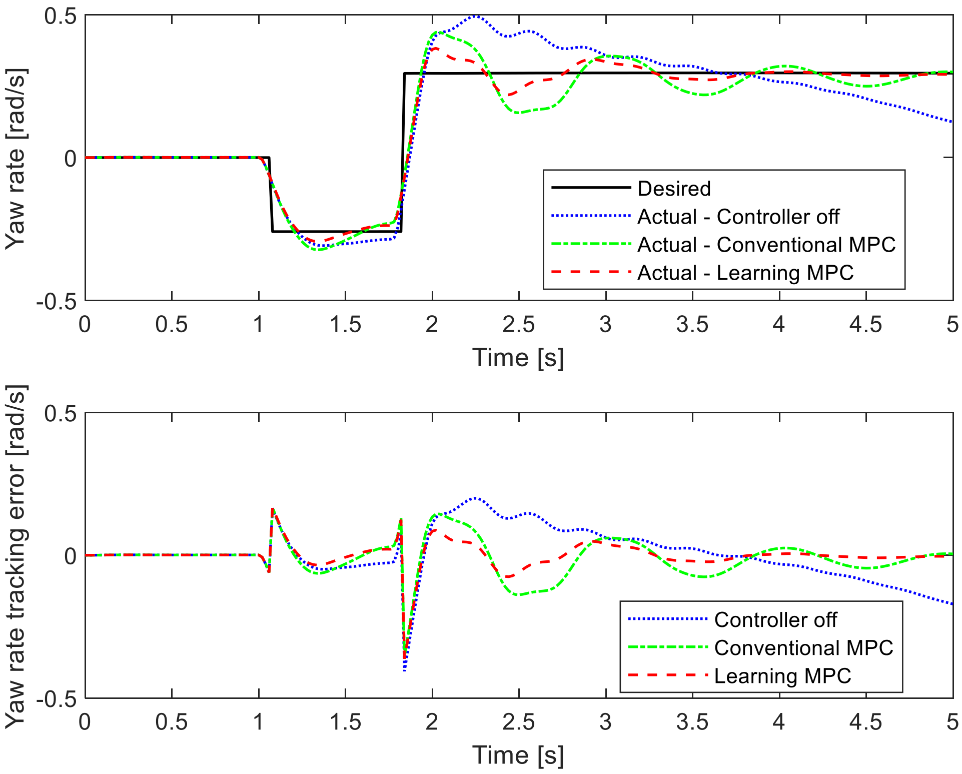


Figure 5.5: Yaw rate tracking in flick maneuver on wet surface.

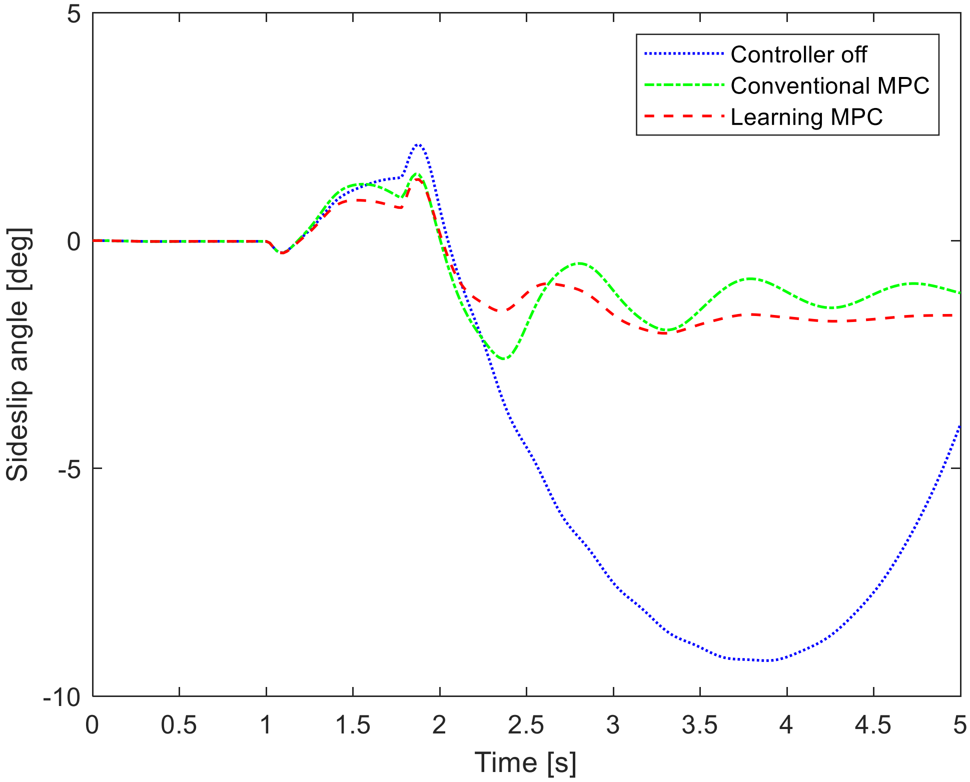
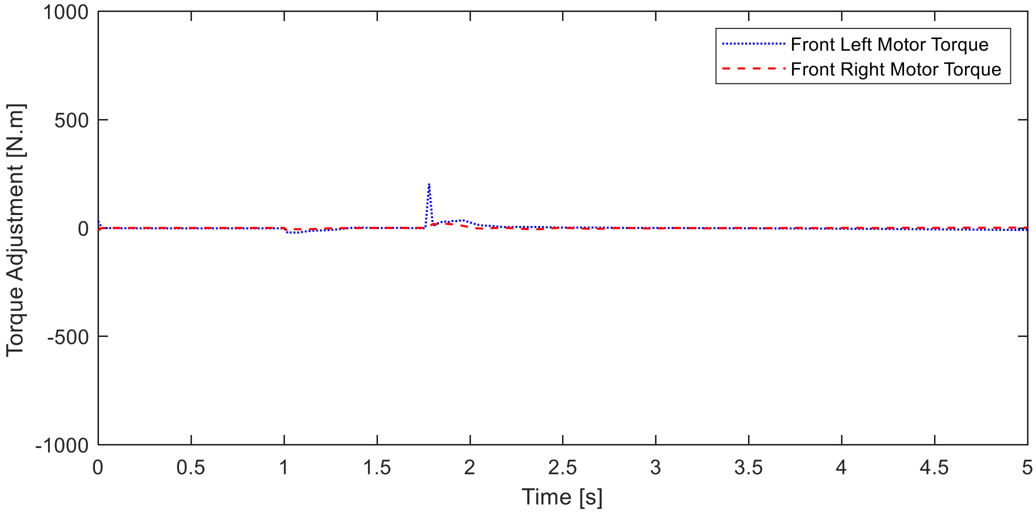
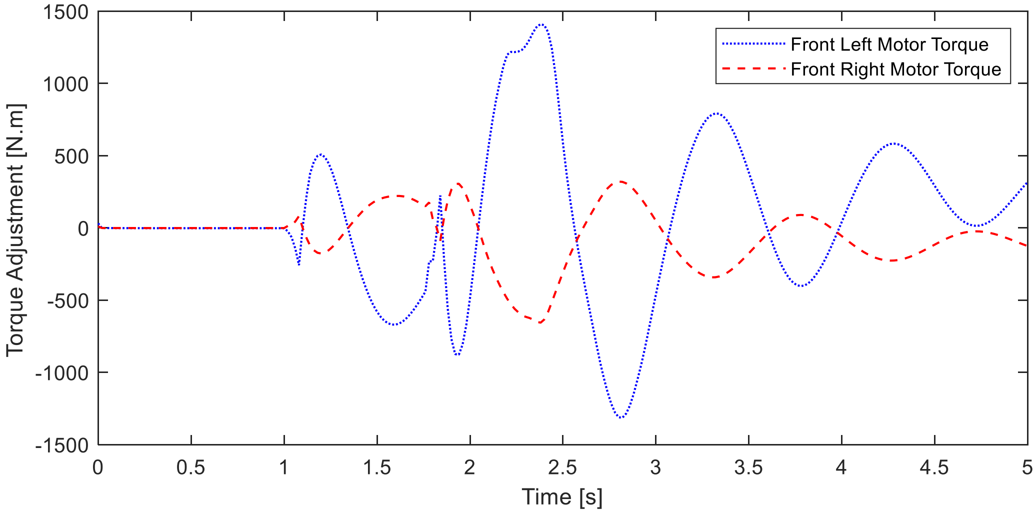


Figure 5.6: Vehicle sideslip in flick maneuver on wet surface.

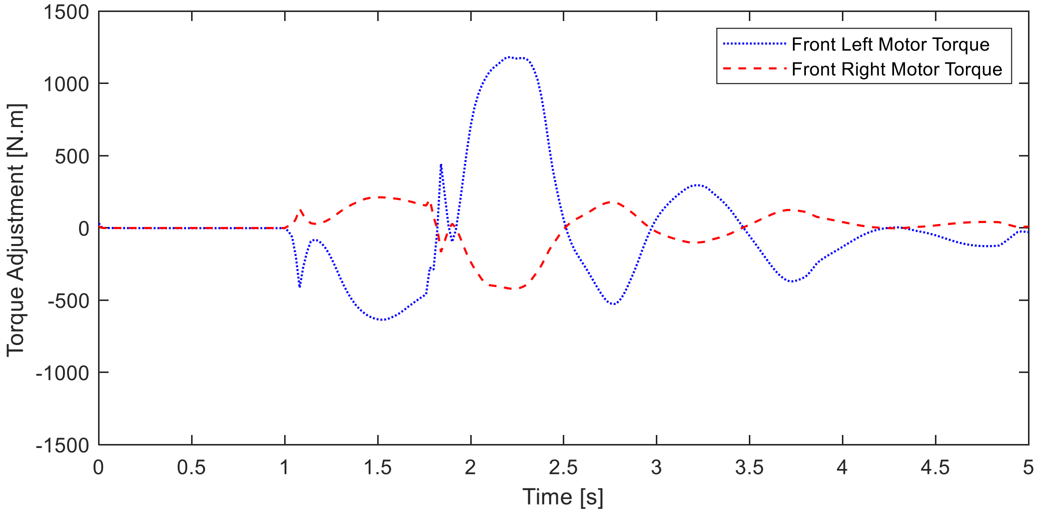
The yaw rate responses and the vehicle sideslip angle during a flick maneuver are shown in Figure 5.5 and Figure 5.6, respectively. The first observation from the figure is that in the controller-off mode, the yaw rate tracking error was relatively large. This suggests that without any control intervention, the vehicle's yaw rate deviated significantly from the desired value during the maneuver. Additionally, the vehicle sideslip response indicates that the vehicle was unstable in this case, as the sideslip angle (the angle between the vehicle's velocity vector and its longitudinal axis) was likely too large. In contrast, both the conventional MPC and the hybrid learning MPC demonstrated desirable yaw rate tracking responses. They were able to closely track the desired yaw rate signal during the flick maneuver. Furthermore, the vehicle sideslip response shows that the vehicle was stable, with the sideslip angle remaining below 5 degrees. This indicates that the control systems effectively stabilized the vehicle's motion during the maneuver. When comparing the conventional MPC with the hybrid learning MPC, some significant differences were observed. The learning MPC controller resulted in a smaller yaw rate tracking error and a smaller vehicle sideslip angle compared to the conventional MPC. This suggests that the hybrid learning MPC approach outperformed the conventional MPC in terms of yaw rate tracking and stability during the flick maneuver. Moreover, the hybrid learning MPC also showed a smaller settle time to track the desired yaw rate when the driver's steering angle remained constant after 2 seconds. Settle time refers to the time it takes for the system to reach and maintain the desired yaw rate after a change in the input signal. The shorter settle time in the hybrid learning MPC indicates a faster and more responsive control system, leading to improved tracking performance.



(a) Torque adjustment with controller off



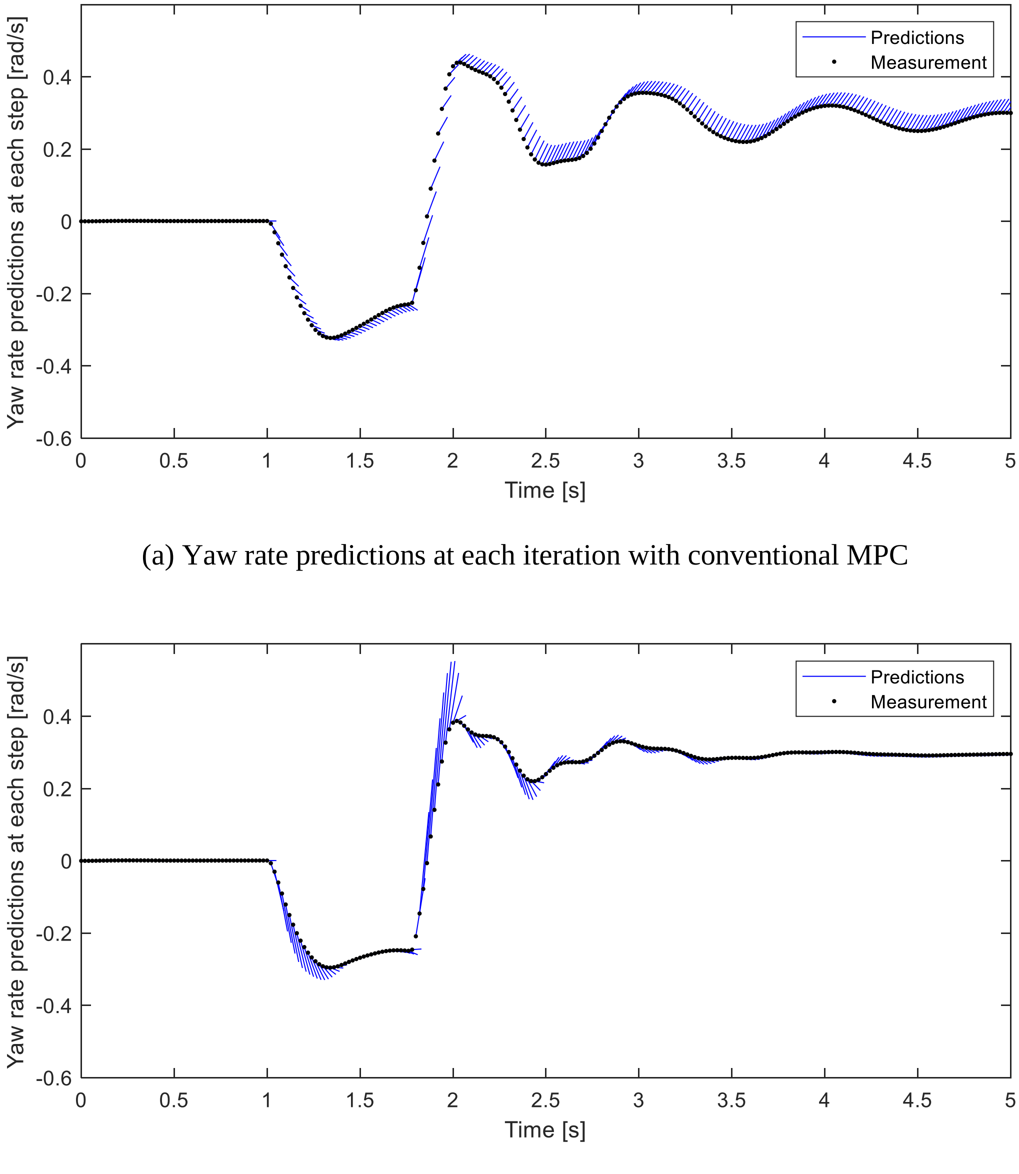
(b) Torque adjustment with conventional MPC



(c) Torque adjustment with learning MPC

Figure 5.7: Torque adjustment in flick maneuver on wet surface.

In this study, a front torque vectoring (TV) system was employed as the actuation system. The TV system allows for independent adjustments of torque to the front left and front right wheels of the vehicle. Figure 5.7 illustrates the torque adjustments applied to these wheels under different control modes. In Figure 5.7(a), it is evident that there are no torque adjustments made when the controller is in the off mode. This suggests that the front torque vectoring system remains inactive, and no torque distribution modifications occur during this mode. In Figure 5.7(b), the torque adjustments using a conventional MPC strategy are presented. The results indicate how the conventional MPC makes torque adjustments to the front wheels based on control inputs and the vehicle's state to achieve the desired yaw rate tracking during the maneuver. It is noticeable that the torque adjustments produced by the conventional MPC are relatively larger in magnitude, resulting in higher yaw moments. This implies that the conventional MPC tends to apply more significant interventions in response to the vehicle's dynamic behavior. On the other hand, Figure 5.7(c) illustrates the torque adjustments generated by the hybrid learning MPC approach. Comparing the results from the hybrid learning MPC with the conventional MPC, it becomes evident that the torque adjustments produced by the learning-based controller are smaller in magnitude, resulting in reduced yaw moments. This indicates that the hybrid learning MPC exerts less unnecessary intervention in response to the driver's input and the vehicle's motion, leading to more efficient energy consumption. Moreover, the results from the hybrid learning MPC reveal a more proactive control behavior compared to the conventional MPC. The torque adjustments peak with a faster response to the vehicle's yaw rate tracking error. This suggests that the learning-based controller is capable of promptly detecting deviations from the desired yaw rate and applying appropriate torque adjustments to correct the deviation in a more timely manner.



