Lane-Keeping System with Adaptive Model Predictive Control using Online Dynamic Mode Decomposition

by
Okan ERTÜRK

Submitted to
the Graduate School of Engineering and Natural Sciences
in partial fulfilment of the requirements for the degree of
Master of Science

Sabanci University
December, 2024

© 2024 by Okan ERTÜRK.
All Rights Reserved.

Lane-Keeping System with Adaptive Model Predictive Control using Online Dynamic Mode Decomposition

Okan ERTÜRK

ME, Master of Science Thesis, 2024

Thesis Advisor: Prof. Dr. Mustafa Ünel

Keywords: Data-Driven Modeling, Dynamic Mode Decomposition, Model Predictive Control, Lane-Keeping Control, Vehicle Dynamics, Adaptive Control

Abstract

Autonomous vehicles have become an integral part of modern transportation systems, enabling a wide range of applications from personal mobility to industrial automation. The increasing complexity of real-world driving environments has necessitated advanced control strategies and real-time modeling techniques to ensure precision and safety. Among these challenges, lane-keeping control remains a critical focus for achieving robust and accurate trajectory tracking under dynamic and uncertain conditions.

This thesis presents an integrated approach combining Online Dynamic Mode Decomposition (Online DMD) with advanced control architectures for lane-keeping applications in autonomous vehicles. The Online DMD model continuously identifies system dynamics in real time, providing the basis for adaptive control strategies. A hierarchical control framework is implemented, with upper-level proportional controllers generating reference signals and lower-level controllers—Model Predictive Control (MPC) and Linear Quadratic Regulator (LQR)—executing the trajectory tracking.

The proposed methodology is evaluated in a high-fidelity simulation environment using realistic road scenarios, including constant curvature roads, sharp turns, and city-like environments. External disturbances and measurement noise are incor-

porated to test robustness. The results demonstrate that Online DMD effectively captures system dynamics, enabling the controllers to maintain lane-keeping accuracy and stability. The study highlights the potential of data-driven modeling and adaptive control strategies for advancing real-world autonomous vehicle applications.

Özet

Çevrimiçi Dinamik Mod Ayrışımı Kullanarak Uyarlamalı Model Öngörülü Kontrol ile Şerit Takip Sistemi

Okan ERTÜRK

ME, Yüksek Lisans Tezi, 2024

Tez Danışmanı: Prof. Dr. Mustafa Ünel

Anahtar Kelimeler: Veriye Dayalı Modelleme, Dinamik Mod Ayrıştırma, Model Tahminli Kontrol, Şerit Takip Kontrolü, Araç Dinamikleri, Uyarlamalı Kontrol

Otonom araçlar, kişisel mobiliteden endüstriyel otomasyona kadar çok çeşitli uygulamalara olanak tanıyarak modern ulaşım sistemlerinin ayrılmaz bir parçası haline gelmiştir. Gerçek dünyadaki sürüş ortamlarının artan karmaşıklığı, hassasiyet ve güvenliği sağlamak için gelişmiş kontrol stratejileri ve gerçek zamanlı modelleme tekniklerini gerekli kılmıştır. Bu zorluklar arasında, şerit takip kontrolü dinamik ve belirsiz koşullar altında sağlam ve doğru yörünge takibi elde etmek için kritik bir odak noktası olmaya devam etmektedir.

Bu tez, otonom araçlarda şerit takip uygulamaları için Çevrimiçi Dinamik Mod Ayrıştırmayı (Çevrimiçi DMD) gelişmiş kontrol mimarileriyle birleştiren entegre bir yaklaşım sunmaktadır. Çevrimiçi DMD modeli, sistem dinamiklerini gerçek zamanlı olarak sürekli olarak belirleyerek uyarlanabilir kontrol stratejileri için temel sağlar. Üst düzey orantılı kontrolörlerin referans sinyalleri ürettiği ve alt düzey kontrolörlerin (Model Tahmini Kontrol (MPC) ve Doğrusal Kuadratik Regülatör (LQR)) yörünge takibini gerçekleştirdiği hiyerarşik bir kontrol çerçevesi uygulanır.

Önerilen metodoloji, sabit virajlı yollar, keskin dönüşler ve şehir benzeri ortamlar dahil olmak üzere gerçekçi yol senaryoları kullanılarak yüksek doğruluklu bir simülasyon ortamında değerlendirilir. Sağlamlığı test etmek için harici rahatsızlıklar ve ölçüm gürültüsü dahil edilir. Sonuçlar, Çevrimiçi DMD'nin sistem dinamik-

lerini etkili bir şekilde yakaladığını ve denetleyicilerin şerit tutma doğruluğunu ve kararlılığını korumasını sağladığını göstermektedir. Çalışma, gerçek dünyadaki otonom araç uygulamalarını ilerletmek için veri odaklı modelleme ve uyarlanabilir kontrol stratejilerinin potansiyelini vurgulamaktadır.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my thesis advisor, Prof. Dr. Mustafa Ünel, for his unwavering guidance, invaluable support, and insightful advice throughout my graduate studies. His vast knowledge, enthusiasm for research, and dedication to teaching have been a source of constant inspiration for me. I am sincerely thankful for the time and effort he devoted to mentoring me, as well as the opportunity to work under his supervision. His encouragement and belief in my abilities have greatly contributed to both my academic and personal growth, and I will carry his lessons forward in all my future endeavors.

I would also like to extend my heartfelt thanks to the members of my thesis jury for their constructive feedback and thoughtful suggestions, which have helped improve the quality of my work. Their expertise and input have been invaluable in shaping the final version of this thesis.

Lastly, I am deeply indebted to my family for their unconditional love, patience, and encouragement. My parents, Ünal and Meral Ertürk, my sibling, Volkan Ertürk, and my girlfriend, Gülara, have always been my strongest pillars of support. Their belief in me has provided me with the strength and determination to overcome challenges and pursue my dreams. To them, I dedicate this work.

Contents

Abstract **iv**

Özet **vi**

Acknowledgments **vii**

Contents **x**

List of Figures **xii**

List of Tables **xiii**

1 Introduction **1**

1.1 Motivation 3

1.2 Contributions of the Thesis 4

1.3 Outline of the Thesis 4

2 Literature Review and Background **6**

2.1 Vehicle Dynamics 6

2.1.1 Coordinate System 6

2.1.2 Single-Track Bicycle Model 7

2.1.3 Linear Tire Model 8

2.1.4 Model Assumptions and Limitations 9

2.2 Dynamic Mode Decomposition 9

2.2.1 Dynamic Mode Decomposition 10

2.2.2 Dynamic Mode Decomposition with Control 10

2.2.3 Extended Dynamic Mode Decomposition 12

2.2.4 Physics-Informed Dynamic Mode Decomposition 13

2.2.5 Online DMD Algorithm 15

2.3 Recursive Least Squares and its Comparison with Online DMD . . . 17

2.3.1 Overview of RLS Algorithm 17

2.3.2 Comparison with Online DMD 18

2.4 Lane-Keeping System Control Strategies 19
2.4.1 Geometry-Based Controllers 19
2.4.2 Robust Control 20
2.4.3 Optimization-Based Controllers 21
2.4.4 Data-Driven Controllers 22

3 Lateral Vehicle Dynamics with Online Dynamic Mode Decomposition 23
3.1 Modeling the Vehicle Dynamics System 23
3.1.1 Model Description 23
3.1.2 Dynamics for the Lateral Vehicle Model 24
3.2 Simulation Setup and Implementation 25
3.3 Simulation Results 26
3.3.1 Evaluation Metrics 27
3.3.2 Sinusoidal Steering Input at Low Speed 27
3.3.3 Chirp Signal Steering Input 28
3.4 Discussion 32
3.4.1 Performance Across Different Scenarios 33

4 LQR and Adaptive MPC Design using Online DMD 35
4.1 Upper-Level Controller 36
4.2 Lower-Level Controller 37
4.2.1 Overview of Lower-Level Control Strategies 37
4.2.2 Linear Quadratic Regulator (LQR) Design 38
4.2.3 Adaptive Model Predictive Control Design 39
4.3 Integration of Online DMD in Control Design 41

5 Simulation Results and Discussion 42
5.1 Simulation Setup 42
5.1.1 Vehicle Parameters 42
5.1.2 Sensor Setup 43
5.1.3 Controller Parameters 44
5.1.4 Scenario Design 46
5.1.5 Initialization 46
5.1.6 Disturbance Force Configuration 47
5.2 Case 1: Constant Curvature Road 48
5.2.1 Trajectory Visualization 48
5.2.2 Model Evaluation 49
5.2.3 Control Evaluation 51
5.3 Case 2: Sharp Turn 54

5.3.1 Trajectory Visualization 54
5.3.2 Model Evaluation 55
5.3.3 Control Evaluation 57
5.4 Case 3: City-Like Road 60
5.4.1 Trajectory Visualization 60
5.4.2 Model Evaluation 62
5.4.3 Control Evaluation 64
5.5 RLS and Online DMD for Sharp Turn 66
5.6 Discussion 69

6 Conclusion and Future Works **72**

Bibliography **74**

List of Figures

1.1 SAE J3016 Levels of Driving Automation (SAE, 2016) [47]. 2

2.1 Vehicle dynamics coordinate system in a road-aligned frame [51]. . . . 7
2.2 piDMD: An example application to a traveling wave system, where shift-equivariant properties guide the choice of a circulant matrix model. Adapted from [3]. 15
2.3 Online DMD: $\mathbf{A}_k$ is updated with each new snapshot pair $(\mathbf{x}_{k+1}, \mathbf{y}_{k+1})$, adapting to time-varying dynamics. 15
2.4 Windowed DMD: $\mathbf{A}_k$ depends only on the most recent w snapshots, discarding older data as new snapshots arrive. 16