(a) Yaw rate predictions at each iteration with conventional MPC

(b) Yaw rate predictions at each iteration with learning MPC

Figure 5.8: Yaw rate predictions at each interaction in flick maneuver on wet surface.

As discussed in previous sections, a notable prediction error arises due to inaccuracies in the mathematical model or the linearization of the model within the MPC control scheme. While MPC employs feedback to iteratively update predictions in the predictive horizon to mitigate cumulative errors, an inherent predictive error still persists in relation to the actual response. Reducing modeling errors through learning techniques is a significant step towards enhancing control performance. To address this, GPR is implemented to estimate the model predictive error of yaw rate, that is, the difference between the real and the predictive yaw rate of each computation iteration of MPC. Figure 5.8 shows the predictions of yaw rate at each computational iteration and the real yaw rate feedback. It can be observed that the model in the conventional MPC tends to overpredict the dynamics of yaw rate as it uses the default setting of the road friction efficiency of 1.0, while the real road friction efficiency is 0.5. This mismatch of the road condition parameters leads to the prediction error of the conventional physics-based vehicle model, resulting in degraded control performance of the MPC. Furthermore, the figure demonstrates that the prediction error in the learning MPC is comparatively smaller than that in the conventional MPC. This is attributed to the introduction of a learning term into the physics-based vehicle model to counteract prediction errors. As the learning model accumulates more data during simulation, the prediction error is continually mitigated. Interestingly, under the learning MPC control mode, a notable deviation arises between the predicted yaw rates and the measured yaw rate when the driver's steering angle undergoes significant changes, such as around the 2-second mark. This deviation arises because the GPR utilized in this study exclusively learns from vehicle states and doesn't take into account the driver's intentions or inputs.

5.3 Experiment Vehicle and Test Track

The test vehicle utilized for this study is a Chevrolet Equinox electric vehicle from the Mechatronics Vehicle System Lab (MVSL) at the University of Waterloo. The vehicle is depicted in Figure 5.9, and its detailed specifications are listed in Table 5.2. This particular vehicle is designed as an all-wheel drive (AWD) model, featuring four electric motors, one mounted on each corner of the vehicle. The focus of this study involves torque vectoring on the front axle, while propulsion torque is applied through the two motors located on the rear axle.

An essential data collection setup is integrated into the vehicle. A 6-axis IMU sensor captures crucial information regarding the vehicle's acceleration, orientation, and angular rates. Additionally, wheel speed and torque sensors record rotational speed and force exertion for each individual wheel. To measure the vehicle's longitudinal and lateral velocities accurately, a high-definition Global Navigation Satellite System (GNSS) is utilized. The developed learning MPC is implemented using MATLAB. The controller operates on a laptop computer through the MATLAB/Simulink desktop real-time environment and establishes communication with other vehicle controllers via a Controller Area Network (CAN) bus. The experimental setup is shown as Figure 5.10. To ensure precise and replicable steering maneuvers for the purpose of comparing the performance of various control modes, a robotic steering system with a predefined steering profile as input was employed. The longitudinal speed of the vehicle is managed by a skilled technician from the MVSL. To simulate low-traction road conditions, a wet sealer is applied, shown in Figure 5.11.

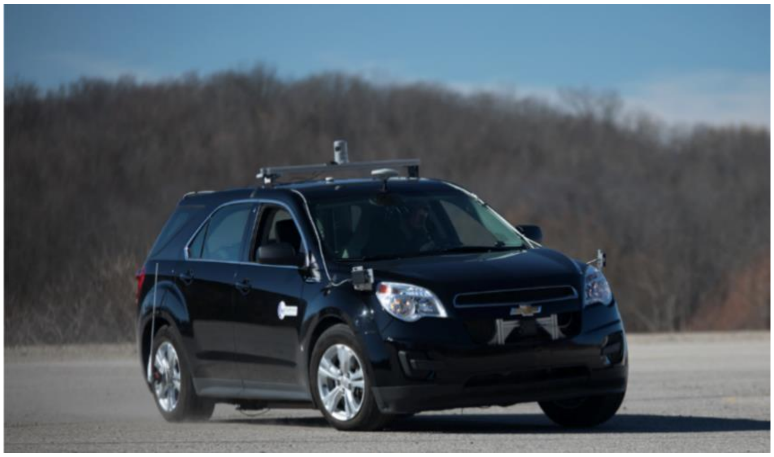


Figure 5.9: Test vehicle used in the experimental verification.

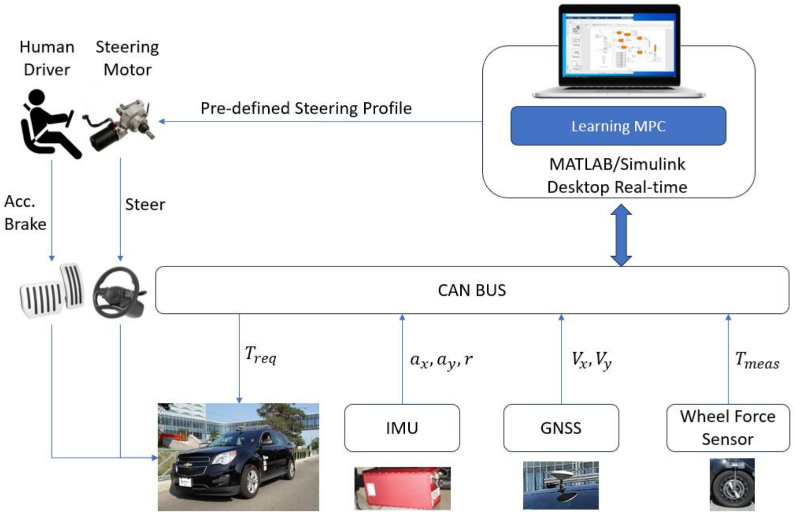


Figure 5.10: Experimental setup diagram of the test vehicle

Table 5.2: Experimental vehicle specification

Parameter	Unit	Value	Description
m	[kg]	2270	Vehicle mass
L_{wb}	[m]	2085	Wheelbase
l_f	[m]	1.41	CG distance to front axle
l_r	[m]	1.44	CG distance to rear axle
H_{CG}	[m]	0.72	CG hight
R_e	[m]	0.35	Effective radius of the tires
$C_{\alpha f}$	[N/rad]	130000	Front tires cornering stiffness
$C_{\alpha r}$	[N/rad]	130000	Rear tires cornering stiffness
l_s	[m]	1.63	Front and rear track width
I_z	[kg.m^2]	4600	Vehicle yaw moment of inertia

Figure 5.11: Test track used in the experimental verification.

5.4 Double Lane Change Maneuver Experiment

In this experimental study, the controller is being tested using a DLC input. The test is conducted at a constant vehicle speed of 50 km/h on a wet surface. The purpose of the test is to compare and verify the control performance when using different types of control modes: without the MPC controller, with a conventional MPC controller, and with the developed hybrid learning MPC controller. The driver steering input and the vehicle speed are shown in Figure 5.12. It can be noticed that this is a harsh driving maneuver with an approximate amplitude of 200 degrees of steering wheel angle on a wet road, which is to push the vehicle dynamics to the limit in order to evaluate the performance of the learning MPC. Despite small discrepancies, the controlled and uncontrolled maneuvers are conducted in the same way.

Figure 5.13 presents the yaw rate tracking performance of the testing vehicle across the three different control modes. The comparison is illustrated using two lines on each graph: a dotted red line represents the desired yaw rate, while a solid blue line represents the actual yaw rate achieved by the vehicle. In the initial scenario shown in Figure 5.13(a), referred to as the uncontrolled maneuver, the vehicle's ability to track the desired yaw rate is noticeably inadequate. This deficiency is highlighted by the significant maximum yaw rate tracking error of 0.8 radians per second (rad/s). The disparity between the desired and actual yaw rates implies that the vehicle struggles to maintain the intended turning behavior. Upon introducing a conventional MPC controller into the system, as depicted in Figure 5.13(b), the yaw rate tracking performance undergoes an improvement. The conventional MPC controller aids the vehicle in better aligning its yaw rate with the desired value. Consequently, the yaw rate tracking error diminishes to approximately 0.5 rad/s. This reduction signifies an enhancement in the vehicle's ability to follow the intended yaw rate trajectory, thereby resulting in smoother and more accurate turning motions. Furthermore, in Figure 5.13(c), the utilization of the hybrid learning MPC takes the yaw rate tracking performance a step further. With the learning MPC strategy in action, the yaw rate tracking error experiences a further reduction, now limited to a maximum of 0.3 rad/s. This outcome showcases the effectiveness of incorporating machine learning techniques into the control framework.

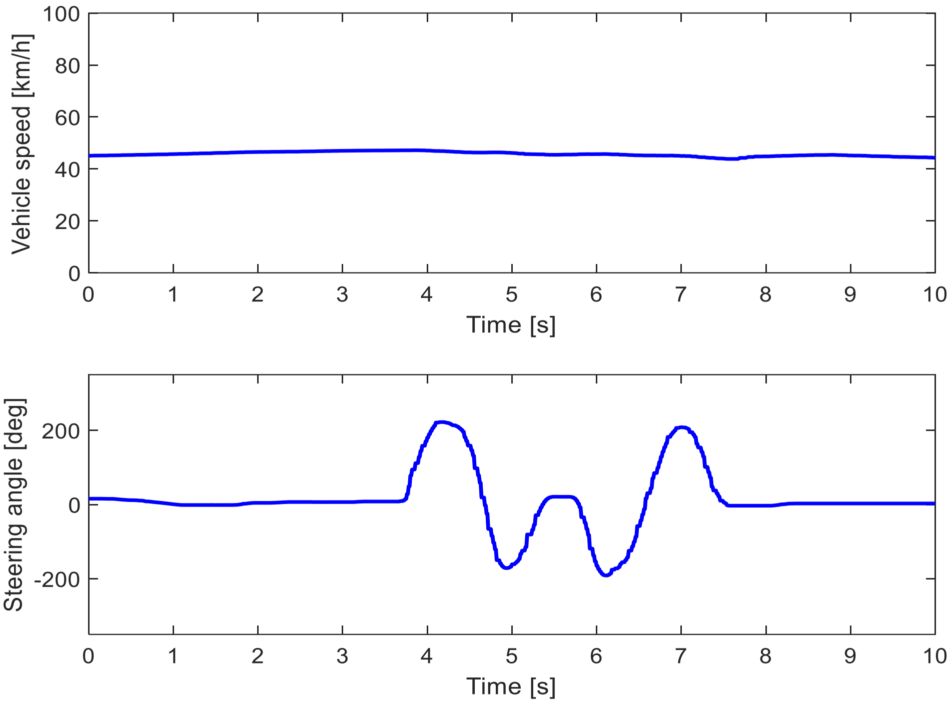
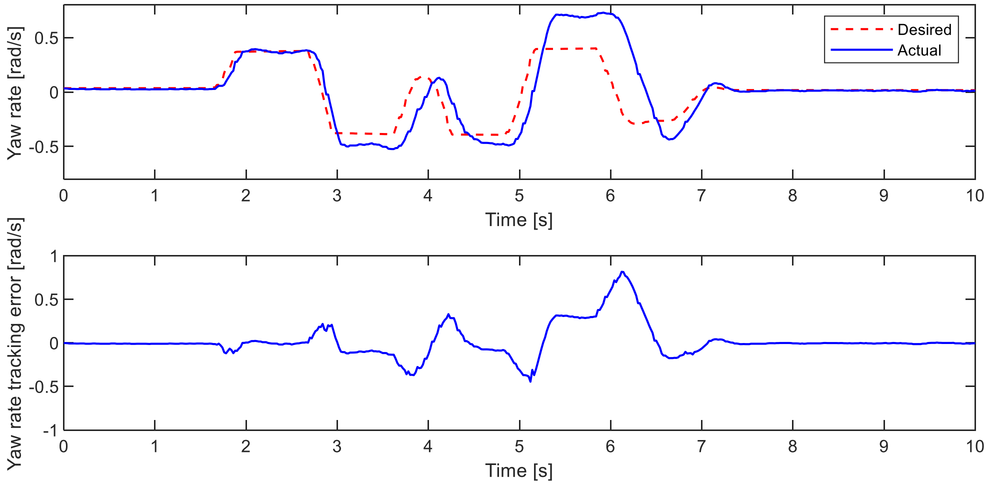
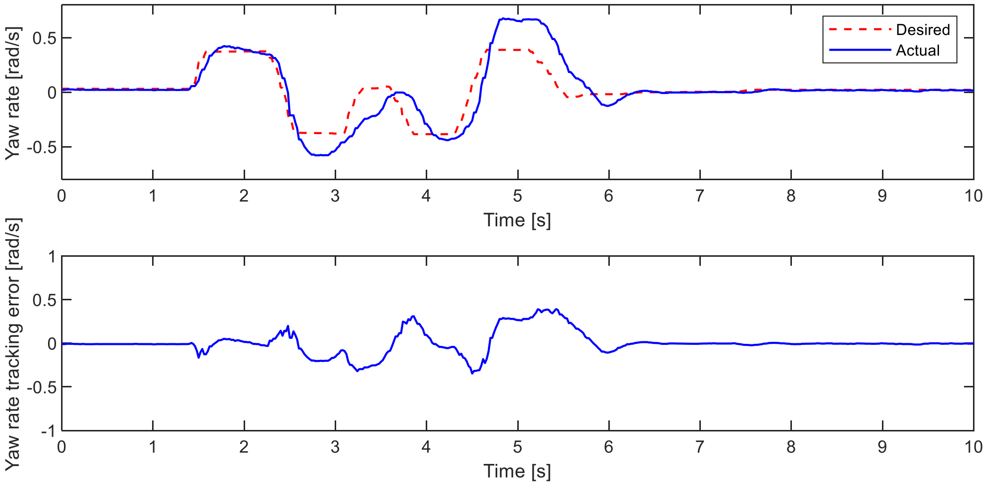


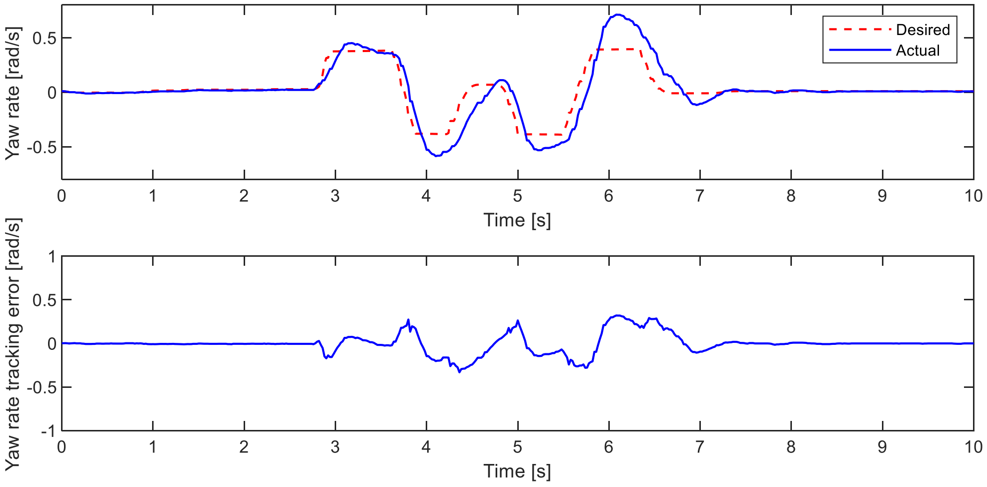
Figure 5.12: Vehicle speed and steering wheel angle during the DLC maneuvers.