3.1 Sinusoidal steering wheel input 25
3.2 Chirp signal (sine sweep) input for broad frequency excitation. 26
3.3 Comparison of lateral velocity for sinusoidal input at 5 m/s ($\mu = 0.9$). 27
3.4 Comparison of yaw rate for sinusoidal input at 5 m/s ($\mu = 0.9$). . . . 28
3.5 Comparison of lateral velocity for chirp signal input at 5 m/s ($\mu = 0.9$). 29
3.6 Comparison of yaw rate for chirp signal input at 5 m/s ($\mu = 0.9$). . . 29
3.7 Comparison of lateral velocity for chirp signal input at 5 m/s ($\mu = 0.3$). 30
3.8 Comparison of yaw rate for chirp signal input at 5 m/s ($\mu = 0.3$). . . 30
3.9 Comparison of lateral velocity for chirp signal input at 15 m/s ($\mu = 0.9$). 31
3.10 Comparison of yaw rate for chirp signal input at 15 m/s ($\mu = 0.9$). . . 31
3.11 Comparison of lateral velocity for chirp signal input at 15 m/s ($\mu = 0.3$). 32
3.12 Comparison of yaw rate for chirp signal input at 15 m/s ($\mu = 0.3$). . . 32

4.1 Hierarchical control system architecture for the lane-keeping system. . 36
4.2 Upper-level controller design: lane information from both the left and right lane markers is used to compute the lane center. 37

5.1 Driving scenarios designed for evaluation: (a) Constant Curvature, (b) Sharp Turn, and (c) City-Like Road. 46
5.2 Lateral disturbance force applied during the city-like road scenario. . 48

5.3 Trajectory visualization for the constant curvature road scenario. . . . 49
5.4 Real-time modeling error for lateral velocity on the constant curvature road. 50
5.5 Real-time modeling error for yaw rate on the constant curvature road. 50
5.6 Lateral deviation for the constant curvature road scenario. 52
5.7 Relative yaw angle for the constant curvature road scenario. 52
5.8 Steering angle input over time for the constant curvature road scenario. 53
5.9 Trajectory visualization for the sharp turn scenario. 55
5.10 Real-time modeling error for lateral velocity on the sharp turn scenario. 56
5.11 Real-time modeling error for yaw rate on the sharp turn scenario. . . 56
5.12 Lateral deviation for the sharp turn scenario. 58
5.13 Relative yaw angle for the sharp turn scenario. 58
5.14 Steering angle input over time for the sharp turn scenario. 59
5.15 Trajectory visualization for the city-like road scenario. 61
5.16 Real-time modeling error for lateral velocity on the city-like road scenario. 62
5.17 Real-time modeling error for yaw rate on the city-like road scenario. . 63
5.18 Lateral deviation for the city-like road scenario. 64
5.19 Relative yaw angle for the city-like road scenario. 65
5.20 Steering angle input over time for the city-like road scenario. 65
5.21 Lateral deviation for the sharp turn scenario using Online DMD and RLS. 67
5.22 Yaw angle deviation for the sharp turn scenario using Online DMD and RLS. 67
5.23 Steering angle input for the sharp turn scenario using Online DMD and RLS. 68

List of Tables

3.1 Evaluation metrics for lateral velocity. 33
3.2 Evaluation metrics for yaw rate. 33

5.1 Simulation Parameters for Vehicle Dynamics 43
5.2 Upper-Level Controller Parameters 44
5.3 Adaptive MPC Parameters 45
5.4 LQR Parameters 45
5.5 Modeling errors for the constant curvature road scenario. 51
5.6 Lateral deviation and yaw angle errors for the constant curvature road. 53
5.7 Modeling errors for the sharp turn scenario. 57
5.8 Lateral deviation and yaw angle errors for the sharp turn scenario. . . 59
5.9 Modeling errors for the city-like road scenario. 63
5.10 Lateral deviation and yaw angle errors for the city-like road. 65
5.11 Lateral and yaw angle errors for the sharp turn scenario using Online DMD and RLS. 68

1

Introduction

Autonomous vehicle technology has advanced rapidly, creating opportunities to enhance road safety, reduce traffic congestion, and improve overall efficiency. These benefits rely heavily on the integration of advanced systems that automate various driving tasks. The Society of Automotive Engineers (SAE) defines six levels of driving automation, ranging from no automation (Level 0) to full automation (Level 5), as depicted in Figure 1.1. This classification has become a global standard, offering clear guidelines for the development and deployment of automated driving systems. As vehicles progress toward higher levels of automation, the need for sophisticated systems to manage dynamic driving tasks becomes increasingly critical.

Modern vehicles are equipped with numerous advanced driver assistance systems (ADAS), each designed to enhance safety and convenience. Examples include Adaptive Cruise Control (ACC), which maintains a safe distance from the vehicle ahead, and Autonomous Emergency Braking (AEB), which reduces collision risks by applying brakes automatically. Lane-Keeping Assist (LKA) is another vital function, helping vehicles stay within their designated lanes by detecting lane markings and intervening when needed. These systems represent critical steps toward achieving higher levels of driving automation and play a significant role in ensuring vehicle stability and safety.

Among the various functions of ADAS, maintaining lane position accurately and consistently, especially in dynamic and uncertain environments, remains a key challenge. Lane-Keeping Assist systems must operate under diverse conditions, including varying road geometries, speeds, and external disturbances. This necessitates the development of robust control algorithms and adaptive vehicle models capable of real-time adjustments.

The work presented in this thesis addresses critical challenges in the development of motion control systems for autonomous vehicles, particularly in lane-keeping appli-

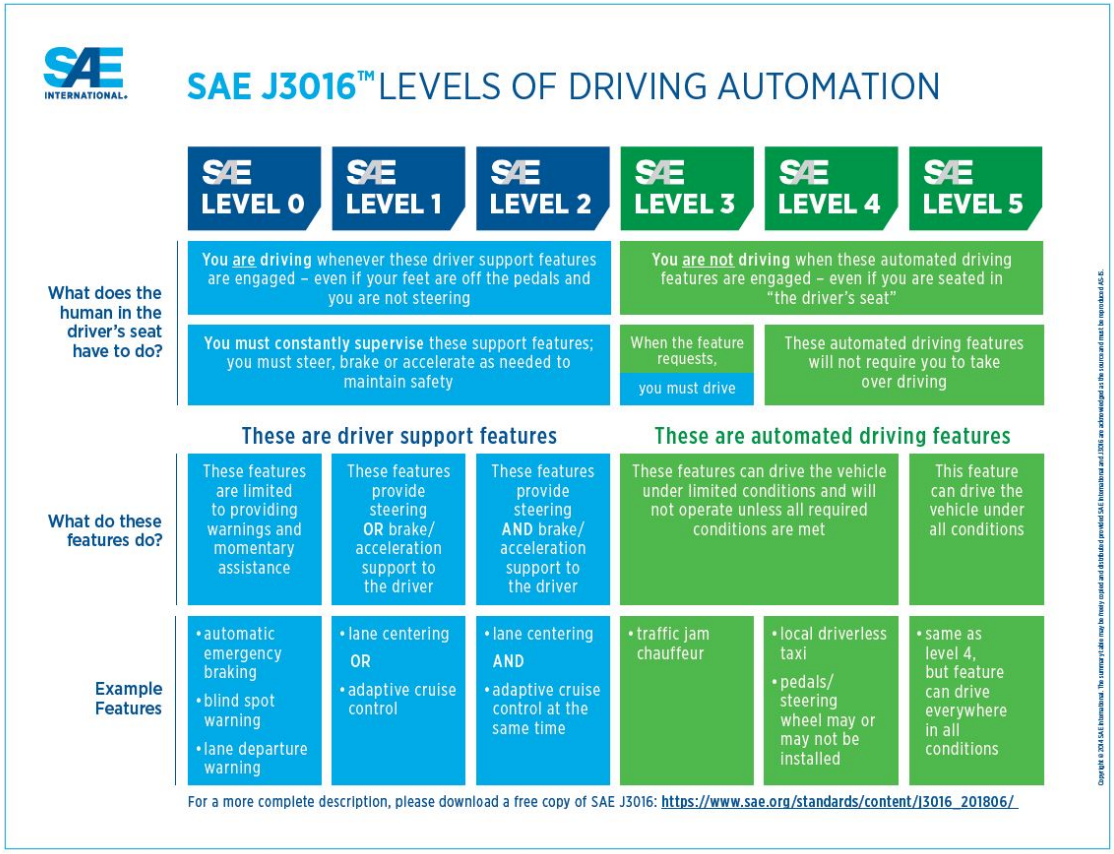


Figure 1.1: SAE J3016 Levels of Driving Automation (SAE, 2016) [47].

cations. From a control perspective, the primary difficulties stem from the nonlinear and complex dynamic characteristics of vehicles, combined with uncertainties concluding in their models. First, the mechanism of tire force generation is highly nonlinear, influenced by factors such as tire wear, tire design environmental conditions and many other factors that can change with tire design. While significant research efforts (e.g., [40]) have been dedicated to creating accurate models, these models often overfit to specific scenarios and fail to generalize to external factors such as temperature, road surface variations, vehicle speeds and tire age. Second, the suspension and tire dynamics are difficult to incorporate fully into real-time control designs. As a result, simplifications are frequently made, which introduce uncertainties into the representation of vehicle dynamics.

This thesis investigates the potential of data-driven modeling approaches and adaptive control strategies to address these challenges. By focusing on techniques that enable real-time adaptability and robustness, the work aims to provide solutions that ensure reliable performance across diverse operating conditions. These considerations reflect the tight requirements for safety and reliability in autonomous vehicle control systems.