(a) Uncontrolled



(b) Conventional MPC

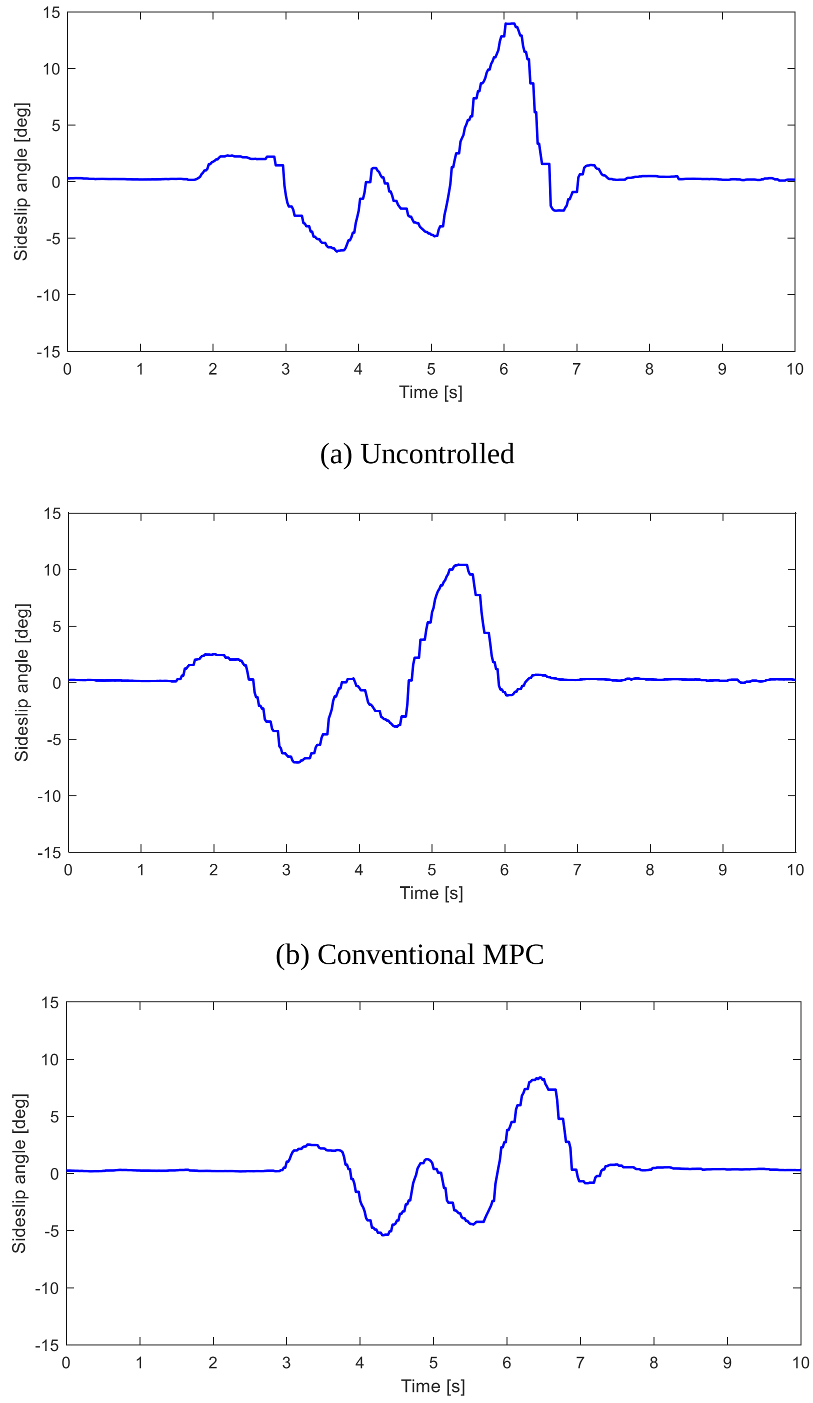


(c) Learning MPC

Figure 5.13: Comparative analysis of yaw rate tracking during the DLC maneuvers: (a) uncontrolled (b) conventional MPC (c) learning MPC.

Figure 5.14 presents a visual representation of the vehicle sideslip angle under different control modes, highlighting how these modes affect the vehicle's stability. In the uncontrolled maneuver, a sideslip angle of around 15 degrees is observed, as shown in Figure 5.14(a), which suggests that the vehicle is experiencing significant sliding or skidding. In Figure 5.14(b), the introduction of a conventional MPC reduces the maximum vehicle sideslip angle to 12 degrees. This reduction in sideslip angle indicates improved stability compared to the initial scenario without MPC, but there is still room for enhancement. In Figure 5.14(c), the hybrid learning MPC has been applied, resulting in a further reduction in the vehicle sideslip angle to below 8 degrees. This improvement signifies a higher level of stability and control compared to both the initial unstable state and the conventional MPC case. The alignment between the improved sideslip angle and yaw rate tracking performance suggests that the learning MPC not only reduces the sideslip angle but also enhances the vehicle's ability to follow the desired yaw rate, resulting in smoother and more controlled motions. The torque adjustments of the three control modes in this scenario are shown in Figure 5.15. Compared with the mode without the MPC and the mode with the MPC, it can be seen that the MPC controller generates yaw moment to reduce the overshot of yaw rate tracking. In the learning MPC mode, the controller responds more proactively to deviation of the actual yaw rate from the desired one. A similar trend to the previous driving scenario is seen in this simulation case study as well. As discussed, this can be interpreted as a more accurate yaw rate prediction due to the learning method used.

In the given scenario, Figure 5.15 presents how torque adjustments are utilized by various control modes to manage the vehicle's yaw rate and enhance stability. Figure 5.15(a) depicts the scenario where no specific control mechanism is actively adjusting the torque applied to the wheels. In this case, the vehicle's behavior is likely to be erratic and potentially unsafe, as there is no regulation of yaw rate or stability through torque interventions. The visualization in Figure 5.15(b) illustrates the torque adjustments made by the conventional MPC in response to the vehicle's behavior. The primary goal of these adjustments is to minimize the overshooting of the yaw rate tracking. The torque interventions generated by the conventional MPC are aimed at curbing any excessive yaw rate deviations. Figure 5.15(c) presents the torque adjustments performed by the learning MPC. Compared to the conventional MPC, the learning MPC is observed to respond more proactively to deviations between the actual and desired yaw rates. This responsiveness suggests that the learning MPC is able to predict and anticipate yaw rate deviations more accurately, likely due to its learning mechanism. The description highlights that the torque adjustments made by the learning MPC are not only more proactive but also less pronounced in magnitude compared to those of the conventional MPC. This indicates that the learning MPC is able to fine-tune its torque interventions more precisely. The decreased magnitude of torque interventions suggests that the learning MPC is eliminating unnecessary and potentially destabilizing interventions that the conventional MPC might have applied. This efficiency in torque application can contribute to smoother vehicle behavior and improved stability during maneuvers.

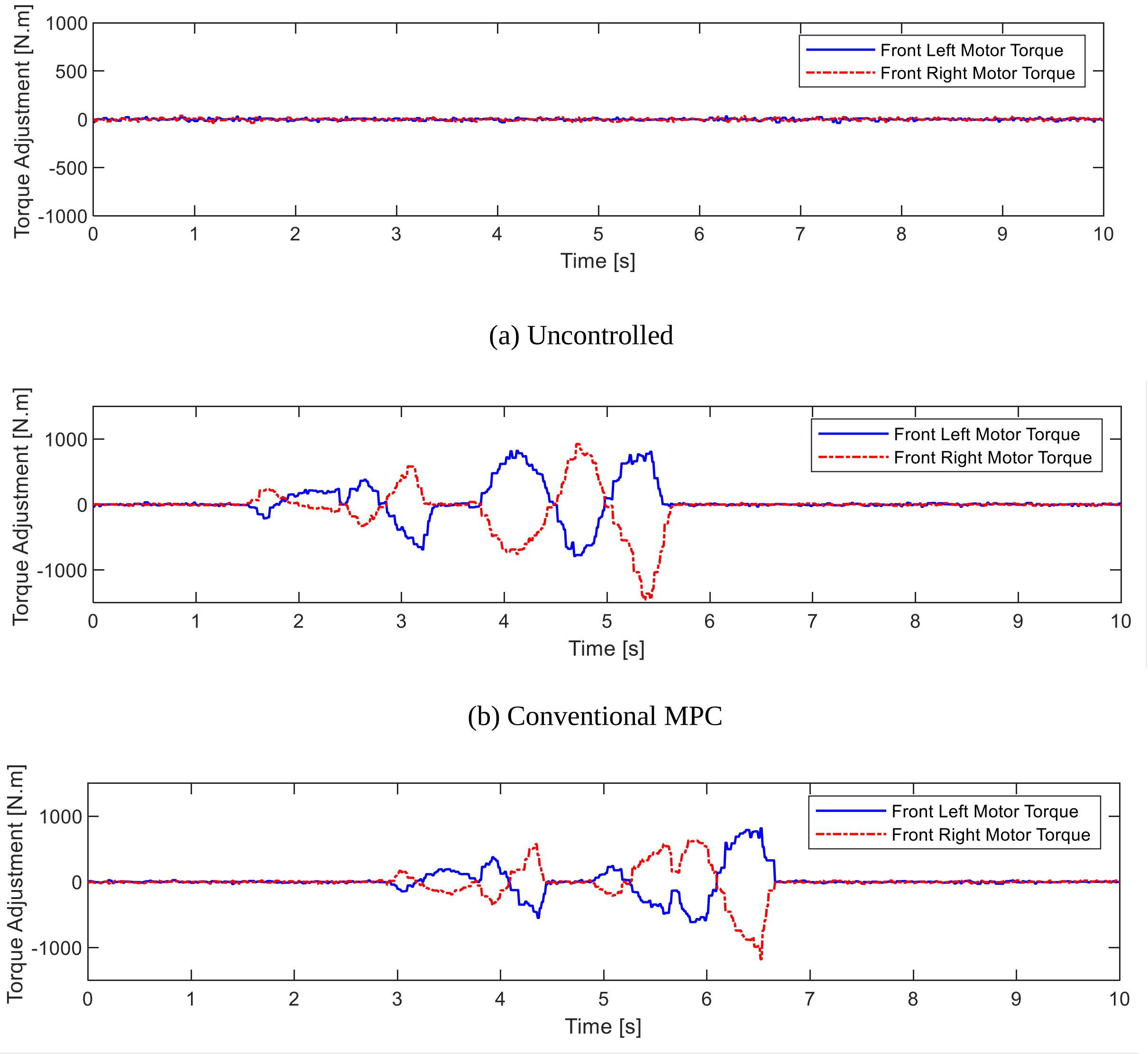


(a) Uncontrolled

(b) Conventional MPC

(c) Learning MPC

Figure 5.14: Comparative analysis of vehicle sideslip angle during the DLC maneuvers: (a) uncontrolled (b) conventional MPC (c) learning MPC.



(a) Uncontrolled

(b) Conventional MPC

(c) Learning MPC

Figure 5.15: Comparative analysis of torque adjustment during the DLC maneuvers: (a) uncontrolled (b) conventional MPC (c) learning MPC.

5.5 Repeated Double Lane Change Maneuver

In this experiment, the performance of a developed learning MPC controller is evaluated through three repeated DLC maneuvers on a wet surface. The vehicle's speed and steering wheel angle are depicted in Figure 5.16, with DLC maneuvers executed at intervals of 5-10 seconds, 41-46 seconds, and 65-70 seconds at a constant speed of 50 km/h. Despite the need for a U-turn due to track

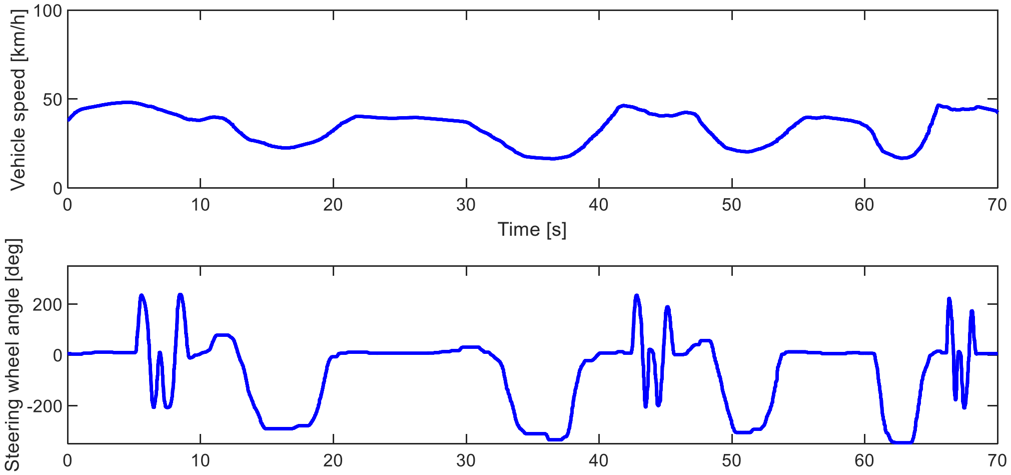


Figure 5.16: Vehicle speed and steering wheel angle in the repeated DLC maneuver.

limitations, the DLC maneuvers are consistently conducted in the same manner and conditions, allowing assessment of the learning MPC controller's real-time learning capabilities for improved control during DLC maneuvers. Throughout the test, a continuous stream of data is collected to capture the vehicle's dynamic behavior and responses during the DLC maneuvers. This data is then utilized as input for GPR prediction.

Figure 5.17 illustrates the yaw rate tracking performance of the vehicle under the influence of the learning MPC during repeated DLC maneuvers. A closer examination of this figure reveals the dynamic changes in yaw rate tracking errors across the three DLC maneuvers and highlights the impact of the developed learning MPC control approach. As shown in Figure 5.17, the yaw rate tracking error progressively diminishes over the course of the repeated DLC maneuvers. Specifically, the tracking error is observed to be reduced in the third DLC maneuver when compared to the first and second DLC maneuvers. This decreasing trend in yaw rate tracking error signifies the online learning effect of the learning MPC control strategy. The learning MPC system learns from the vehicle's responses and interactions during the earlier maneuvers, enabling it to make more informed and accurate control decisions in subsequent maneuvers. This adaptation is evidenced by the progressively improved yaw rate tracking performance, reflecting the learning MPC's capacity to refine its control strategy over time. Additionally, Figure 5.18 complements this narrative by presenting the vehicle's sideslip angle, which quantifies the degree of lateral sliding or skidding during the DLC maneuvers. The figure showcases a consistent and continuous enhancement in vehicle stability through successive DLC maneuvers. In Figure 5.18, the vehicle sideslip angle is observed to undergo a sequence of reductions across the three DLC maneuvers. Specifically, the maximum sideslip angle diminishes from 7.5 degrees in the first DLC maneuver to 6.5 degrees in the second, and further down to 4 degrees in the third DLC maneuver when utilizing the developed learning MPC. The reduction in sideslip angle aligns with the reduced yaw rate tracking error observed in Figure 5.17. This trend underscores the learning MPC's ability to progressively minimize the extent of lateral motion and improve vehicle stability over time.

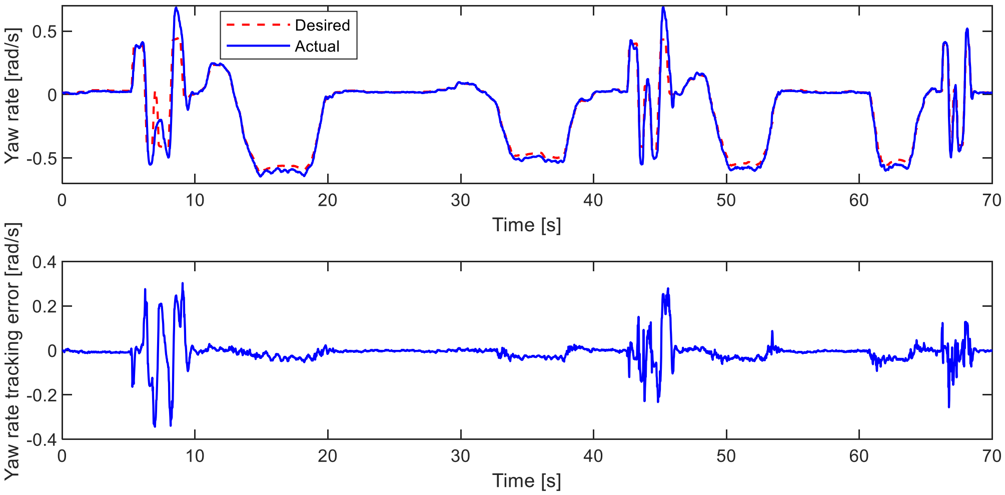


Figure 5.17: Yaw rate tracking in the repeated DLC maneuver.