1.1 Motivation

Ensuring safe, efficient, and reliable transportation is a fundamental challenge in the modern world, driven by the increasing demands of economic growth and population expansion. As the global population is projected to surpass 10 billion in the coming decades [54], the movement of people and goods must become more sustainable, energy-efficient, and safe. Autonomous vehicle technologies have emerged as a promising solution to address these demands, leveraging advancements in automation, connectivity, and electrification to overcome the limitations of traditional transportation systems [12].

Among the many functions of autonomous vehicles, Lane Keeping (LK) systems play a critical role in maintaining a vehicle's position within its designated lane. These systems are essential for ensuring passenger safety and compliance with traffic regulations, particularly as autonomous vehicles progress toward higher levels of automation. However, achieving reliable and robust LK performance in diverse and dynamic conditions remains a significant challenge.

LK systems must operate under complex scenarios, including sharp road curvatures, varying road surfaces such as asphalt and wet roads, and external factors like wind, rain, and vehicle dynamics changes. Traditional approaches to LK often rely on fixed, pre-defined vehicle models and control strategies, which lack the adaptability to respond to real-time changes [13]. This limitation restricts their effectiveness, especially during high-speed maneuvers or in the presence of unpredictable disturbances.

Modern advancements in sensing and computational technologies enable LK systems to incorporate real-time data, significantly improving their adaptability and responsiveness. Data-driven modeling techniques offer a promising avenue for capturing the dynamic behaviors of vehicles with greater accuracy. These methods allow LK systems to adapt to variations in tire conditions, road geometries, and vehicle dynamics, ensuring a higher degree of reliability. Furthermore, integrating adaptive control strategies with these models addresses the inherent nonlinear and coupled dynamics of vehicle control, facilitating more stable and robust performance.

Despite these advancements, significant challenges remain. Vehicle dynamics are inherently complex and subject to uncertainties arising from factors such as tire wear, road surface variations, and unmodeled environmental conditions [40]. Control strategies must address these challenges while maintaining computational efficiency for real-time implementation. Moreover, external disturbances like wind forces and abrupt road changes can destabilize the system, necessitating robust adaptive solu-

tions to mitigate these effects.

The combination of increasing demands for safe and efficient transportation, the complexity of vehicle dynamics, and the limitations of traditional approaches highlight the need for innovative solutions. This work addresses these challenges by exploring data-driven modeling and adaptive control strategies, aiming to develop a robust and reliable Lane Keeping system. By overcoming the limitations of conventional methods, this research contributes to the broader goal of enabling safe, efficient, and sustainable autonomous vehicle systems capable of navigating diverse and dynamic environments.

1.2 Contributions of the Thesis

The contributions of this thesis are summarized as follows:

- A high-fidelity simulation environment is developed using MATLAB's Vehicle Dynamics Blockset and Automated Driving Toolbox, incorporating sensors such as cameras and radar.
- Comparative analysis of lateral vehicle dynamics are performed by simulating:
  - High-Fidelity Simulator
  - Linear vehicle model,
  - Online Data-driven vehicle model.
- An Online Dynamic Mode Decomposition framework is developed and validated using high-fidelity simulator.
- Adaptive control strategies are implemented and tested:
  - An Adaptive Model Predictive Controller is constructed and evaluated,
  - A Linear Quadratic Regulator is developed for comparative analysis.
- Various challenging scenarios are simulated to evaluate the proposed approach such as varying road surfaces, sharp curvature roads and different vehicle speeds.

1.3 Outline of the Thesis

This thesis is organized into five chapters. These chapters provide the following information:

Chapter 2 provides a literature survey and background on vehicle dynamics, online system identification techniques, path-tracking controllers and control strategies for autonomous vehicle systems.

Chapter 3 explains lateral vehicle dynamics modeling using Online Dynamic Mode Decomposition, including the integration of system identification techniques and their role in capturing vehicle behavior.

Chapter 4 details the design and implementation of control strategies, including Linear Quadratic Regulator and Adaptive Model Predictive Control, leveraging the models developed with Online Dynamic Mode Decomposition.

Chapter 5 presents a detailed discussion of simulation results, including various test scenarios such as varying road surfaces, sharp curvatures, and external disturbances. The chapter evaluates the performance of the proposed control strategies and compares them against baseline models.

Finally, Chapter 6 provides concluding remarks and highlights potential directions for future research.

2

Literature Review and Background

2.1 Vehicle Dynamics

This section introduces the vehicle dynamics models used in this work, focusing on the road-aligned representation of vehicle motion. The single-track model, also known as the bicycle model, is employed due to its simplicity and effectiveness in capturing essential lateral and longitudinal dynamics. By aligning the dynamics with the road, this approach ensures a practical and computationally efficient foundation for further analysis and control design.

2.1.1 Coordinate System

Vehicle dynamics can be represented using different coordinate systems, either in the inertial frame or the body-centered frame. In this work, the Cartesian coordinate system, as illustrated in Figure 2.1, is employed to describe vehicle motion relative to the road. This system allows the position and orientation of the vehicle to be represented using X-Y coordinates, with the longitudinal motion aligned along the X-axis and the lateral motion along the Y-axis.

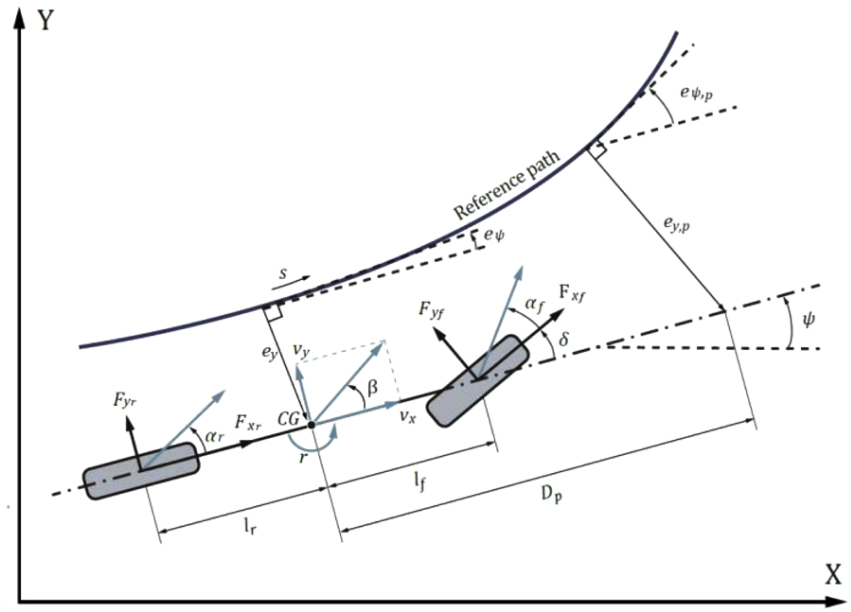


Figure 2.1: Vehicle dynamics coordinate system in a road-aligned frame [51].

This approach provides a clear framework for analyzing kinematic and dynamic relationships, which are essential for understanding vehicle behavior. Throughout this thesis, vehicle dynamics will be modeled in a road-aligned frame, ensuring consistency and practicality for controller development and road geometry inclusion.

2.1.2 Single-Track Bicycle Model

The single-track bicycle model simplifies the vehicle dynamics by combining the left and right wheels into single equivalent wheels at the front and rear axles. This model effectively captures the lateral and yaw dynamics of the vehicle while assuming a rigid body structure and small slip angles.

Dynamic Equations

The equations of motion for the single-track bicycle model are derived using Newton's second law, with the dynamics expressed in terms of lateral velocity (v_y), yaw rate (r), and longitudinal velocity (v_x):

$$m\left(\dot{v}_y + v_x r\right) = F_{yf} + F_{yr},$$
$$I_z \dot{r} = l_f F_{yf} - l_r F_{yr},$$

where:

- m: Vehicle mass,

- I_z: Yaw moment of inertia,
- l_f, l_r: Distances from the center of gravity to the front and rear axles, respectively,
- F_{yf}, F_{yr}: Lateral tire forces at the front and rear axles.

The model assumes a constant longitudinal velocity (v_x), simplifying the dynamics to focus on lateral and yaw motion.

2.1.3 Linear Tire Model

The Linear Tire Model assumes a proportional relationship between the lateral tire force and the slip angle, which is valid for small slip angles. This model is widely used due to its simplicity and computational efficiency, though it is less accurate under high-slip conditions or at high speeds [43].

The slip angle α is defined as the angle between the orientation of the tire and the direction of the velocity vector of the wheel. For the front and rear tires, the slip angles are given by:

$$\alpha_f = \delta - \theta_{vf} \tag{2.1}$$

$$\alpha_r = -\theta_{vr} \tag{2.2}$$

where:

- α_f: Slip angle of the front tire.
- α_r: Slip angle of the rear tire.
- δ: Steering angle of the front wheels.
- θ_{vf}: Angle of the front tire's velocity vector with respect to the vehicle's longitudinal axis.
- θ_{vr}: Angle of the rear tire's velocity vector with respect to the vehicle's longitudinal axis.

Using the slip angles, the lateral forces for the front and rear tires are expressed as:

$$F_{yf} = 2C_{\alpha f}(\delta - \theta_{vf}) \tag{2.3}$$

$$F_{yr} = 2C_{\alpha r}(-\theta_{vr}) \tag{2.4}$$

where:

- F_{yf}: Lateral force of the front tires.

- F_{yr}: Lateral force of the rear tires.
- $C_{\alpha f}$: Cornering stiffness of each front tire.
- $C_{\alpha r}$: Cornering stiffness of each rear tire.

The angles θ_{vf} and θ_{vr} represent the angles that the front and rear velocity vectors make with the vehicle's longitudinal axis. These are given by:

$$\tan(\theta_{vf}) = \frac{v_y + l_f\dot{\psi}}{v_x} \tag{2.5}$$

$$\tan(\theta_{vr}) = \frac{v_y - l_r\dot{\psi}}{v_x} \tag{2.6}$$

Using a small-angle approximation, we can simplify these equations as follows:

$$\theta_{vf} \approx \frac{\dot{v}_y + l_f\dot{\psi}}{v_x} \tag{2.7}$$

$$\theta_{vr} \approx \frac{\dot{v}_y - l_r\dot{\psi}}{v_x} \tag{2.8}$$

These approximations allow for more efficient computations in real-time applications where small slip angles are assumed, providing a balance between accuracy and computational simplicity.

2.1.4 Model Assumptions and Limitations

The single-track bicycle model with a linear tire approximation is valid for small slip angles and normal driving conditions. While it provides a computationally efficient framework, it does not capture nonlinear tire dynamics, suspension effects, or transient behaviors at high slip angles or extreme conditions. These limitations are addressed in this work by incorporating data-driven modeling techniques, as detailed in subsequent sections.

2.2 Dynamic Mode Decomposition

In recent years, Dynamic Mode Decomposition (DMD) has emerged as a powerful data-driven technique for modeling complex systems, particularly in the context of control systems[53]. This part discusses DMD extensions and some applications that are in the literature.

2.2.1 Dynamic Mode Decomposition

Dynamic Mode Decomposition (DMD) is a data-driven method used to analyze and model the dynamics of complex systems based on time-series data [53]. It provides a mathematical framework for decomposing high-dimensional data into spatiotemporal modes, which represent the dominant dynamic patterns of the system. This decomposition enables the identification of key behaviors, such as oscillatory modes or transient dynamics, without requiring prior knowledge of the governing equations.

The central idea of DMD is to approximate the dynamics of a system as a linear map between consecutive states. Consider a set of snapshots of the system states, $\{x_1, x_2, \ldots, x_N\}$, where each x_k represents the system state at time t_k. DMD seeks to approximate the evolution of these states as:

$$x_{k+1} = Ax_k, \tag{2.9}$$

where $A \in \mathbb{R}^{n_x \times n_x}$ is the system matrix. Using Singular Value Decomposition (SVD), the DMD algorithm identifies a low-rank approximation of A based on the data, extracting the dominant eigenvalues and eigenvectors that describe the system's behavior.

DMD has been widely applied in fluid dynamics, robotics, and control systems for analyzing high-dimensional datasets and uncovering low-dimensional representations of complex dynamics. Its simplicity and computational efficiency make it a powerful tool for system identification and analysis in a variety of fields [57].

2.2.2 Dynamic Mode Decomposition with Control

The concept of DMD was first extended to include control inputs in the work by Proctor, Brunton, and Kutz (2016) [41]. Their paper, *"Dynamic Mode Decomposition with Control"*, presents a framework for incorporating control inputs into the DMD algorithm, thereby enhancing its applicability for control systems.

Dynamic Mode Decomposition with Control (DMDc) extends the traditional DMD approach by including control inputs in the state-space representation [41]. The methodology involves collecting time-series data of system states and control inputs, then applying Singular Value Decomposition (SVD) to decompose the data matrix into modes that capture the system's dynamics. The DMDc algorithm identifies a linear model that approximates the system's behavior, including the effects of control actions.

The primary advantage of DMDc is its ability to model complex, nonlinear systems

using linear techniques. This makes it highly suitable for real-time control applications, where computational efficiency and simplicity are crucial. By incorporating control inputs, DMDc provides a more accurate representation of the system's response to external inputs, enabling the design of effective control strategies [41].

DMDc is a data-driven technique used to derive a linear model that represents the system's dynamics, including the effects of control inputs [41]. Unlike traditional methods, it does not require prior knowledge of the system dynamics or parameters, making it a powerful tool for modeling complex systems.

DMDc assumes that the system is linear, as represented by:

$$x_{k+1} = Ax_k + Bu_k \tag{2.10}$$

where $x_k \in \mathbb{R}^{n_x}$ represents the system state, $u_k \in \mathbb{R}^{n_u}$ represents the input, and $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$ are the system matrix and input matrix, respectively. DMDc aims to discover a linear model described by this equation by processing the system data with a one-step difference [41].

The data matrices are constructed as follows:

$$X' = AX + B\Gamma = \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} X \\ \Gamma \end{bmatrix} = G\Omega \tag{2.11}$$

where

$$X = \begin{bmatrix} x_1 & x_2 & \cdots & x_{N-1} \end{bmatrix} \in \mathbb{R}^{n_x \times N-1} \tag{2.12}$$

$$X' = \begin{bmatrix} x_2 & x_3 & \cdots & x_N \end{bmatrix} \in \mathbb{R}^{n_x \times N-1} \tag{2.13}$$

$$\Gamma = \begin{bmatrix} u_1 & u_2 & \cdots & u_{N-1} \end{bmatrix} \in \mathbb{R}^{n_u \times N-1} \tag{2.14}$$

By analyzing the relationship between the state X of the previous time step, the input variable Γ, and the subsequent state variable X' through singular value decomposition, a linearized system G is derived as shown in the following equations:

$$G\Omega = X' \tag{2.15}$$

$$\text{SVD}(\Omega) = U\Sigma V^T \tag{2.16}$$

$$G = X'V\Sigma^{-1}U^T = \begin{bmatrix}\hat{A} & \hat{B}\end{bmatrix} \tag{2.17}$$

where $\hat{A}$ and $\hat{B}$ are the estimated system and input matrices. The matrices $\hat{A}$ and $\hat{B}$ were derived using DMD solely based on the system states and inputs, without prior knowledge of the system dynamics and parameters [53]. This highlights the strength of DMDc in modeling complex systems purely from data, making it an effective tool for system identification and control design [41]. In this approach, the key steps include:

- 1. Collecting time-series data of system states and inputs.
- 2. Constructing data matrices X, X', and Γ.
- 3. Applying SVD to the combined data matrix Ω.
- 4. Deriving the linear model matrices $\hat{A}$ and $\hat{B}$.

2.2.3 Extended Dynamic Mode Decomposition

The extended DMD algorithm (eDMD) is fundamentally similar to the standard DMD algorithm [53, 7], but instead of performing regression on direct measurements of the state, it performs regression on an augmented vector that contains nonlinear measurements of the state. This augmented state allows eDMD to capture more complex, nonlinear dynamics by projecting the data into a higher-dimensional space, enabling a more accurate representation of the system.

The development of eDMD draws from the variational approach of conformation dynamics, pioneered by Noé and Nuske in 2013 [38]. This approach allows eDMD to approximate the Koopman operator, which provides a linear description of nonlinear systems by capturing how observables evolve over time.

In eDMD, an augmented state vector y is constructed by applying a function $\theta(x)$ to the original state vector x:

$$y = \theta(x) = \begin{bmatrix}\theta_1(x) \\ \theta_2(x) \\ \vdots \\ \theta_p(x)\end{bmatrix} \tag{2.18}$$

where the function θ may include the original state x along with additional nonlinear measurements. Typically, $p \geq n$, meaning the augmented state y has more dimensions than the original state [7].

Two data matrices are then constructed from this augmented state, similar to the approach in DMD:

$$Y = \begin{bmatrix} y_1 & y_2 & \cdots & y_m \end{bmatrix}, \quad Y' = \begin{bmatrix} y_2 & y_3 & \cdots & y_{m+1} \end{bmatrix}. \tag{2.19}$$

A best-fit linear operator A_Y is then identified to map Y to Y', achieving a least-squares solution:

$$A_Y = \arg\min_{A_Y} \|Y' - A_Y Y\| = Y'Y^{\dagger}, \tag{2.20}$$

where $Y^{\dagger}$ denotes the pseudoinverse of Y.

This regression can also be expressed in terms of data matrices $\theta(X)$ and $\theta(X')$:

$$A_Y = \arg\min_{A_Y} \|\theta(X') - A_Y \theta(X)\| = \theta(X')\theta(X)^{\dagger}. \tag{2.21}$$

Because the augmented vector y may be significantly larger than the original state x, kernel methods are often used to efficiently compute the regression. This enriched library of observables θ creates a larger basis in which to approximate the Koopman operator [57, 7].

In theory, with an infinite number of snapshots, the extended DMD operator converges to the Koopman operator projected onto the subspace spanned by θ [57]. However, if θ does not span a Koopman invariant subspace, the projected operator may include spurious eigenvalues and eigenvectors that do not accurately represent the true system dynamics.

Thus, it is essential to use validation techniques to ensure that eDMD models are not overfit. For instance, eDMD cannot represent systems with multiple fixed points or periodic orbits if the original state x is included as a measurement. Such systems require a model that captures their nonlinear attractors, which cannot be represented by a finite-dimensional linear system [7].

2.2.4 Physics-Informed Dynamic Mode Decomposition

Physics-informed machine learning (piML) integrates known physical principles into ML algorithms by embedding specific biases that align with these principles. There are generally three approaches to introducing these biases: through data augmentation techniques, by embedding physical constraints into the loss function, or by

embedding mathematical constraints directly into the model architecture. The last approach is considered the most robust, as it produces models that inherently satisfy physical laws [24]. Physics-Informed DMD (piDMD) aligns with this third approach, constraining the matrix manifold of the DMD regression problem to incorporate specific physical principles.

piDMD diverges from standard DMD by favoring matrix structures tailored to the physical problem rather than merely seeking a low-rank approximation of the data matrix. For instance, piDMD can incorporate structures such as circulant, symmetric, or upper-triangular matrices to capture unique physical properties relevant to specific applications.