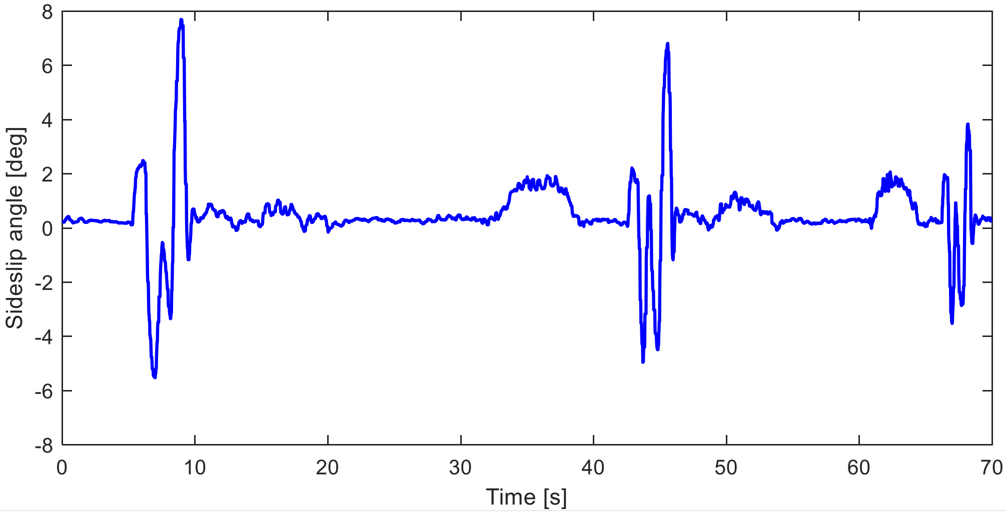
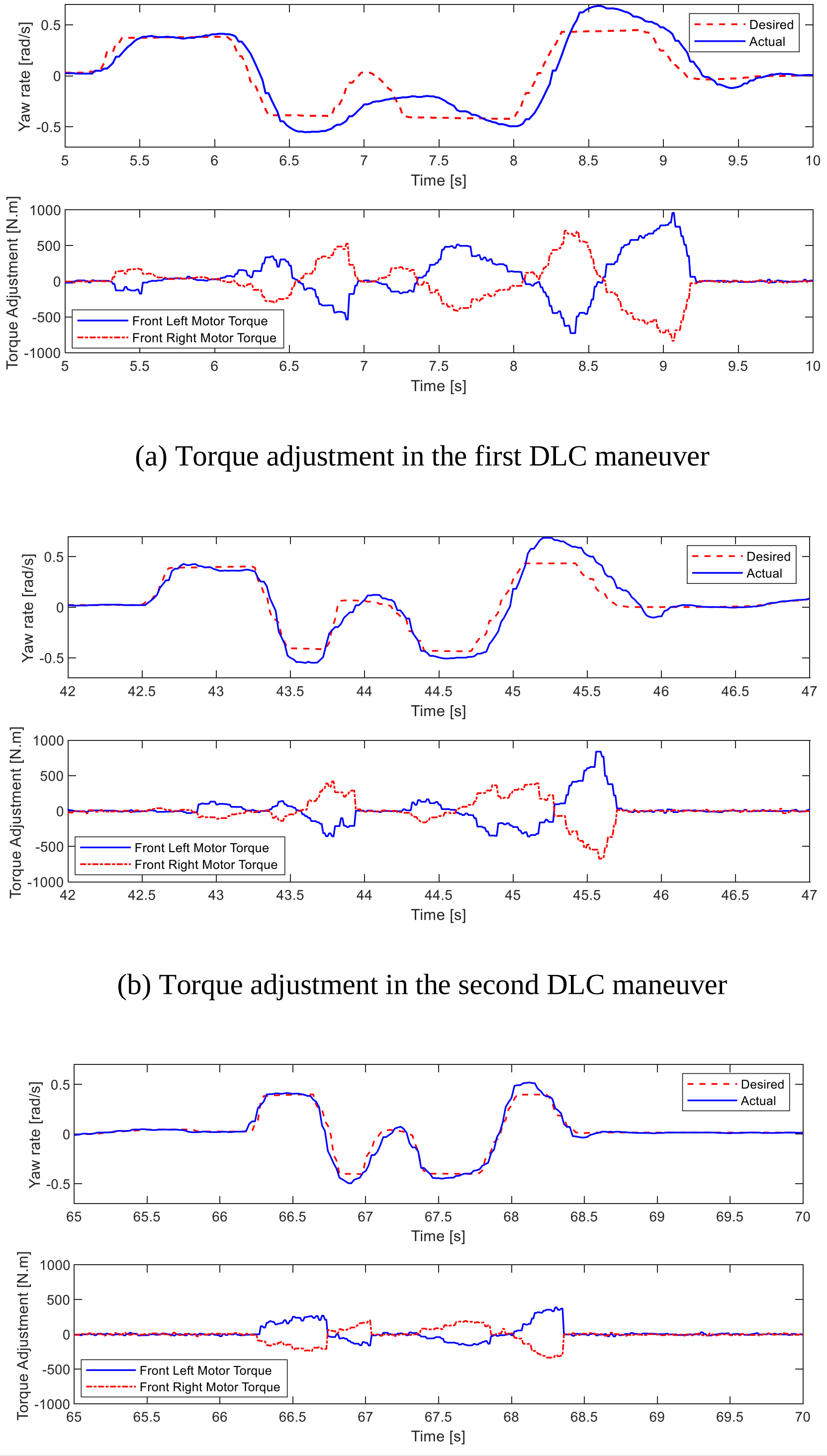


Figure 5.18: Vehicle sideslip angle in the repeated DLC maneuver.



(a) Torque adjustment in the first DLC maneuver

(b) Torque adjustment in the second DLC maneuver

(c) Torque adjustment in the third DLC maneuver

Figure 5.19: Torque adjustment in the repeated DLC maneuver.

The torque adjustments of the three DLC maneuvers in this scenario are shown in Figure 5.19. Compared with the first DLC maneuver and the second DLC maneuver, it can be seen that the learning MPC controller generates more proactive yaw moments to reduce the yaw rate tracking error. This also results in smaller torque intervention. As discussed, this can be interpreted as a more accurate yaw rate prediction due to the learning method used.

In Figure 5.19, the torque adjustments applied during the three repeated DLC maneuvers are presented. Comparing the torque adjustments during the first and second DLC maneuvers with those in the third maneuver, a distinct pattern emerges. The learning MPC's torque adjustments during the third maneuver exhibit a more proactive nature, designed to mitigate yaw rate tracking errors more effectively. This responsiveness suggests that the learning MPC, having accumulated more data from the previous maneuvers, has refined its control strategy to preemptively counteract deviations in yaw rate. Furthermore, an intriguing observation pertains to the quantity and magnitude of torque interventions. In the third DLC maneuver, the number of torque interventions is notably reduced, and their magnitudes are smaller compared to the first and second maneuvers. This outcome aligns with the observation of more accurate yaw rate prediction, a result of the learning mechanism incorporated within the MPC. The torque adjustment patterns demonstrated in Figure 5.19 underscore the learning MPC's adaptability and its capacity to fine-tune control strategies with accumulating data.

5.6 Summary

The chapter presents the verification of the developed hybrid learning model predictive control (MPC) strategy to address the challenges in vehicle stability control. The proposed hybrid learning MPC combines the strengths of traditional model-based MPC with data-driven techniques, particularly Gaussian Process Regression (GPR). GPR is employed to enhance the prediction accuracy of the vehicle dynamics model by learning from historical data and real-time sensor measurements. To evaluate the efficacy of the proposed approach, comprehensive simulations are conducted using the CarSim-Simulink co-simulation environment, allowing for a detailed analysis of vehicle behavior under various driving maneuvers. Furthermore, experimental validation is carried out using the Equinox EV testing vehicle. The developed hybrid learning MPC with front torque vectoring is put to the test in a range of maneuvers, including double lane change (DLC), repeated DLC, and flick maneuver on dry and slippery road conditions. The results from both simulation and experimental studies demonstrate the superiority of the hybrid learning MPC over conventional MPC and uncontrolled mode. The system exhibits enhanced yaw rate tracking, better sideslip control, and improved overall vehicle stability during aggressive maneuvers. The integration of GPR significantly boosts the prediction accuracy of yaw rate, enabling more precise and proactive control actions through torque adjustment. By fusing model-based predictions with data-driven learning, the proposed control system achieves remarkable improvements in yaw rate tracking and vehicle sideslip control. The successful verification through both simulations and experimental tests validates its effectiveness and potential for real-world vehicle stability applications.

Chapter 6

Simulation and Experimental Verification of Adaptive MPC

In this chapter, an adaptive MPC is developed to optimize the performance of vehicle stability control. The developed adaptive MPC controller is implemented in Matlab and evaluated through a Matlab/Simulink CarSim co-simulation as well as real vehicle experiments. A high-fidelity CarSim model is used in the simulations and a Chevrolet Equinox electric testing vehicle is used for experimental evaluations.

6.1 Proposed Adaptive-Prediction-Horizon MPC

Integrating learning MPC to enhance model prediction accuracy is a crucial step in improving the controller's ability to make accurate predictions based on real-time data. However, even with improved predictive accuracy, the standard MPC's fixed prediction horizon might not always be optimal for all driving scenarios and conditions. This is where the adaptive prediction horizon comes into play. In certain situations, like high-speed driving or sudden changes in road conditions, a longer prediction horizon might be beneficial to anticipate and respond to potential challenges. In contrast, during low-speed urban driving or stable conditions, a shorter prediction horizon might suffice, enabling quicker and more agile control responses.

In this study, an adaptive-prediction-horizon MPC is proposed and implemented, shown as Figure 6.1. In this adaptive MPC, the prediction interval is adjusted in real time based on observed vehicle behaviors, while the prediction steps are kept the same, which can extend the prediction horizon while maintaining the computational cost at the same level. The proposed strategy to adjust the prediction interval for the adaptive MPC is as follows: In this strategy, the algorithm monitors the current yaw rate and compares it with the maximum allowable yaw rate with a tolerance. If the yaw rate exceeds the acceptable limit, the prediction interval is increased to provide the controller more time to respond. On the other hand, if the yaw rate is well within the limit, the prediction interval is decreased to focus on shorter-term control actions. The increase and decrease factors, as well as the maximum and minimum prediction intervals, can be adjusted based on the specific application.

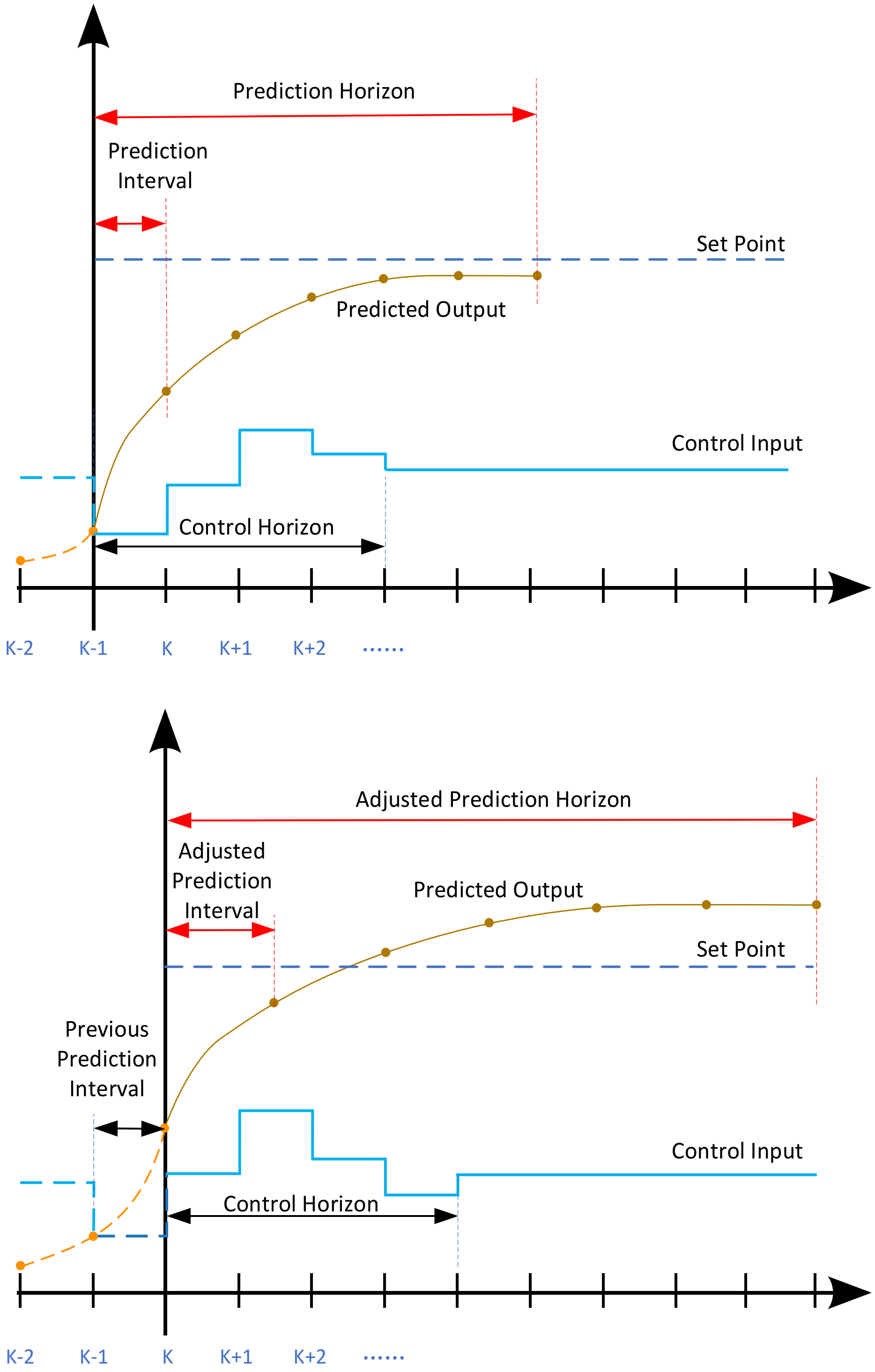
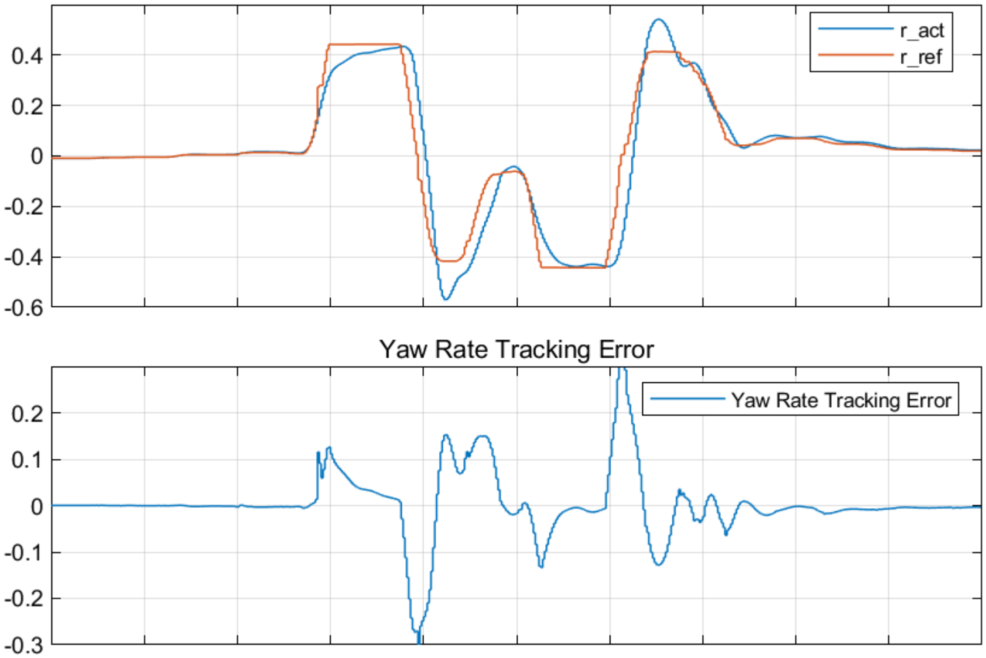


Figure 6.1: Proposed adaptive MPC concept diagram.

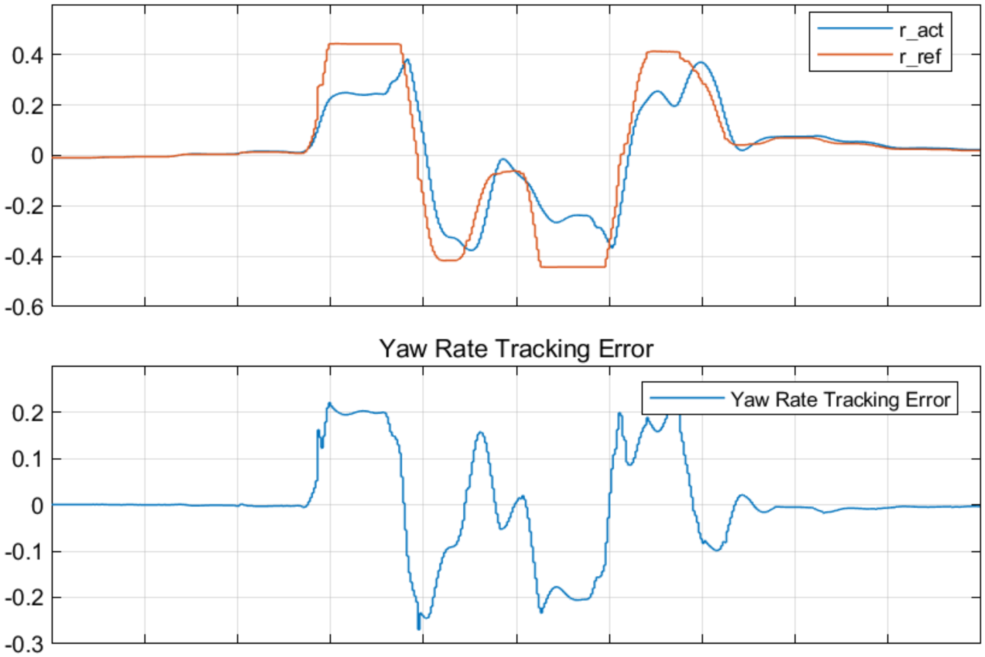
6.2 Double Lane Change Maneuver Simulation

The developed adaptive MPC was validated using a DLC maneuver within a Matlab/Simulink CarSim co-simulation. The test is conducted at a constant vehicle speed of 40 km/h on a wet surface with a friction coefficient of 0.4. The objective of this test is to compare and analyze the results obtained from employing various control modes: fixed-prediction-horizon MPC with a 10ms prediction interval, fixed-prediction-horizon MPC with a 100ms prediction interval, and adaptive-prediction-horizon MPC with a prediction interval that alternates between 10ms and 100ms based on the adaptive strategy. The graphical representation of the yaw rate response is depicted in Figure 6.2. When employing a relatively smaller prediction interval (10ms), the yaw rate closely follows the desired yaw rate, but this can lead to notable overshooting or oversteering effects, as shown in Figure 6.2(a). Conversely, when adopting a relatively larger prediction interval (100ms), the yaw rate exhibits more pronounced undershoot or understeering effects, evident in Figure 6.2(b). These observations highlight that a shorter prediction horizon may result in overshoot due to the limited reaction time available. However, it's important to note that an excessively long horizon might not always be advantageous, as it can cause the controller to react too conservatively. In contrast, the adaptive MPC, which dynamically adjusts the prediction interval from 10ms to 100ms as the yaw rate approaches, offers an improved yaw tracking performance with a smaller tracking error. This improvement is attributed to the longer prediction interval providing the controller with an extended horizon to anticipate approaching limits and more time to formulate a controlled response.