1. **Modelling:** The process begins with identifying known or expected physical properties of the system. This involves summarizing any pre-existing knowledge about the system's behavior and dynamics.
2. **Interpretation:** The next step is to translate these physical properties into mathematical constraints on the matrix manifold, which may reflect spatial or temporal characteristics of the data collection process.
3. **Optimization:** Using the defined matrix manifold, piDMD solves the corresponding constrained regression problem. The optimization process is adjusted to ensure that the resulting model complies with the physical principles identified in the modeling phase.
4. **Diagnostics:** The final stage involves deriving physical insights from the resulting model matrix. This can include analyzing eigenvalues, decomposing the data into modal structures, or using the piDMD model to make predictions, assessing how well it adheres to the specified physical constraints.

Further details on these steps are provided in the supplementary materials [1]. For instance, Figure 2.2 demonstrates the application of piDMD to a traveling wave system.

By incorporating physical constraints, piDMD introduces a balance between accurate data reconstruction and model generalizability. Unlike standard DMD, which focuses on achieving a low-rank data approximation, piDMD sacrifices some degrees of freedom to impose a global physical structure on the learned model. This regularization enables piDMD to better capture the true dynamics of the underlying system when applied to unseen test data.

Some matrix structures used in piDMD, such as symmetric, tridiagonal, or circulant matrices, are particularly suited to systems with specific properties and offer efficient computation benefits, like fast eigendecomposition. For applications where physical

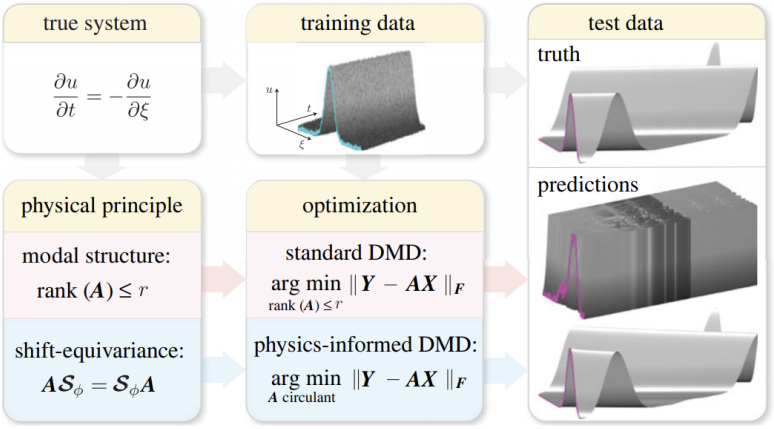


Figure 2.2: piDMD: An example application to a traveling wave system, where shift-equivariant properties guide the choice of a circulant matrix model. Adapted from [3].

knowledge is available, piDMD can significantly enhance both the interpretability and accuracy of the DMD model.

2.2.5 Online DMD Algorithm

The Online DMD algorithm extends the traditional DMD by updating the matrix $\mathbf{A}_k$ in real-time as new data becomes available. This allows the algorithm to capture time-varying dynamics efficiently without the need for recomputation over the full dataset [60]. Let $\mathbf{X}_k = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k]$ and $\mathbf{Y}_k = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k]$ represent the input and output snapshots, respectively, where each pair $(\mathbf{x}_i, \mathbf{y}_i)$ satisfies $\mathbf{y}_i \approx \mathbf{A}_k \mathbf{x}_i$.

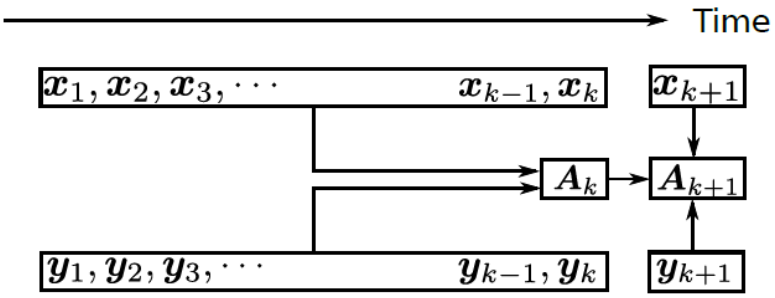


Figure 2.3: Online DMD: $\mathbf{A}_k$ is updated with each new snapshot pair $(\mathbf{x}_{k+1}, \mathbf{y}_{k+1})$, adapting to time-varying dynamics.

The DMD matrix $\mathbf{A}_k$ is updated at each time step based on the following recursive formula:

$$\mathbf{A}_{k+1} = \mathbf{A}_k + \gamma_{k+1}(\mathbf{y}_{k+1} - \mathbf{A}_k \mathbf{x}_{k+1})\mathbf{x}_{k+1}^\top \mathbf{P}_k, \tag{2.22}$$

where: - γ_{k+1} is a weighting factor that determines how much the new data point

influences the update, given by:

$$\gamma_{k+1} = \frac{1}{1 + \mathbf{x}_{k+1}^{\top} \mathbf{P}_k \mathbf{x}_{k+1}}, \tag{2.23}$$

- $\mathbf{P}_k$ is an auxiliary matrix updated recursively as:

$$\mathbf{P}_{k+1} = \mathbf{P}_k - \gamma_{k+1} \mathbf{P}_k \mathbf{x}_{k+1} \mathbf{x}_{k+1}^{\top} \mathbf{P}_k. \tag{2.24}$$

Windowed DMD Algorithm

The Windowed DMD algorithm limits its model to only the most recent snapshots within a defined "window" size, allowing it to adapt to systems with changing dynamics by ignoring older snapshots.

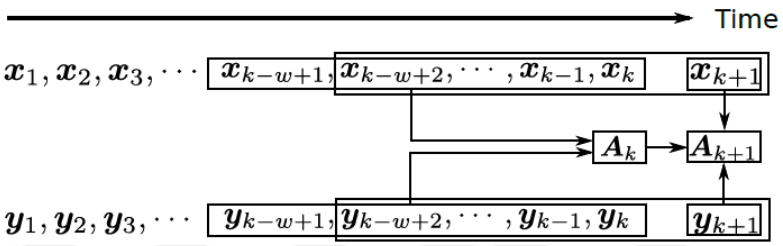


Figure 2.4: Windowed DMD: $\mathbf{A}_k$ depends only on the most recent w snapshots, discarding older data as new snapshots arrive.

In the windowed approach, $\mathbf{A}_k$ depends only on the w most recent snapshots. When a new snapshot $(\mathbf{x}_{k+1}, \mathbf{y}_{k+1})$ is obtained, the window shifts to include this new snapshot, while the oldest snapshot is removed. This method provides adaptability to recent data without being influenced by outdated information, making it well-suited for applications with evolving dynamics.

Initialization

To initialize the algorithm, we compute the initial DMD matrix $\mathbf{A}_0$ and $\mathbf{P}_0$. A common approach is to perform a batch DMD using the first set of available snapshots:

$$\mathbf{A}_0 = \mathbf{Y}_0 \mathbf{X}_0^{+} \quad \text{and} \quad \mathbf{P}_0 = (\mathbf{X}_0 \mathbf{X}_0^{\top})^{-1},$$

where $\mathbf{X}_0^{+}$ is the Moore-Penrose pseudoinverse of $\mathbf{X}_0$. Alternatively, $\mathbf{A}_0$ can be initialized as $\mathbf{A}_0 = 0$ and $\mathbf{P}_0$ can be set to $\alpha \mathbf{I}$, where α is a large scalar and $\mathbf{I}$ is the identity matrix.

Low-Rank Updates

The recursive update for $\mathbf{A}_k$ ensures that the system adapts to new data points without having to recompute the entire matrix. The term $\mathbf{y}_{k+1} - \mathbf{A}_k\mathbf{x}_{k+1}$ represents the prediction error based on the current estimate of $\mathbf{A}_k$, and the update reduces this error by incorporating information from $\mathbf{x}_{k+1}$ and $\mathbf{y}_{k+1}$.

The update of $\mathbf{P}_k$ is a rank-1 modification, making the computation efficient. The complexity of updating $\mathbf{A}_k$ and $\mathbf{P}_k$ is $O(n^2)$, where n is the dimension of the state vector, compared to $O(kn^2)$ for traditional DMD that involves k snapshots.

Adaptation with Streaming Data

As more snapshots $(\mathbf{x}_{k+1}, \mathbf{y}_{k+1})$ are obtained, the Online DMD algorithm adapts in real time to the evolving dynamics of the system. By tuning the weighting factor γ_{k+1}, we can control the extent to which older data is forgotten, making the algorithm suitable for time-varying systems. This is a similar method to Recursive Least Squares algorithm.

This recursive update makes Online DMD highly computationally efficient and ideal for real-time applications such as system identification and control in dynamic environments.

2.3 Recursive Least Squares and its Comparison with Online DMD

Recursive Least Squares (RLS) is a parameter estimation technique widely used in adaptive control and system identification for time-varying systems. The algorithm estimates model parameters by minimizing the sum of squared errors between the predicted and actual outputs using real-time data. Unlike data-driven methods like Online DMD, RLS requires a predefined model structure and assumes that the system can be described using a set of equations [23].

2.3.1 Overview of RLS Algorithm

The RLS algorithm operates by recursively updating parameter estimates as new data becomes available. It is particularly effective for systems with slowly changing dynamics. The key equations of the RLS algorithm are as follows:

- Define the system model:

$$y_k = \phi_k^T \theta_k + \epsilon_k$$

where y_k is the output, ϕ_k is the regressor vector (e.g., past states and inputs), θ_k is the parameter vector to be estimated, and ϵ_k is the prediction error.

- Update the prediction error:

$$\epsilon_k = y_k - \phi_k^T \theta_{k-1}$$

- Update the gain:

$$K_k = \frac{P_{k-1}\phi_k}{\lambda + \phi_k^T P_{k-1}\phi_k}$$

where P_{k-1} is the covariance matrix and λ is the forgetting factor ($0 < \lambda \leq 1$).