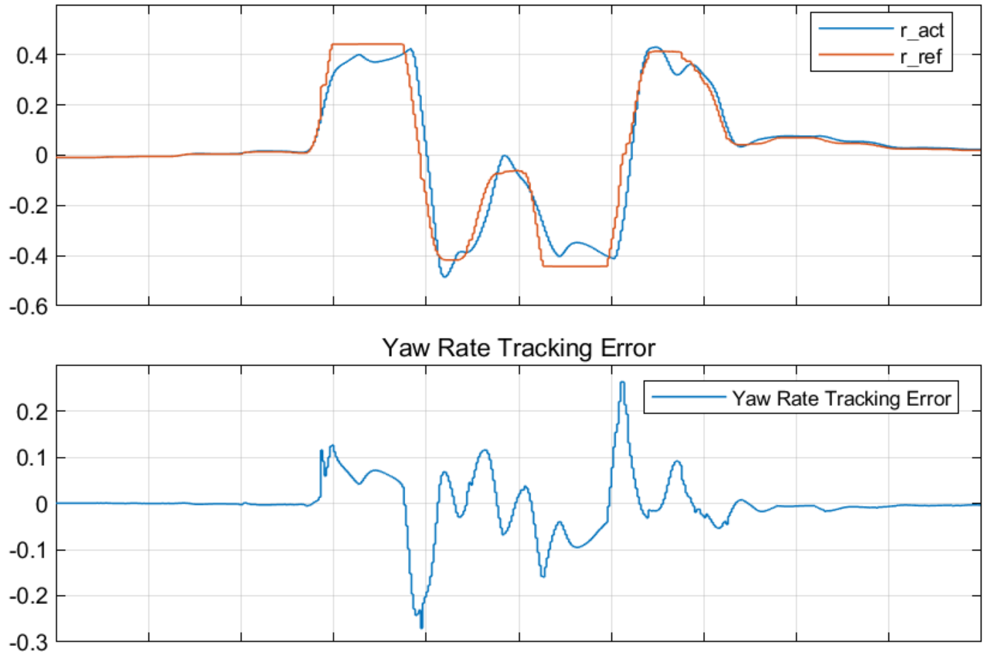
Figure 6.3 illustrates the sideslip angle of the vehicle. As depicted in Figure 6.3(a), the vehicle's sideslip angle is notably amplified (4 degrees) when utilizing a shorter prediction interval (10ms) within a fixed prediction horizon MPC. Conversely, a prolonged prediction interval of 100ms, showcased in Figure 6.3(b), yields a reduced sideslip angle of approximately 1 degree. However, this improvement in sideslip angle comes at the cost of notable understeering, preventing the vehicle from effectively tracking the desired yaw rate. In the context of the adaptive MPC, as displayed in Figure 6.3(c), the sideslip angle of the vehicle remains around 2 degrees, signifying a stable region. This outcome indicates that the developed adaptive MPC strikes a balance between the extremes of oversteering and understeering. By dynamically adjusting the prediction interval, the controller achieves a compromise that enhances both sideslip angle and yaw rate tracking performance, leading to a more stable and controlled vehicle behavior during this maneuver.



(a) Yaw rate tracking with fixed prediction interval (10ms) in MPC



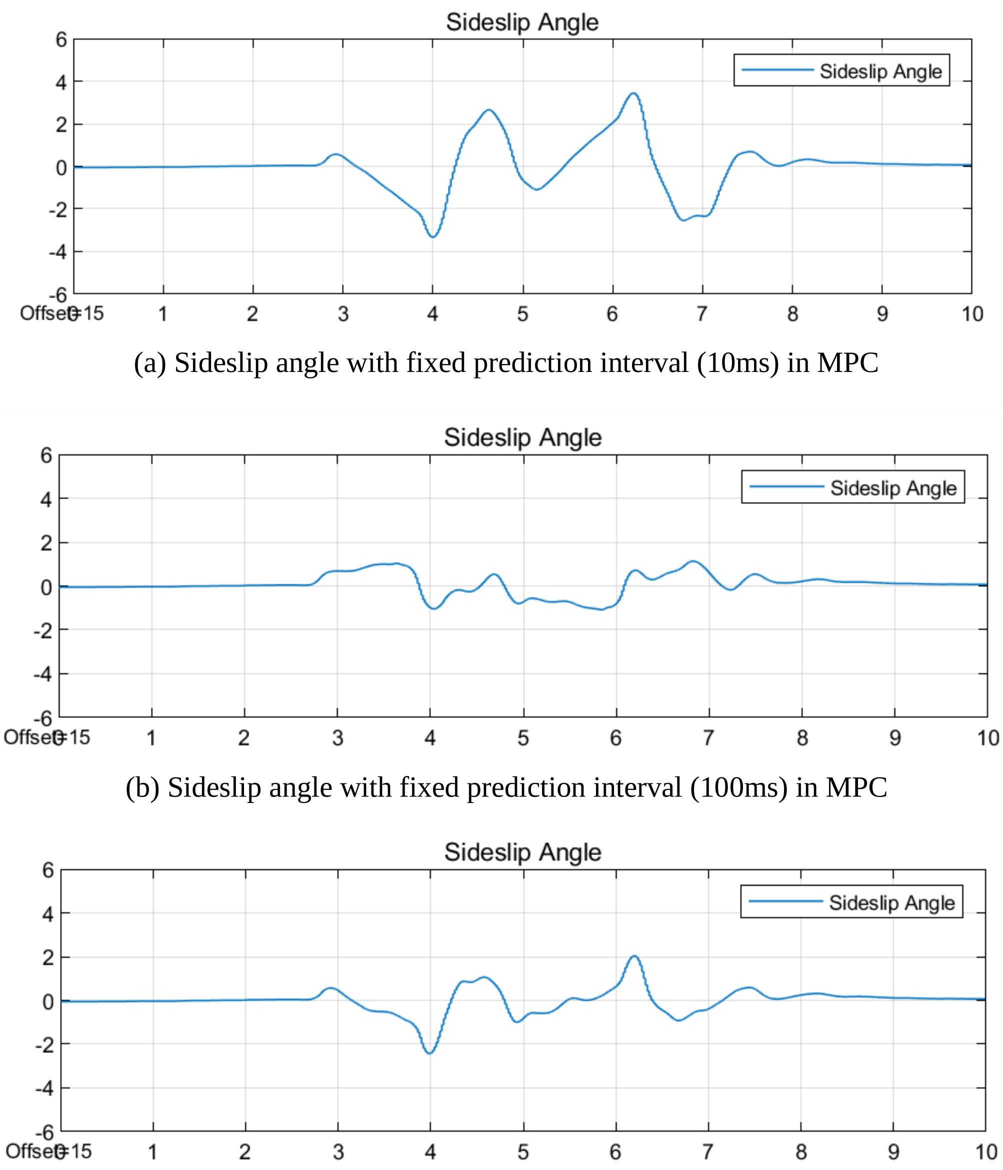
(b) Yaw rate tracking with fixed prediction interval (100ms) in MPC



(c) Yaw rate tracking with adaptive prediction interval in MPC

Figure 6.2: Comparative analysis of yaw rate tracking during the DLC maneuvers: (a) fixed prediction interval (10ms) (b) fixed prediction interval (100ms) (c) adaptive prediction interval.

Figure 6.4 provides an insight into the torque adjustments applied to each front wheel, highlighting the application of electric motors as actuators. The figure unveils a distinctive behavior: when a shorter prediction interval (10ms) is employed, the torque vectoring intervention occurs relatively late, leading to an overshot in yaw rate tracking. Conversely, a longer prediction interval (100ms) prompts the torque vectoring to react too early, resulting in an undershoot of the yaw rate tracking. Notably, the adaptive MPC controller responds to these challenges through an intelligent strategy. It adjusts the prediction interval dynamically, transitioning from 10ms to 100ms as the yaw rate nears its limit value. This adjustment facilitates optimal control: when the yaw rate strays from the limit value, the prediction interval swiftly reverts to 10ms, preventing premature reactions. This nuanced approach empowers the adaptive MPC controller to effectively address the complexities posed by the test scenario. By tailoring the prediction interval based on the vehicle's proximity to the yaw rate limit, the adaptive MPC controller strikes a balance between timely intervention and accurate yaw rate tracking, leading to enhanced overall performance during the maneuver.

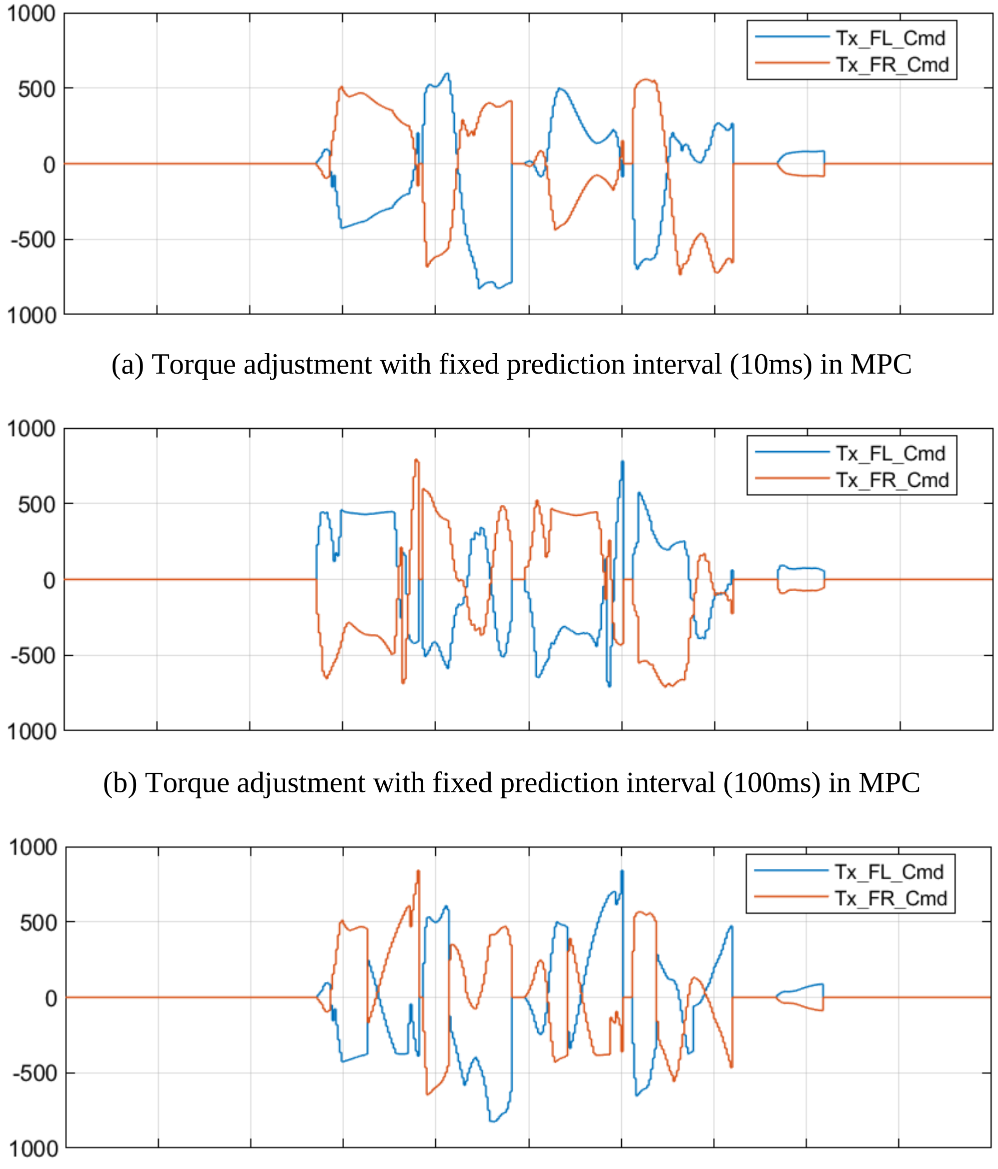


(a) Sideslip angle with fixed prediction interval (10ms) in MPC

(b) Sideslip angle with fixed prediction interval (100ms) in MPC

(c) Sideslip angle with adaptive prediction interval in MPC

Figure 6.3: Comparative analysis of vehicle sideslip angle during the DLC maneuvers: (a) fixed prediction interval (10ms) (b) fixed prediction interval (100ms) (c) adaptive prediction interval.



(a) Torque adjustment with fixed prediction interval (10ms) in MPC

(b) Torque adjustment with fixed prediction interval (100ms) in MPC

(c) Torque adjustment with adaptive prediction interval in MPC

Figure 6.4: Comparative analysis of torque adjustment during the DLC maneuvers: (a) fixed prediction interval (10ms) (b) fixed prediction interval (100ms) (c) adaptive prediction interval.

6.3 Slalom Maneuver Experiment

The effectiveness of the developed adaptive MPC was additionally confirmed through practical experimental tests conducted on the Equinox testing vehicle. In this specific scenario, the controller's performance was evaluated by subjecting it to a sine wave steering input with gradually increasing frequency. This assessment took place under consistent conditions, maintaining a steady vehicle

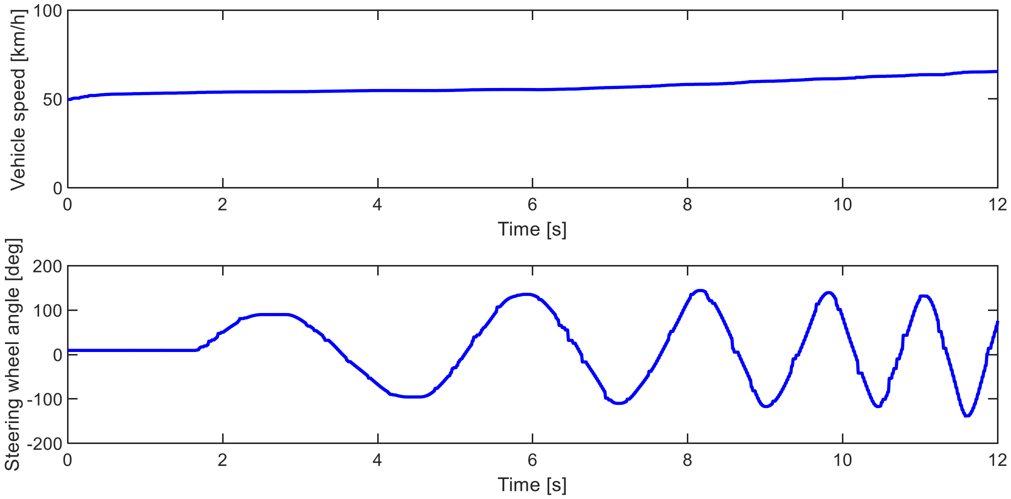
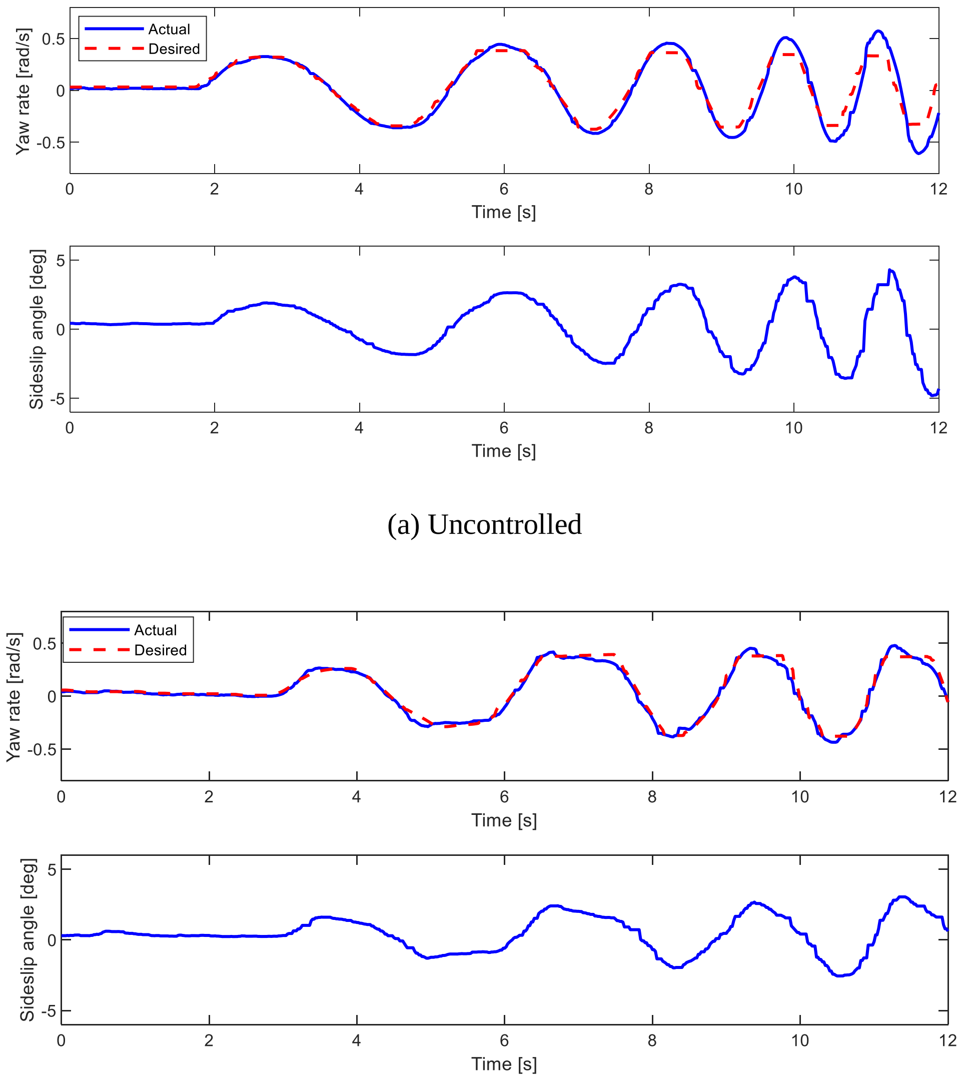


Figure 6.5: Vehicle speed and steering wheel angle in the slalom maneuver.