- Update the parameter vector:

$$\theta_k = \theta_{k-1} + K_k \epsilon_k$$

- Update the covariance matrix:

$$P_k = \frac{1}{\lambda}\left(P_{k-1} - K_k \phi_k^T P_{k-1}\right)$$

In the context of this thesis, Recursive Least Squares (RLS) and Online DMD are compared due to their similar recursive nature in identifying system dynamics.

2.3.2 Comparison with Online DMD

Recursive Least Squares and Online Dynamic Mode Decomposition are both recursive methods designed for real-time system identification, making them suitable for adaptive control applications. Both approaches share the following key characteristic:

- **Recursive Nature:** Both RLS and Online DMD update their estimates iteratively in real-time to adapt to changing system dynamics. However, their recursive mechanisms differ: RLS performs parameter estimation by minimizing the prediction error, while Online DMD minimizes the Frobenius norm to reconstruct system matrices using eigenvalue-based computations.

Despite their similarities in structure, these methods differ in their approach:

- RLS assumes a predefined model structure and estimates the parameters of this model. This makes it more structured but less flexible in handling unknown or highly complex dynamics.
- Online DMD, on the other hand, does not rely on any predefined model.

It directly reconstructs the system dynamics from the state measurements, offering greater flexibility in data-driven scenarios.

This comparison is particularly relevant as both methods aim to solve similar problems but use different approaches. By examining their performance, this study highlights the advantages and trade-offs between model-based and data-driven methods.

2.4 Lane-Keeping System Control Strategies

The ability to accurately track a desired path within a lane is a fundamental requirement for autonomous vehicle (AV) systems, ensuring safe and efficient navigation in dynamic traffic environments. Lane-keeping systems focus on generating control inputs that maintain the vehicle's lateral position within the lane boundaries while considering vehicle stability, passenger comfort, and road constraints. As AV technology advances, robust and adaptive lane-keeping strategies are increasingly necessary to address uncertainties such as varying road surfaces, environmental conditions, and vehicle dynamics.

Over the years, numerous control strategies have been proposed to tackle the challenges associated with lane-keeping in AVs. These approaches aim to provide precise lateral control, accommodating the nonlinear and dynamic behavior of vehicles. This section reviews the evolution of lane-keeping control strategies, emphasizing their relevance to modern autonomous driving applications. The discussion sets the stage for identifying the most effective methods for the objectives of this thesis.

2.4.1 Geometry-Based Controllers

Pure Pursuit Controller

The Pure Pursuit Controller (PPC), initially developed for missile guidance [46], was adapted for vehicle control to maintain lane position by using road imagery [56]. PPC selects a target point at a fixed "look-ahead distance" on the reference path and calculates the steering angle by fitting an arc connecting the rear wheel to this point [10].

PPC is known for its simplicity, computational efficiency, and effectiveness at low speeds, as demonstrated in the DARPA Grand Challenge [9]. While it performs well in scenarios with small lateral errors [44], its accuracy diminishes at higher speeds, requiring adjustments to the look-ahead distance for improved stability [39]. Despite its limitations, PPC remains a foundational approach for geometry-based vehicle control.

Stanley Controller

The Stanley controller, used by the 2005 DARPA Grand Challenge winner [52], computes steering actions based on the lateral error (e_l) and the angular error (α) between the vehicle's heading and the reference trajectory [17]. Unlike PPC, it does not rely on a predefined look-ahead distance, making it more flexible.

The controller uses a tunable gain (k) to balance responsiveness and stability [49], and practical modifications ensure the steering angle respects physical limits [17]. Adaptive techniques, such as gain optimization through Particle Swarm Optimization and Fuzzy Logic Systems, further enhance performance under varying conditions [2].

The Stanley controller is robust and computationally efficient, making it a preferred choice for handling diverse path tracking scenarios in autonomous vehicles.

2.4.2 Robust Control

Robust control strategies are essential in autonomous vehicle path tracking, addressing uncertainties in simplified models and external disturbances while balancing accuracy and computational feasibility [48, 28]. Two prominent robust control approaches in this domain are Sliding Mode Control (SMC) and H_∞ control.

Sliding Mode Control (SMC)

SMC ensures robustness by driving system states to a predefined sliding surface in state-space, simplifying the dynamics once on the surface [48, 28]. This method effectively handles uncertainties and disturbances, but its implementation can lead to chattering—a high-frequency oscillation caused by discontinuous control actions. Techniques like the Boundary Layer Method [11] and second-order SMC [16, 59] address this issue. Adaptive strategies using fuzzy logic and neural networks further enhance SMC by dynamically adjusting to uncertainty bounds [11].

H_∞ Control

H_∞ control frames the control problem as an optimization task, ensuring resilience to parametric uncertainties and disturbances [19]. This approach is particularly effective for nonlinearities like backlash-type hysteresis in steering systems. Integrating H_∞ control with fuzzy logic systems enhances its ability to manage complex vehicle dynamics [18, 60]. Observer-based H_∞ controllers, using Takagi–Sugeno fuzzy models, offer robust representation of nonlinear behaviors [37].

Both SMC and H_∞ control provide reliable performance in uncertain and dynamic

conditions, solidifying their role in the robust control of autonomous vehicle path tracking systems.

2.4.3 Optimization-Based Controllers

Linear Quadratic Regulator (LQR)

The Linear Quadratic Regulator (LQR) is a widely utilized method in path tracking control due to its capability to achieve an optimal trade-off between tracking accuracy and control effort. Based on a linearized vehicle model, such as the bicycle model, LQR minimizes a predefined quadratic cost function by solving the Riccati equation to derive the optimal feedback gain [36]. This approach ensures effective mitigation of tracking errors and control effort.

LQR employs an error-based state-space model that incorporates parameters like cross-track error and heading angle error, enabling accurate reference trajectory tracking [35, 42, 29]. Path curvature is often treated as an external disturbance, enhancing the controller's robustness to parameter uncertainties and unmodeled dynamics [35]. However, steady-state errors can arise when tracking paths with significant curvature [49]. A feedforward component is frequently added to the LQR framework to compensate for curvature-induced disturbances, improving its performance [29].

At higher speeds, model uncertainties become more pronounced, limiting LQR's effectiveness. Nevertheless, its robustness and computational efficiency make it a reliable strategy for path tracking in controlled environments, where disturbances and uncertainties are moderate.

Model Predictive Control (MPC)

Model Predictive Control (MPC) predicts future system states over a defined time horizon using a system model and optimizes control sequences to minimize a cost function. This iterative process enables MPC to effectively handle multi-variable systems and constraints, making it highly applicable to autonomous vehicle path tracking. However, its real-time computational demands present challenges, particularly in dynamic and high-speed scenarios [14, 34].

MPC can be implemented using either nonlinear models, which capture system dynamics with high accuracy but are computationally intensive and often restricted to low-speed applications [6, 27], or linear approximations, which linearize the system around the current operating point. The latter approach strikes a balance between computational efficiency and accuracy, making it more suitable for real-time applica-

tions at higher speeds [26, 15]. Advanced techniques, such as the Linear Parameter Varying (LPV) approach, adapt model parameters dynamically to reduce computational demands without compromising accuracy [25]. Additionally, adaptive and switching MPC frameworks further enhance performance by adjusting prediction horizons and model fidelity based on current operating conditions [5].

While computational challenges remain, methods such as explicit MPC and machine learning-based strategies aim to mitigate real-time processing requirements [58]. These innovations extend MPC's applicability, reinforcing its position as a versatile and effective control strategy for path tracking in autonomous vehicles.

2.4.4 Data-Driven Controllers

Data-driven approaches leverage system identification techniques to model vehicle dynamics from observed data, enabling adaptability to changing conditions. Various methods have been explored in the literature:

N4SID has been applied for estimating linear state-space models to identify vehicle parameters like mass and cornering stiffness, aiding adaptive control [55]. LFT modeling captures parametric uncertainties in tire dynamics for robust lateral control [4]. Neural networks have been used to learn nonlinear vehicle dynamics directly from data, providing real-time adaptability to road conditions [50].

DMDc integrates control inputs into linear system identification, supporting applications like lane keeping and trajectory tracking [30]. Similarly, Koopman Operators approximate nonlinear dynamics in a lifted linear space, enabling precise control for tasks like lane keeping [31].

These methods highlight the effectiveness of data-driven modeling in addressing real-world complexities in autonomous vehicle control.

3

Lateral Vehicle Dynamics with Online Dynamic Mode Decomposition

3.1 Modeling the Vehicle Dynamics System

Model-based controllers rely on accurate system models to quantify prediction errors and optimize control actions [45]. Developing such models often involves physics-based dynamics [8], but in this work, we employ the Online Dynamic Mode Decomposition (Online DMD) algorithm, which identifies the model using only time-series state and input data [41]. This approach enables real-time adaptability and eliminates the need for detailed physical system modeling.

In this chapter, we describe the modeling and simulation setup for lateral vehicle dynamics. Section 3.1 provides an overview of the lateral dynamics model used in simulations. Section 3.2 details the model identification process using the Online DMD algorithm. Section 3.3 describes the simulation setup and the evaluation of Online DMD in comparison with a baseline linear model.

3.1.1 Model Description

The vehicle system considered in this study has a state-space representation consisting of inputs, states, and outputs, which are defined as follows:

The system states represent the lateral and yaw dynamics of the vehicle:

$$\chi = \begin{bmatrix} v_y \\ r \end{bmatrix},$$

where v_y is the lateral velocity of the vehicle, and r is the yaw rate.

The system input is defined as:

$$u = \begin{bmatrix} \delta \end{bmatrix},$$

where δ represents the steering angle of the vehicle.

The system outputs correspond to the observed states, making the system output matrix:

$$y = \begin{bmatrix} v_y \\ r \end{bmatrix}.$$

This system is modeled as a Single Input, Multiple Output (SIMO) system, where the lateral velocity and yaw rate are outputs, and the steering angle serves as the control input.

The vehicle parameters used in the simulations are:

- $m = 1575$ kg: Mass of the car
- $I_z = 2875$ kg $\cdot$ m^2: Moment of inertia about the Z-axis
- $l_f = 1.2$ m: Distance between the center of gravity and the front axle
- $l_r = 1.6$ m: Distance between the center of gravity and the rear axle
- $C_f = 19000$ N/rad: Cornering stiffness of the front tires
- $C_r = 33000$ N/rad: Cornering stiffness of the rear tires
- $C_d = 0.30$: Drag resistance coefficient (dimensionless)
- $A_f = 2.2$ m^2: Frontal area of the vehicle
- $C_{rr} = 0.01$: Rolling resistance coefficient (dimensionless)
- $h = 0.55$ m: Height of the center of gravity

The simulations were conducted using the *Vehicle Dynamics Blockset* [22], which provides a high-fidelity modeling environment for vehicle dynamics. The system was tested across varying speeds and road coefficients to evaluate its performance under diverse conditions. This setup ensures a realistic simulation environment for analyzing and validating the Online Dynamic Mode Decomposition algorithm.

3.1.2 Dynamics for the Lateral Vehicle Model

The lateral dynamics of the vehicle were modeled using two approaches: a Linear Dynamics Model and a data-driven model derived using the Online Dynamic Mode Decomposition (Online DMD) algorithm. The Linear Dynamics Model, based on the

bicycle model, assumes constant velocity and small-angle approximations, providing a computationally efficient baseline.

Online DMD, on the other hand, identifies the system dynamics directly from observed data and adapts to time-varying conditions in real time. This eliminates the need for prior assumptions about the system, making it suitable for scenarios with changing dynamics. Both methods are compared in simulations to evaluate their performance in capturing lateral vehicle dynamics.

3.2 Simulation Setup and Implementation

The simulations were conducted using MATLAB and Simulink, with the *Vehicle Dynamics Blockset* [22] to model and analyze the lateral dynamics of a passenger car. This environment allows for realistic testing under varied conditions.

The following variations were applied to evaluate the performance and adaptability of the system models:

- **Steering Inputs:** Two types of steering inputs were used to ensure sufficient excitation of the system for accurate parameter identification, satisfying the persistence of excitation condition:
  - **Sinusoidal Input:** Applied based on steering angle limits the vehicle can provide with a low-to-medium frequency, providing excitation within the steering limits at a constant vehicle velocity. The profile of the sinusoidal input is shown in Figure 3.1.

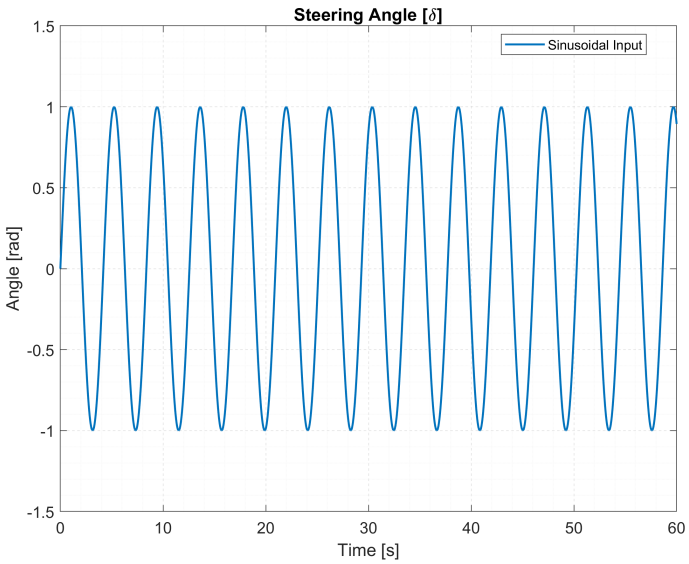


Figure 3.1: Sinusoidal steering wheel input

– **Chirp Signal (Sine Sweep):** Provides excitation across a wide range of frequencies. This input design is cited from [30], and its profile is shown in Figure 3.2.

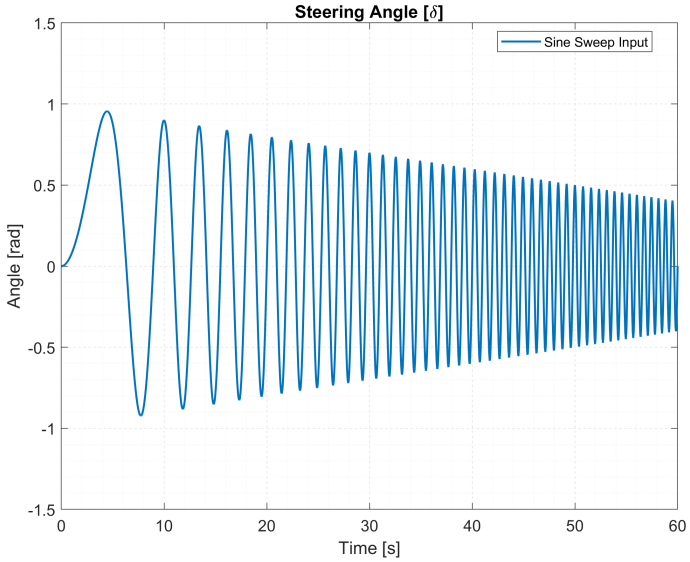


Figure 3.2: Chirp signal (sine sweep) input for broad frequency excitation.

- **Vehicle Speeds:** Simulations were conducted at constant speeds such as 5 m/s and 15 m/s to evaluate model performance across different dynamic regimes.
- **Road Surface Conditions:** The coefficient of friction (μ) was varied to simulate different road conditions:
  - $\mu = 0.9$: Dry asphalt, representing high-traction scenarios.
  - $\mu = 0.3$: Icy roads, representing low-traction scenarios.

Measurement noise was added to simulate sensor imperfections and environmental disturbances, creating a more realistic testing environment for assessing model robustness.

3.3 Simulation Results

The simulation results include the lateral velocity and yaw rate predictions obtained from the Online DMD and Linear Models under various testing conditions. These conditions include variations in steering inputs, vehicle speeds, and road surface coefficients, as described in Section 3.1. The performance is analyzed using three key metrics, as described in the following subsection.

3.3.1 Evaluation Metrics

Three metrics are used to evaluate the accuracy of Online DMD and Linear Models in predicting lateral velocity and yaw rate:

- **Root Mean Square Error (RMSE):** Captures overall accuracy by emphasizing larger errors, making it ideal for assessing general model performance.
- **Mean Absolute Error (MAE):** Measures average error magnitude and provides insights into the model's consistency, with less sensitivity to outliers.
- **Maximum Deviation:** Highlights the worst-case error, ensuring reliability in extreme scenarios, which is critical for safety.

These metrics provide complementary insights into accuracy, consistency, and robustness, forming a comprehensive evaluation framework.

3.3.2 Sinusoidal Steering Input at Low Speed

5 m/s, Dry Asphalt ($\mu = 0.9$)

The simulation results for a sinusoidal steering input at a low speed of 5 m/s on a high-traction surface ($\mu = 0.9$) are shown in Figures 3.3 and 3.4. These figures compare the lateral velocity and yaw rate outputs of the Vehicle Model (actual simulation output), the Linear Model, and the Online Dynamic Mode Decomposition (ODMD) Model.

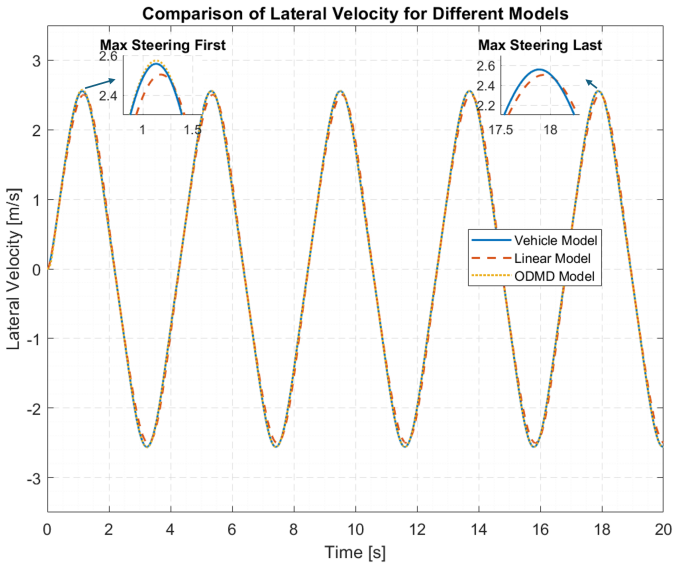


Figure 3.3: Comparison of lateral velocity for sinusoidal input at 5 m/s ($\mu = 0.9$).

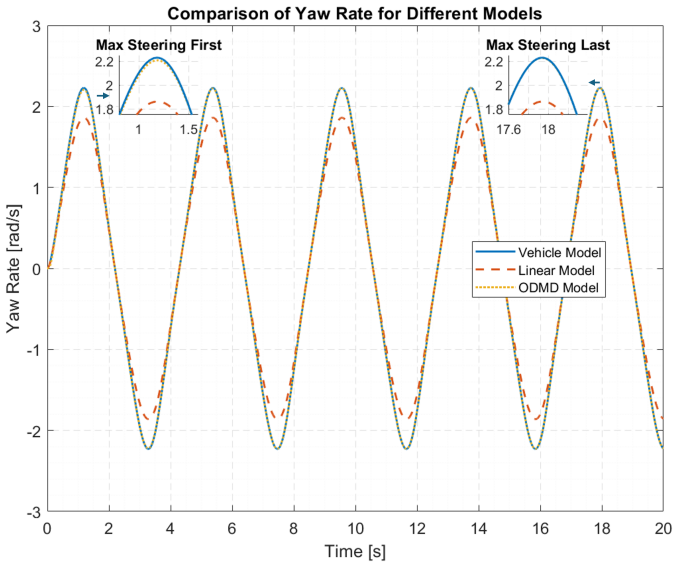


Figure 3.4: Comparison of yaw rate for sinusoidal input at 5 m/s ($\mu = 0.9$).