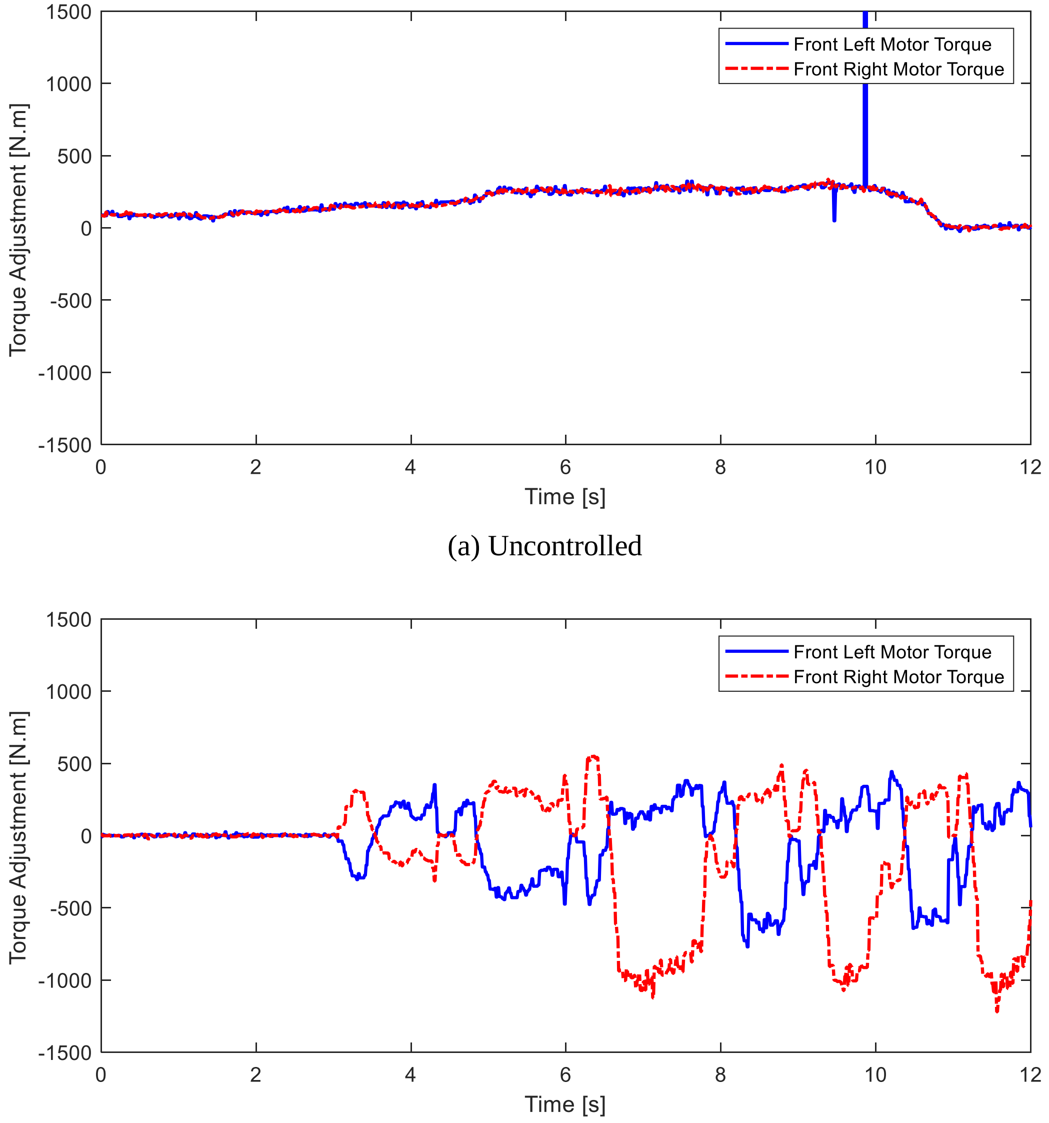
speed of approximately 60 km/h on a wet surface. The principal aim of this test was to conduct a comparative analysis between two scenarios: one involving the adaptive MPC controller and another without any MPC controller. The driver's steering input and vehicle speed are depicted in Figure 6.5. While slight variations might exist, both controlled and uncontrolled maneuvers were executed following similar patterns. Figure 6.6 offers a comprehensive comparison between the yaw rate tracking and sideslip angle of the vehicle in both controlled and uncontrolled maneuvers. In the absence of the controller, the vehicle's yaw rate exhibits a pronounced trend of surpassing limit values as the steering frequency escalates. This phenomenon has the potential to induce oversteering, a situation where the vehicle's rear end tends to lose traction and may lead to instability. Conversely, when the adaptive MPC controller is operational, the vehicle showcases a remarkable capacity to precisely follow the desired yaw rate without notable instances of overshooting or undershooting. This controlled response implies that the adaptive MPC effectively regulates the vehicle's yaw rate within acceptable limits, mitigating the risk of oversteering and promoting stability. The observation extends to the sideslip angle, which is equally illustrative of the controller's impact. The adaptive MPC's involvement ensures that the sideslip angle remains controlled and within acceptable ranges, contributing to overall vehicle stability and maintaining desired trajectory. Figure 6.7 shows the torque adjustments orchestrated by the adaptive MPC controller between controlled and uncontrolled maneuvers. In the uncontrolled maneuver, an evident motor torque noise pattern emerges. Conversely, within the adaptive MPC control mode, as the steering frequency escalates, the adaptive MPC controller strategically orchestrates increased torque adjustments. This particular response serves a crucial purpose: by demanding greater torque adjustments, the controller effectively counteracts the potential for yaw rate overshooting. This is a notable achievement as the overshoot of yaw rate, which can lead to instability, is substantially minimized under the control of the adaptive MPC.



(a) Uncontrolled

(b) Adaptive MPC

Figure 6.6: Comparative analysis of yaw rate tracking and vehicle sideslip angle during the slalom maneuvers: (a) uncontrolled (b) adaptive MPC.



(a) Uncontrolled

(b) Adaptive MPC

Figure 6.7: Comparative analysis of torque adjustment during the slalom maneuvers: (a) uncontrolled (b) adaptive MPC.

6.4 Double Lane Change Maneuver Experiment

In this experimental study, a controller is being tested using a double lane change input. The test is conducted at a constant vehicle speed of 60 km/h on a wet surface. The purpose of the test is to compare and verify the control performance when using different types of controllers: without the MPC controller, with a fixed-prediction-horizon MPC controller, and with an adaptive-prediction-horizon

MPC controller. The driver steering input and the vehicle speed are shown in Figure 6.8. Despite small discrepancies, the controlled and uncontrolled maneuvers are conducted in the same way. Figure 6.9 compares the yaw rate tracking of the vehicle in the controlled and uncontrolled maneuvers. It can be seen in the uncontrolled maneuver; the vehicle yaw rate exceeds the limit values with the maximum tracking error of 0.2 rad/s. With the conventional MPC, the yaw rate tracking error is reduced with the maximum tracking error of 0.1 rad/s. When the adaptive MPC controller is active, the vehicle tracks the desired yaw rate very well with the maximum tracking error of 0.05 rad/s without significant overshoot or undershoot.

Figure 6.10 illustrates torque adjustments by the MPC controller, revealing that the conventional MPC approach generates yaw moments to achieve better short-term yaw rate tracking performance, such as counterclockwise yaw moments induced by imbalanced front wheel torques when the driver initiates a leftward steer at 3 seconds. However, this reactive approach might compromise the controller's response time for predicting yaw rate limits, potentially causing overshoot. Conversely, the adaptive MPC strategy showcases more precisely timed torque adjustments, attributed to its dynamic prediction horizon, culminating in enhanced control performance by reducing overshoot tendencies and promoting accurate yaw rate tracking.

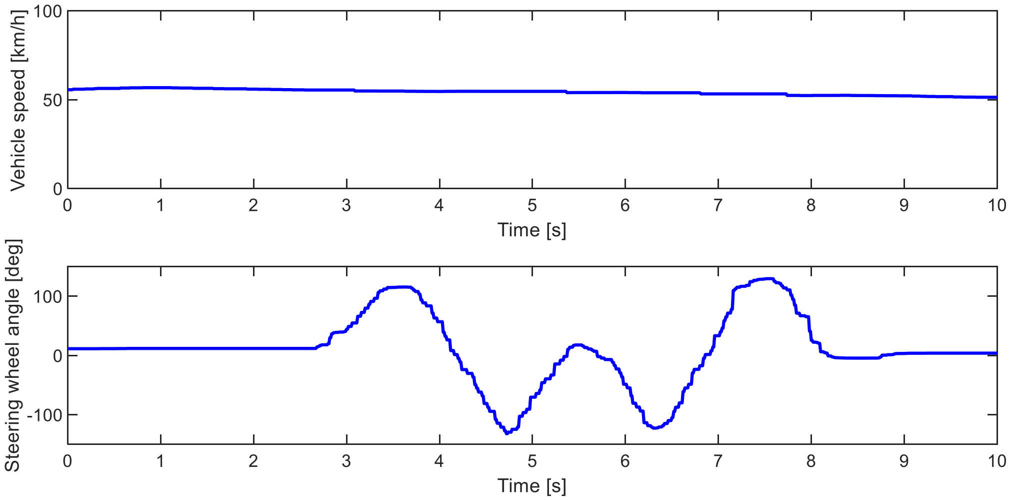
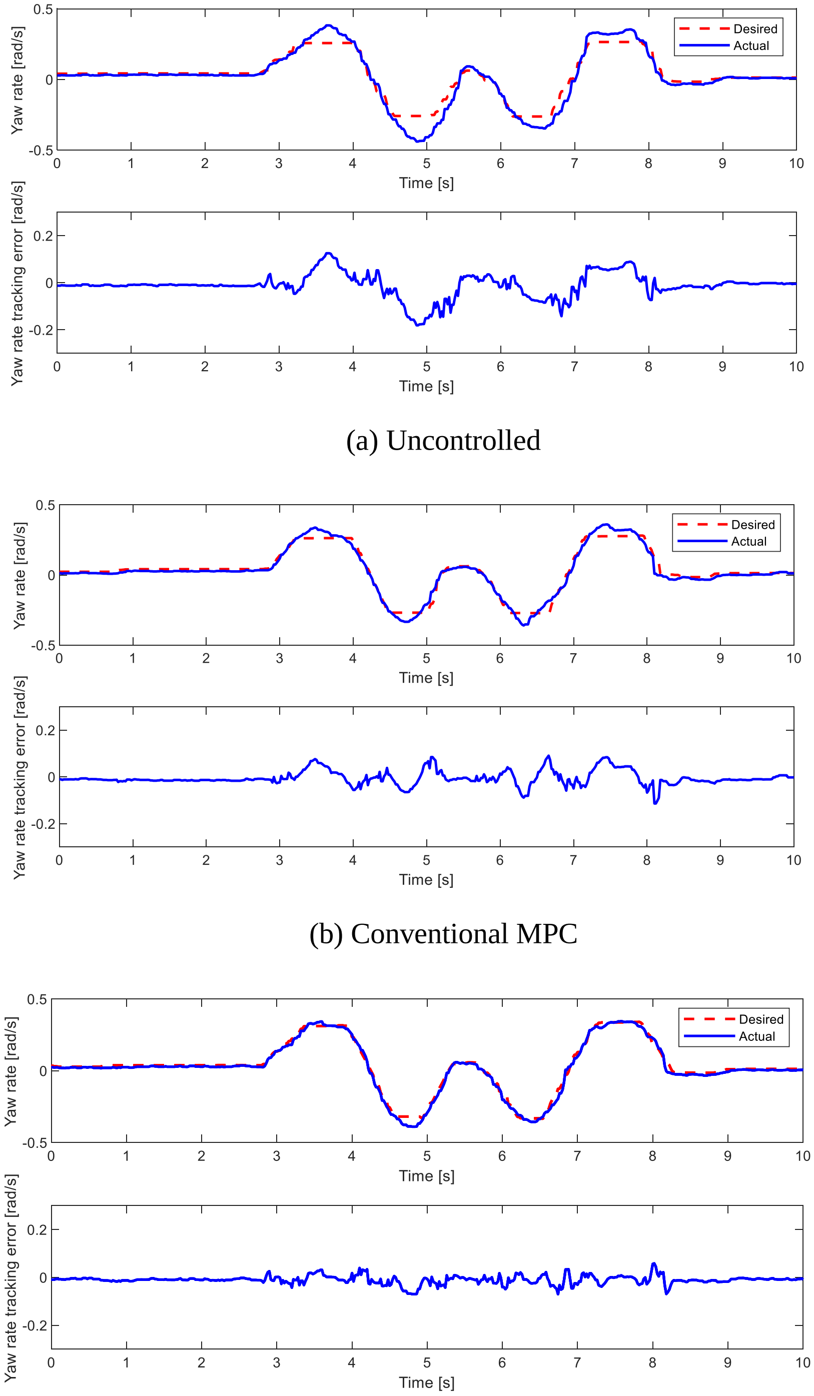


Figure 6.8: Vehicle speed and steering wheel angle in the DLC maneuver.

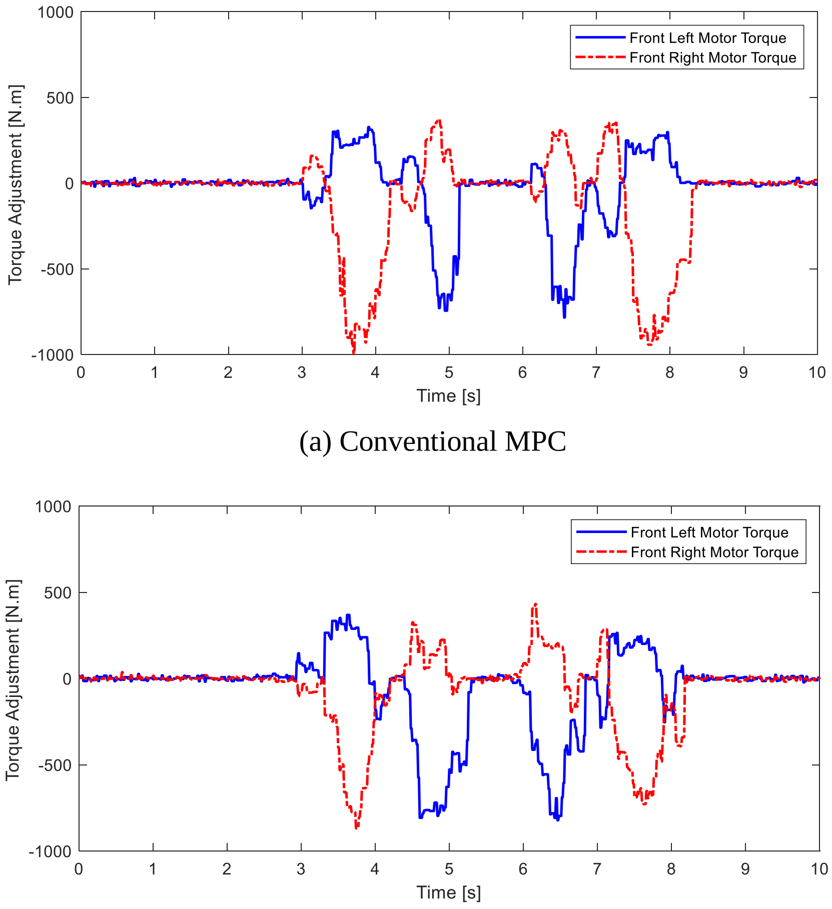


(a) Uncontrolled

(b) Conventional MPC

(c) Adaptive MPC

Figure 6.9: Comparative analysis of yaw rate tracking during the DLC maneuvers: (a) uncontrolled (b) conventional MPC (c) adaptive MPC.



(a) Conventional MPC

(b) Adaptive MPC

Figure 6.10: Comparative analysis of torque adjustment during the DLC maneuvers: (a) conventional MPC (b) adaptive MPC.

6.5 Summary

In this chapter, an adaptive MPC was developed to further enhance vehicle stability control. In the developed adaptive MPC, it continuously monitored the yaw rate and made necessary adjustments to the prediction interval when the yaw rate approached or exceeded its limit. When the yaw rate neared the limit, the controller increased the prediction interval to provide the controller with more time to react and mitigate yaw rate overshoot. Conversely, when the yaw rate remained well within the limit, the prediction interval was reduced, focusing on shorter-term control actions to avoid undershoot in the yaw rate tracking. Remarkably, this adaptative MPC strategy is achieved without compromising the computational efficiency, allowing for real-time vehicle implementation.

The control performance of the developed adaptive MPC was evaluated in simulations. A high-fidelity CarSim model was used to model the vehicle response and comprehensive simulations were conducted using the CarSim-Simulink co-simulation environment. In the simulation study involving a double lane change maneuver at a consistent vehicle speed. Simulation results revealed that with a relatively shorter prediction interval, the yaw rate closely followed the desired yaw rate, indicating effective tracking. However, this tighter prediction horizon led to prominent overshooting or oversteering effects. Conversely, when a relatively longer prediction interval was applied, the yaw rate displayed more significant undershoot or understeering tendencies. These findings underscore the influence of prediction horizon length on control behavior, where a shorter horizon can lead to overshooting due to limited reaction time. In contrast, the adaptive MPC strategy, which dynamically adjusts the prediction interval as the yaw rate approaches the limit, achieves an enhanced yaw tracking performance with reduced tracking error.

The Equinox EV test vehicle was used to further verify the control performance of the adaptive MPC. A slalom maneuver and a double lane change maneuver were applied to the vehicle in the experiments while maintaining a constant vehicle speed. The results from experimental studies demonstrate that the superior performance of the developed adaptive MPC to track the designed yaw rate while maintaining the vehicle in a stable region compared to the fix-prediction-horizon MPC and the non-control mode. By dynamically adjusting the prediction horizon based on real-time system behavior, the control system can make more proactive torque adjustments, mitigating the risk of overshot during yaw rate tracking near its limit.

The study further establishes that the developed adaptive MPC strategy, characterized by its ability to dynamically adjust the prediction interval while maintaining a consistent level of prediction steps, effectively manages the computation cost of the MPC for both real-time implementation and experimental validation. By fine-tuning the prediction interval online to suit the vehicle's dynamic behavior, the adaptive MPC strategy achieves this optimization without compromising the overall computation cost.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this study, an innovative learning Model Predictive Control (MPC) controller was formulated for Holistic Vehicle Control (HVC) by leveraging a novel hybrid approach that combines elements of both physics-based modeling and data-driven techniques. This hybrid control paradigm capitalizes on the strengths of deterministic and resilient physics-based predictive control frameworks while harnessing the learning capabilities of data-driven systems. The effectiveness of this newly devised hybrid learning MPC controller was assessed through a combination of Matlab/Simulink co-simulation with the CarSim vehicle dynamics software and real-world vehicle experiments on a Chevrolet Equinox electric testing vehicle under a variety of harsh driving maneuvers. The research yields several noteworthy outcomes and contributions, summarized as follows.

The foremost contribution of this research lies in the creation of a hybrid learning MPC framework tailored for HVC. Unlike a traditional MPC that typically uses one physics-based prediction model, the proposed hybrid learning MPC maintains two models, a physics-based nominal system dynamics model with bounds on system states and control inputs, as well as a data-based learned system dynamics model with an oracle that can be updated by statistical methods. This innovative structure seamlessly merges the strengths of two distinct approaches – physics-based modeling and data-driven techniques – to yield a control paradigm that is both adaptive and robust. By combining the predictive capabilities of physics-based control with the learning potential of data-driven systems, this hybrid approach introduces a novel dimension to vehicle control strategies, promising improved performance in the realm of vehicle stability control. With the decoupled structure of learning MPC, performance and robust safety are guaranteed. These properties of learning MPC are important for developing an evolving vehicle stability control system.

The second significant contribution of this research involves the establishment of a data-driven prediction model for vehicle dynamics, achieved through the utilization of Gaussian Process Regression (GPR). Through meticulous training and data acquisition, this GPR-based vehicle model exhibits a notable capability for predicting vehicle behaviors. Remarkably, when provided with an ample number of training data points, the GPR-based learned vehicle model demonstrates an enhanced capacity for predicting yaw rate, outperforming the conventional physics-based vehicle model in terms of prediction accuracy.

Thirdly, this research introduced an innovative data management approach designed to strike a balance between prediction performance and computational efficiency. Specifically, a subset-of-data (SoD) methodology was adopted to address the computational challenges associated with GPR in real-time vehicle control systems. The SoD technique effectively curtails the computational load by selecting a limited set of data points that enclose the query point within a bounding box in the feature space. This subset approximates the predictions derived from the comprehensive GPR method, which typically employs the entire training dataset. By implementing the developed SoD strategy, the computational demands of the learning MPC controller are effectively managed, ensuring the system's real-time feasibility for practical application on a test vehicle.

One more noteworthy contribution of this research involves the development of a model authentication technique designed to assess and contrast two distinct vehicle models: the conventional physics-based vehicle model and the acquired data-driven vehicle model. This authentication approach hinges on the context of the query point, which encapsulates the current vehicle operating conditions in the feature space. To gauge the efficacy of the GPR model, this method computes a 95% credible interval that captures the model's predictive uncertainty. By considering the number of training points available and this 95% credible interval, the authentication method discerns the optimal prediction model – either the nominal physics-based model or the learned vehicle model. This discernment process ensures the selection of the most appropriate model for prediction, facilitating accurate and reliable control decisions based on the prevailing conditions.

Another significant aspect of this research is centered on the development of an adaptive MPC approach tailored for HVC. This adaptive strategy introduces a dynamic mechanism that continually monitors the vehicle's yaw rate throughout maneuvers and effectively adapts the prediction interval based on its proximity to its limits. When the yaw rate nears or surpasses these limits, the strategy extends the prediction interval to grant the MPC controller additional reaction time for counteracting yaw rate deviations. In contrast, when the yaw rate remains well within acceptable boundaries, the prediction interval can be compressed, prioritizing shorter-term control actions. By seamlessly adjusting the prediction interval while keeping the prediction steps consistent within the MPC framework, this adaptive strategy guarantees that computational costs remain consistent, even when a more extensive prediction horizon is utilized. This pivotal feature ensures the feasibility of real-time implementation, safeguarding effective control responses in varying dynamic scenarios.

The efficacy of the proposed learning MPC controller and the adaptive MPC controller was thoroughly assessed through rigorous computer simulations conducted in MATLAB/Simulink and CarSim environments. Leveraging a high-fidelity model of available electric SUVs in CarSim, the vehicle's response to prescribed inputs was meticulously simulated. Performance evaluations were carried out across a spectrum of driving conditions, including diverse road surfaces such as wet and dry, to gauge the robustness of the controllers. Comparative analyses were performed, contrasting the vehicle responses under the learning MPC, adaptive MPC, uncontrolled, and conventional MPC scenarios. Impressively, the simulations showcased the superior performance of both the learning MPC and the adaptive MPC controllers. Across various driving scenarios, both controllers consistently upheld vehicle stability.

Subsequently, the proposed methodologies were effectively translated to practical implementation using a Chevrolet Equinox electric testing vehicle, with front torque vectoring as the core actuator. This real-world evaluation encompassed a comprehensive examination of controller behavior during maneuvers like the double lane change and slalom, under differing road conditions. The resulting enhancements in vehicle response and stability reinforced the adaptability and efficacy of the developed controllers, implying their potential to address diverse control challenges beyond vehicle stability in various domains. Importantly, the versatility of the proposed hybrid learning MPC and adaptive MPC methodologies extends beyond vehicle stability applications, proving applicable to a broad spectrum of MPC problems.

Despite its advantages, one notable observation is that hybrid learning MPC can introduce non-smooth control actions, primarily attributed to the switching of prediction models. This phenomenon presents challenges in the context of vehicle stability control. The abrupt or discontinuous control inputs resulting from model transitions can result in passenger discomfort and elevate the risk of vehicular instability.

7.2 Future Work

The realms of learning MPC and adaptive MPC for holistic vehicle control stand as burgeoning domains with immense promise for elevating vehicle stability performance. As the trajectory of this field continues to unfold, several avenues for future exploration and advancements come to the fore:

Tire Dynamics Integration: An important avenue for future research lies in incorporating tire or wheel dynamics within the physics-based vehicle model, enhancing the accuracy and predictive capabilities of learning and adaptive MPC controllers. By integrating sophisticated tire models and learning tire dynamics in real-time, these controllers can better capture complex tire interactions with the road, adapt to changing conditions, and optimize control strategies accordingly.

Enhanced Learning Strategies: Delving deeper into machine learning techniques could unlock novel strategies for learning MPC controllers. Exploring advanced algorithms like reinforcement learning or neural network-based approaches might enable controllers to adapt more dynamically to varying driving scenarios, further refining stability and response.

Enhanced Data Management: Looking ahead, the integration of active learning techniques, like information gain or similarity measurement, could play a pivotal role in optimizing datasets by identifying and eliminating redundant or unnecessary data points. Future research endeavors might focus on devising more streamlined and potent data collection methods, as well as refining approaches for representing diverse driving scenarios, ultimately advancing the efficacy of learning-based control strategies for vehicle stability and control.

Multi-Objective Optimization: Integrating multi-objective optimization within learning and adaptive MPC frameworks could pave the way for controllers that optimize not only stability but also energy efficiency, comfort, and other driving parameters. This would create a holistic approach to vehicle control that balances multiple objectives.

Learning-based Adaptive MPC: By combining data-driven machine learning methods with rule-based adaptation strategies, the system could achieve continuous performance improvement over time. This integration would entail leveraging historical data to refine the prediction model and dynamically adjusting the prediction horizon based on predefined adaptation rules. Such an approach holds significant potential for enhancing the adaptive capabilities of the learning MPC controller, allowing it to effectively adapt to evolving dynamics and optimize control responses, thus contributing to further advancements in vehicle stability and control.

References

[1] A. A. Mohammed, K. Ambak, A. M. Mosa, and D. Syamsunur, "A review of traffic accidents and related practices worldwide," *The Open Transportation Journal,* vol. 13, no. 1, 2019.

[2] A. Chouinard and J.-F. Lécuyer, "A study of the effectiveness of Electronic Stability Control in Canada," *Accident Analysis & Prevention,* vol. 43, no. 1, pp. 451-460, 2011.

[3] S. A. Ferguson, "The effectiveness of electronic stability control in reducing real-world crashes: a literature review," *Traffic injury prevention,* vol. 8, no. 4, pp. 329-338, 2007.

[4] M. Choi and S. B. Choi, "Model predictive control for vehicle yaw stability with practical concerns," *IEEE Transactions on Vehicular Technology,* vol. 63, no. 8, pp. 3539-3548, 2014.

[5] D. Hrovat, S. Di Cairano, H. E. Tseng, and I. V. Kolmanovsky, "The development of model predictive control in automotive industry: A survey," in *2012 IEEE International Conference on Control Applications,* 2012: IEEE, pp. 295-302.

[6] L. Del Re, F. Allgöwer, L. Glielmo, C. Guardiola, and I. Kolmanovsky, *Automotive model predictive control: models, methods and applications*. Springer, 2010.

[7] A. Aswani, P. Bouffard, and C. Tomlin, "Extensions of learning-based model predictive control for real-time application to a quadrotor helicopter," in *2012 American Control Conference (ACC),* 2012: IEEE, pp. 4661-4666.

[8] A. Aswani, H. Gonzalez, S. S. Sastry, and C. Tomlin, "Provably safe and robust learning-based model predictive control," *Automatica,* vol. 49, no. 5, pp. 1216-1226, 2013/05/01/ 2013.

[9] A. Aswani, N. Master, J. Taneja, D. Culler, and C. Tomlin, "Reducing Transient and Steady State Electricity Consumption in HVAC Using Learning-Based Model-Predictive Control," *Proceedings of the IEEE,* vol. 100, no. 1, pp. 240-253, 2012, doi: 10.1109/JPROC.2011.2161242.

[10] A. Aswani, N. Master, J. Taneja, A. Krioukov, D. Culler, and C. Tomlin, "Energy-Efficient Building HVAC Control Using Hybrid System LBMPC," *IFAC Proceedings Volumes,* vol. 45, no. 17, pp. 496-501, 2012/01/01/ 2012.

[11] M. Jalaliyazdi, "Integrated vehicle stability control and power distribution using model predictive control," 2016.

[12] R. K. Subroto, C. Z. Wang, and K. L. Lian, "Four-wheel independent drive electric vehicle stability control using novel adaptive sliding mode control," *IEEE Transactions on Industry Applications,* vol. 56, no. 5, pp. 5995-6006, 2020.

[13] H. Wang, H. Zhang, J. Han, and C. Liu, "A novel H∞ observer for sensor fault diagnosis of vehicle electronic stability control system with disturbance and input delay," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering,* p. 09544070221131839, 2022.

[14] J. Wu, D. Wu, Y. Yan, N. Zhang, C. Bao, and F. Wang, "Steering and braking game control architecture based minimax robust stability control for emergency avoidance of autonomous vehicles," *Science China Technological Sciences,* vol. 65, no. 4, pp. 943-955, 2022.

[15] L. Ge, Y. Zhao, F. Ma, and K. Guo, "Towards longitudinal and lateral coupling control of autonomous vehicles using offset free MPC," *Control Engineering Practice,* vol. 121, p. 105074, 2022.

[16] R. Hajiloo, A. Khajepour, H. Zengin, A. Kasaiezadeh, and S.-K. Chen, "A coupled force predictive control of vehicle stability using front/rear torque allocation with experimental verification," *Vehicle System Dynamics,* vol. 60, no. 7, pp. 2541-2563, 2022.

[17] N. Ahmadian, A. Khosravi, and P. Sarhadi, "Driver assistant yaw stability control via integration of AFS and DYC," *Vehicle system dynamics,* vol. 60, no. 5, pp. 1742-1762, 2022.

[18] H. Chen and C. Lv, "RHONN-Modeling-Based Predictive Safety Assessment and Torque Vectoring for Holistic Stabilization of Electrified Vehicles," *IEEE/ASME Transactions on Mechatronics,* vol. 28, no. 1, pp. 450-460, 2022.

[19] N. Guo, X. Zhang, Y. Zou, B. Lenzo, G. Du, and T. Zhang, "A supervisory control strategy of distributed drive electric vehicles for coordinating handling, lateral stability, and energy efficiency," *IEEE Transactions on Transportation Electrification,* vol. 7, no. 4, pp. 2488-2504, 2021.

[20] E. M. Ahmed, E. A. Mohamed, A. Elmelegi, M. Aly, and O. Elbaksawi, "Optimum modified fractional order controller for future electric vehicles and renewable energy-based interconnected power systems," *IEEE Access,* vol. 9, pp. 29993-30010, 2021.

[21] C. G. Bobier-Tiu, C. E. Beal, J. C. Kegelman, R. Y. Hindiyeh, and J. C. Gerdes, "Vehicle control synthesis using phase portraits of planar dynamics," *Vehicle System Dynamics,* vol. 57, no. 9, pp. 1318-1337, 2019.

[22] W. Liu, L. Xiong, B. Leng, H. Meng, and R. Zhang, "Vehicle stability criterion research based on phase plane method," SAE Technical Paper, 0148-7191, 2017.

[23] R. Hajiloo, M. Abroshan, A. Khajepour, A. Kasaiezadeh, and S.-K. Chen, "Integrated Steering and Differential Braking for Emergency Collision Avoidance in Autonomous Vehicles," *IEEE Transactions on Intelligent Transportation Systems,* 2020.

[24] Y. Shibahata, K. Shimada, and T. Tomari, "Improvement of vehicle maneuverability by direct yaw moment control," *Vehicle System Dynamics,* vol. 22, no. 5-6, pp. 465-481, 1993.

[25] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[26] W. Huang, P. K. Wong, K. I. Wong, C. M. Vong, and J. Zhao, "Adaptive neural control of vehicle yaw stability with active front steering using an improved random projection neural network," *Vehicle system dynamics,* vol. 59, no. 3, pp. 396-414, 2021.

[27] M. Jalali, S. Khosravani, A. Khajepour, S.-k. Chen, and B. Litkouhi, "Model predictive control of vehicle stability using coordinated active steering and differential brakes," *Mechatronics,* vol. 48, pp. 30-41, 2017.

[28] A. Nahidi, A. Kasaiezadeh, S. Khosravani, A. Khajepour, S.-K. Chen, and B. Litkouhi, "Modular integrated longitudinal and lateral vehicle stability control for electric vehicles," *Mechatronics,* vol. 44, pp. 60-70, 2017.

[29] Z. Lu, B. Shyrokau, B. Boulkroune, S. Van Aalst, and R. Happee, "Performance benchmark of state-of-the-art lateral path-following controllers," in *2018 IEEE 15th International Workshop on Advanced Motion Control (AMC)*, 2018: IEEE, pp. 541-546.

[30] J. Shah, "Development and Control of Evasive Steer Assist Using Rear Wheel Steering," SAE Technical Paper, 0148-7191, 2015.

[31] C. Ackermann, J. Bechtloff, and R. Isermann, "Collision avoidance with combined braking and steering," in *6th International Munich Chassis Symposium 2015: chassis. tech plus,* 2015: Springer, pp. 199-213.

[32] P. Hang, X. Xia, G. Chen, and X. Chen, "Active safety control of automated electric vehicles at driving limits: A tube-based MPC approach," *IEEE Transactions on Transportation Electrification,* vol. 8, no. 1, pp. 1338-1349, 2021.

[33] H. Peng, W. Wang, Q. An, C. Xiang, and L. Li, "Path tracking and direct yaw moment coordinated control based on robust MPC with the finite time horizon for autonomous independent-drive vehicles," *IEEE Transactions on Vehicular Technology,* vol. 69, no. 6, pp. 6053-6066, 2020.

[34] H. Wang, B. Liu, X. Ping, and Q. An, "Path Tracking Control for Autonomous Vehicles Based on an Improved MPC," *IEEE Access,* vol. 7, pp. 161064-161073, 2019.

[35] S. Cheng, L. Li, H.-Q. Guo, Z.-G. Chen, and P. Song, "Longitudinal collision avoidance and lateral stability adaptive control system based on MPC of autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems,* vol. 21, no. 6, pp. 2376-2385, 2019.

[36] A. Norouzi, H. Heidarifar, H. Borhan, M. Shahbakhti, and C. R. Koch, "Integrating machine learning and model predictive control for automotive applications: A review and future directions," *Engineering Applications of Artificial Intelligence,* vol. 120, p. 105878, 2023.

[37] S. Li, G. Wang, B. Zhang, Z. Yu, and G. Cui, "Vehicle Yaw Stability Control at the Handling Limits Based on Model Predictive Control," *International Journal of Automotive Technology,* vol. 21, no. 2, pp. 361-370, 2020.

[38] O. Barbarisi, G. Palmieri, S. Scala, and L. Glielmo, "LTV-MPC for yaw rate control and side slip control with dynamically constrained differential braking," *European Journal of Control,* vol. 15, no. 3-4, pp. 468-479, 2009.

[39] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Transactions on control systems technology,* vol. 15, no. 3, pp. 566-580, 2007.

[40] P. Falcone, H. Eric Tseng, F. Borrelli, J. Asgari, and D. Hrovat, "MPC-based yaw and lateral stabilisation via active front steering and braking," *Vehicle System Dynamics,* vol. 46, no. S1, pp. 611-628, 2008.

[41] M. Metzler, D. Tavernini, P. Gruber, and A. Sorniotti, "On Prediction Model Fidelity in Explicit Nonlinear Model Predictive Vehicle Stability Control," *IEEE Transactions on Control Systems Technology,* 2020.

[42] H. Guo, F. Liu, F. Xu, H. Chen, D. Cao, and Y. Ji, "Nonlinear model predictive lateral stability control of active chassis for intelligent vehicles and its FPGA implementation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* vol. 49, no. 1, pp. 2-13, 2017.

[43] B. Yi, S. Gottschling, J. Ferdinand, N. Simm, F. Bonarens, and C. Stiller, "Real time integrated vehicle dynamics control and trajectory planning with MPC for critical maneuvers," in *2016 IEEE intelligent vehicles symposium (IV),* 2016: IEEE, pp. 584-589.

[44] Z. Li, P. Wang, S. Cai, X. Hu, and H. Chen, "NMPC-based controller for vehicle longitudinal and lateral stability enhancement under extreme driving conditions," *ISA transactions,* vol. 135, pp. 509-523, 2023.

[45] Y. Liang, Y. Li, A. Khajepour, and L. Zheng, "Holistic adaptive multi-model predictive control for the path following of 4WID autonomous vehicles," *IEEE Transactions on Vehicular Technology,* vol. 70, no. 1, pp. 69-81, 2020.

[46] C. Zhou, L. Yu, Y. Li, Z. Lu, and J. Song, "A layered roll stability control strategy for commercial vehicles based on adaptive model predictive control," *Vehicle System Dynamics,* pp. 1-22, 2022.

[47] J. Wu, Z. Wang, and L. Zhang, "Unbiased-estimation-based and computation-efficient adaptive MPC for four-wheel-independently-actuated electric vehicles," *Mechanism and Machine Theory,* vol. 154, p. 104100, 2020.

[48] F. Lin, Y. Chen, Y. Zhao, and S. Wang, "Path tracking of autonomous vehicle based on adaptive model predictive control," *International Journal of Advanced Robotic Systems,* vol. 16, no. 5, p. 1729881419880089, 2019.

[49] W. Wang, Y. Zhang, C. Yang, T. Qie, and M. Ma, "Adaptive model predictive control-based path following control for four-wheel independent drive automated vehicles," *IEEE Transactions on Intelligent Transportation Systems,* vol. 23, no. 9, pp. 14399-14412, 2021.

[50] Y. Liang, Y. Li, A. Khajepour, Y. Huang, Y. Qin, and L. Zheng, "A novel combined decision and control scheme for autonomous vehicle in structured road based on adaptive model predictive control," *IEEE Transactions on Intelligent Transportation Systems,* vol. 23, no. 9, pp. 16083-16097, 2022.

[51] A. Busch, M. Wielitzka, T. Ortmaier, and V. Kleyman, "Adaptive model predictive traction control for electric vehicles," in *2019 18th European Control Conference (ECC)*, 2019: IEEE, pp. 1239-1244.

[52] A. Vahidi, A. Stefanopoulou, and H. Peng, "Adaptive model predictive control for co‐ordination of compression and friction brakes in heavy duty vehicles," *International Journal of Adaptive Control and Signal Processing,* vol. 20, no. 10, pp. 581-598, 2006.

[53] B. B. Gupta, A. Gaurav, E. C. Marín, and W. Alhalabi, "Novel graph-based machine learning technique to secure smart vehicles in intelligent transportation systems," *IEEE transactions on intelligent transportation systems,* 2022.

[54] N. Sapoval *et al.*, "Current progress and open challenges for applying deep learning across the biosciences," *Nature Communications,* vol. 13, no. 1, p. 1728, 2022.

[55] G. Revathy, S. A. Alghamdi, S. M. Alahmari, S. R. Yonbawi, A. Kumar, and M. A. Haq, "Sentiment analysis using machine learning: Progress in the machine intelligence for data science," *Sustainable Energy Technologies and Assessments,* vol. 53, p. 102557, 2022.

[56] K. Alanne and S. Sierla, "An overview of machine learning applications for smart buildings," *Sustainable Cities and Society,* vol. 76, p. 103445, 2022.

[57] A. Lavin *et al.*, "Technology readiness levels for machine learning systems," *Nature Communications,* vol. 13, no. 1, p. 6039, 2022.

[58] A. McCarn Deiana *et al.*, "Applications and Techniques for Fast Machine Learning in Science," *Frontiers in Big Data,* 2021.

[59] S. Qiu *et al.*, "Multi-sensor information fusion based on machine learning for real applications in human activity recognition: State-of-the-art and research challenges," *Information Fusion,* vol. 80, pp. 241-265, 2022.

[60] J. P. Usuga Cadavid, S. Lamouri, B. Grabot, R. Pellerin, and A. Fortin, "Machine learning applied in production planning and control: a state-of-the-art in the era of industry 4.0," *Journal of Intelligent Manufacturing,* vol. 31, pp. 1531-1558, 2020.

[61] M. Jafari *et al.*, "Feedback control of bioelectronic devices using machine learning," *IEEE Control Systems Letters,* vol. 5, no. 4, pp. 1133-1138, 2020.

[62] J. Chen, X. Gao, R. Deng, Y. He, C. Fang, and P. Cheng, "Generating adversarial examples against machine learning-based intrusion detector in industrial control systems," *IEEE Transactions on Dependable and Secure Computing,* vol. 19, no. 3, pp. 1810-1825, 2020.

[63] A. M. Koay, R. K. L. Ko, H. Hettema, and K. Radke, "Machine learning in industrial control system (ICS) security: current landscape, opportunities and challenges," *Journal of Intelligent Information Systems,* vol. 60, no. 2, pp. 377-405, 2023.

[64] M. A. Umer, K. N. Junejo, M. T. Jilani, and A. P. Mathur, "Machine learning for intrusion detection in industrial control systems: Applications, challenges, and recommendations," *International Journal of Critical Infrastructure Protection,* vol. 38, p. 100516, 2022.

[65] M. Aliramezani, C. R. Koch, and M. Shahbakhti, "Modeling, diagnostics, optimization, and control of internal combustion engines via modern machine learning techniques: A review and future directions," *Progress in Energy and Combustion Science,* vol. 88, p. 100967, 2022.

[66] M. R. Bachute and J. M. Subhedar, "Autonomous driving architectures: insights of machine learning and deep learning algorithms," *Machine Learning with Applications,* vol. 6, p. 100164, 2021.

[67] Y. Zhang, X. Shi, H. Zhang, Y. Cao, and V. Terzija, "Review on deep learning applications in frequency analysis and control of modern power system," *International Journal of Electrical Power & Energy Systems,* vol. 136, p. 107744, 2022.

[68] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annual Review of Control, Robotics, and Autonomous Systems,* vol. 3, pp. 269-296, 2020.

[69] D. Limon, J. Calliess, and J. M. Maciejowski, "Learning-based nonlinear model predictive control," *IFAC-PapersOnLine,* vol. 50, no. 1, pp. 7769-7776, 2017.

[70] P. Bouffard, A. Aswani, and C. Tomlin, "Learning-based model predictive control on a quadrotor: Onboard implementation and experimental results," in *2012 IEEE International Conference on Robotics and Automation*, 2012: IEEE, pp. 279-284.

[71] C. J. Ostafew, A. P. Schoellig, T. D. Barfoot, and J. Collier, "Learning‐based nonlinear model predictive control to improve vision‐based mobile robot path tracking," *Journal of Field Robotics,* vol. 33, no. 1, pp. 133-152, 2016.

[72] C. J. Ostafew, A. P. Schoellig, and T. D. Barfoot, "Robust constrained learning-based NMPC enabling reliable mobile robot path tracking," *The International Journal of Robotics Research,* vol. 35, no. 13, pp. 1547-1563, 2016.

[73] Z. Zhang, L. Xie, S. Lu, X. Wu, and H. Su, "Vehicle yaw stability control with a two-layered learning MPC," *Vehicle system dynamics,* vol. 61, no. 2, pp. 423-444, 2023.

[74] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger, "Learning-based model predictive control for autonomous racing," *IEEE Robotics and Automation Letters,* vol. 4, no. 4, pp. 3363-3370, 2019.

[75] R. Soloperto, J. Köhler, and F. Allgöwer, "Augmenting MPC schemes with active learning: Intuitive tuning and guaranteed performance," *IEEE Control Systems Letters,* vol. 4, no. 3, pp. 713-718, 2020.

[76] D. Piga, M. Forgione, S. Formentin, and A. Bemporad, "Performance-oriented model learning for data-driven MPC design," *IEEE control systems letters,* vol. 3, no. 3, pp. 577-582, 2019.

[77] P. T. Jardine, S. N. Givigi, and S. Yousefi, "Experimental results for autonomous model-predictive trajectory planning tuned with machine learning," in *2017 Annual IEEE International Systems Conference (SysCon),* 2017: IEEE, pp. 1-7.

[78] P. T. Jardine, S. Givigi, and S. Yousefi, "Parameter tuning for prediction-based quadcopter trajectory planning using learning automata," *IFAC-PapersOnLine,* vol. 50, no. 1, pp. 2341-2346, 2017.

[79] K. P. Wabersich, L. Hewing, A. Carron, and M. N. Zeilinger, "Probabilistic model predictive safety certification for learning-based control," *arXiv preprint arXiv:1906.10417,* 2019.

[80] K. P. Wabersich and M. N. Zeilinger, "Safe exploration of nonlinear dynamical systems: A predictive safety filter for reinforcement learning," *arXiv preprint arXiv:1812.05506,* 2018.

[81] R. P. Borase, D. Maghade, S. Sondkar, and S. Pawar, "A review of PID control, tuning methods and applications," *International Journal of Dynamics and Control,* vol. 9, pp. 818-827, 2021.

[82] H. O. Bansal, R. Sharma, and P. Shreeraman, "PID controller tuning techniques: a review," *J. Control Eng. Technol,* vol. 2, no. 4, pp. 168-176, 2012.

[83] M. Schwenzer, M. Ay, T. Bergs, and D. Abel, "Review on model predictive control: An engineering perspective," *The International Journal of Advanced Manufacturing Technology,* vol. 117, no. 5-6, pp. 1327-1349, 2021.

[84] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice—A survey," *Automatica,* vol. 25, no. 3, pp. 335-348, 1989.

[85] J. Richalet, "Industrial applications of model based predictive control," *Automatica,* vol. 29, no. 5, pp. 1251-1274, 1993.

[86] J. B. Rawlings, "Tutorial overview of model predictive control," *IEEE control systems magazine,* vol. 20, no. 3, pp. 38-52, 2000.

[87] A. Aswani and C. Tomlin, "Reachability algorithm for biological piecewise-affine hybrid systems," in *International Workshop on Hybrid Systems: Computation and Control,* 2007: Springer, pp. 633-636.

[88] F. Borrelli, *Constrained optimal control of linear and hybrid systems*. Springer, 2003.

[89] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[90] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[91] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning* (no. 10). Springer series in statistics New York, 2001.

[92] D. Duvenaud, "Automatic model construction with Gaussian processes," University of Cambridge, 2014.

[93] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning,* 2003: Springer, pp. 63-71.

[94] F. Leclercq, "Bayesian optimization for likelihood-free cosmological inference," *Physical Review D,* vol. 98, no. 6, p. 063511, 2018.

[95] J. Kocijan, *Modelling and control of dynamic systems using Gaussian process models*. Springer, 2016.

[96] R. Rezvani Arany, "Gaussian process model predictive control for autonomous driving in safety-critical scenarios," ed, 2019.

[97] L. Zahedi, F. G. Mohammadi, S. Rezapour, M. W. Ohland, and M. H. Amini, "Search Algorithms for Automated Hyper-Parameter Tuning," *arXiv preprint arXiv:2104.14677,* 2021.

[98] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proceedings of the IEEE,* vol. 104, no. 1, pp. 148-175, 2015.

[99] G. Kopsiaftis, E. Protopapadakis, A. Voulodimos, N. Doulamis, and A. Mantoglou, "Gaussian process regression tuned by bayesian optimization for seawater intrusion prediction," *Computational intelligence and neuroscience,* vol. 2019, 2019.

[100] N. B. Apoorv Agnihotri. (2020) Exploring Bayesian Optimization. *Distill*.

[101] K. Pearson, "VII. Note on regression and inheritance in the case of two parents," *proceedings of the royal society of London,* vol. 58, no. 347-352, pp. 240-242, 1895.

[102] D. S. Moore and S. Kirkland, *The basic practice of statistics*. WH Freeman New York, 2007.

[103] A. Taghavipour, M. Vajedi, and N. L. Azad, *Intelligent control of connected plug-in hybrid electric vehicles*. Springer, 2019.

[104] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *2015 IEEE Intelligent Vehicles Symposium (IV),* 2015: IEEE, pp. 1094-1099.

[105] H. Pacejka, *Tire and vehicle dynamics*. Elsevier, 2005.

[106] A. Y. Ungoren, H. Peng, and H. Tseng, "A study on lateral speed estimation methods," *International Journal of Vehicle Autonomous Systems,* vol. 2, no. 1-2, pp. 126-144, 2004.

[107] H. Liu, Y.-S. Ong, X. Shen, and J. Cai, "When Gaussian process meets big data: A review of scalable GPs," *IEEE transactions on neural networks and learning systems,* vol. 31, no. 11, pp. 4405-4423, 2020.