The zoomed-in areas in the plots focus on the first and last maximum steering inputs to evaluate how the models' performance evolves.

- **Lateral Velocity:** The ODMD model displayed a minor offset during the first maximum steering input, which is not visible by the last input, aligning closely with the actual vehicle output. The Linear Model retained a visible offset throughout, as can be seen in the zoomed regions.
- **Yaw Rate:** Yaw rate errors were more pronounced. The ODMD model started with a small offset that significantly reduced by the last steering input, achieving well model alignment. On the other hand, the Linear Model continued to exhibit noticeable deviations.

For a constant frequency and constant magnitude sinusoidal input, ODMD can update the model and fit the model successfully. Linear model has constant parameters, so, the behaviour of the model remains the same. Also, it shall be noted that in low vehicle speeds and low steering angles, dynamics of the system is more linear. Thus, linear model also performs well at lower steering angles.

3.3.3 Chirp Signal Steering Input

5 m/s, Dry Asphalt ($\mu = 0.9$)

The simulation results for a chirp signal steering input at a low speed of 5 m/s on a high-traction surface ($\mu = 0.9$) are shown in Figures 3.5 and 3.6. The chirp signal's varying frequency and magnitude influence the models' outputs. The ODMD model

maintains high accuracy with diminishing offsets at lower steering magnitudes, while the Linear Model consistently shows visible offsets, particularly at higher steering magnitudes.

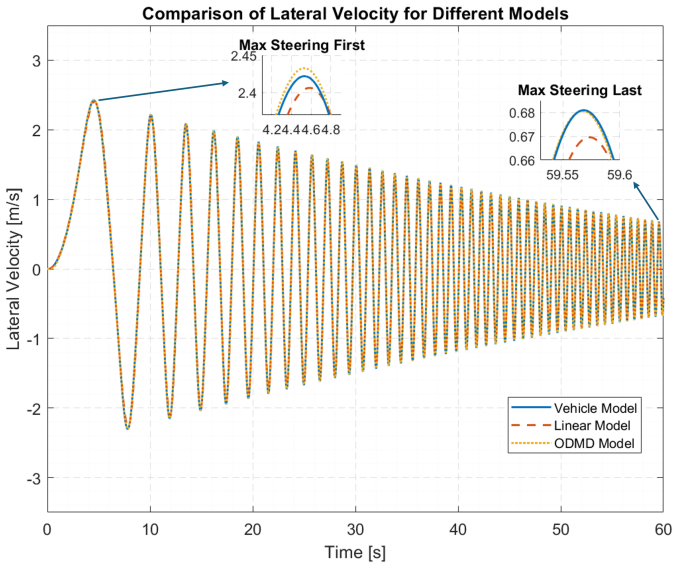


Figure 3.5: Comparison of lateral velocity for chirp signal input at 5 m/s ($\mu = 0.9$).

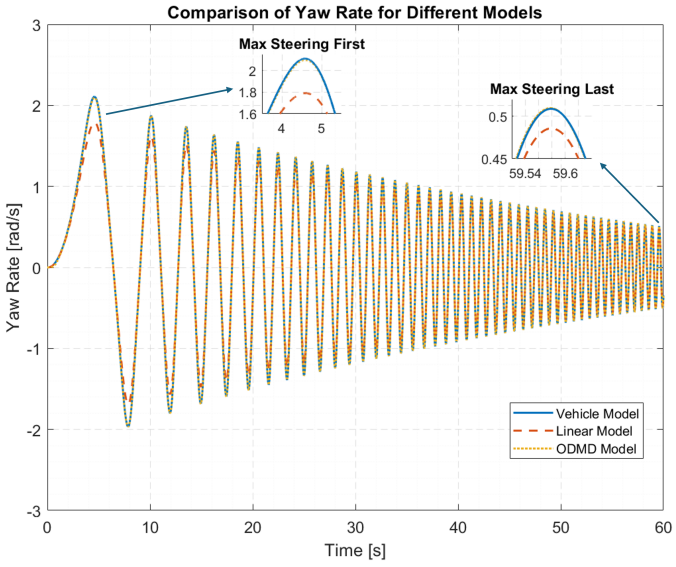


Figure 3.6: Comparison of yaw rate for chirp signal input at 5 m/s ($\mu = 0.9$).

5 m/s, Icy Road ($\mu = 0.3$)

Figures 3.7 and 3.8 show the results for a low speed of 5 m/s on a low-traction surface ($\mu = 0.3$). The Linear Model struggles with significant offsets due to its inability to account for road surface coefficients, though errors decrease at lower steering

magnitudes. The ODMD model remains robust, adapting well to the low-traction condition.

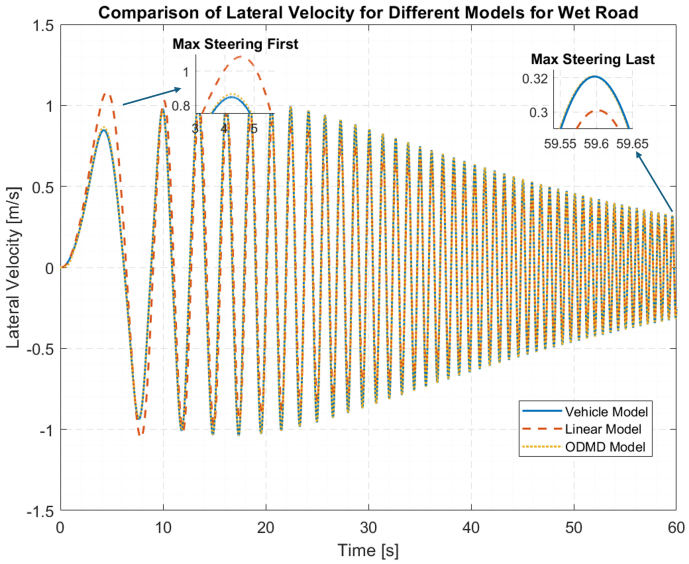


Figure 3.7: Comparison of lateral velocity for chirp signal input at 5 m/s ($\mu = 0.3$).

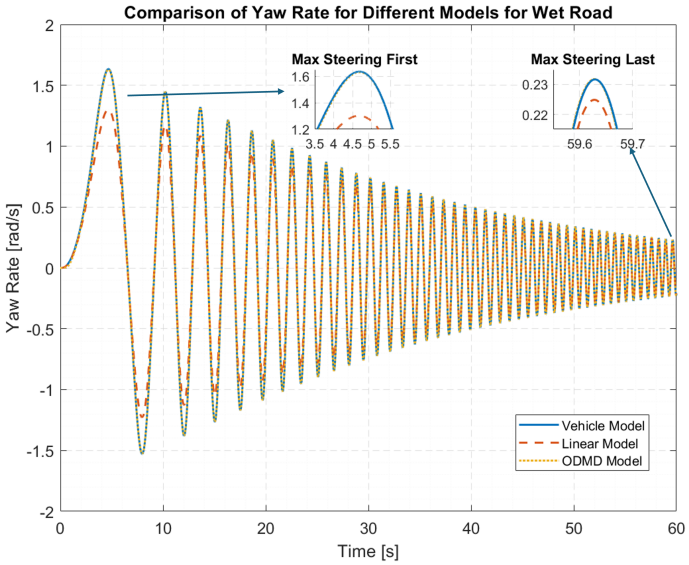


Figure 3.8: Comparison of yaw rate for chirp signal input at 5 m/s ($\mu = 0.3$).

15 m/s, Dry Asphalt ($\mu = 0.9$)

The results for a chirp signal steering input at a medium speed of 15 m/s on a high-traction surface ($\mu = 0.9$) are illustrated in Figures 3.9 and 3.10. The Linear Model displays visible offsets throughout, which are more pronounced at higher magnitudes. The ODMD model continues to perform well, demonstrating consistent accuracy under medium-speed conditions.

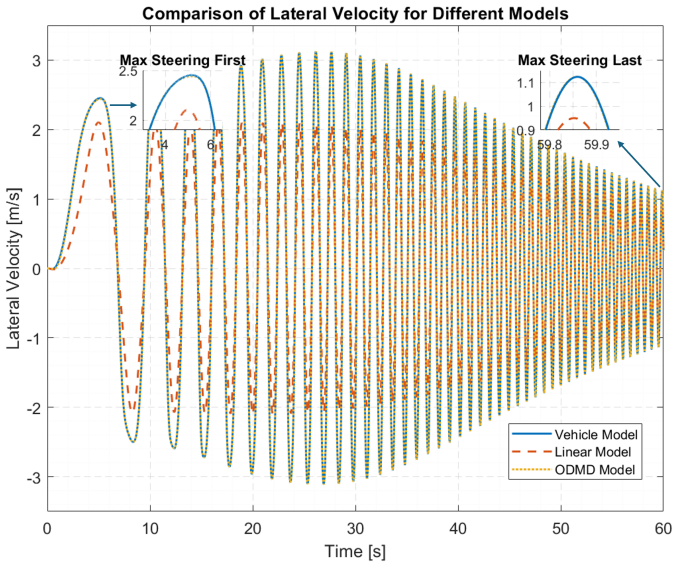


Figure 3.9: Comparison of lateral velocity for chirp signal input at 15 m/s ($\mu = 0.9$).

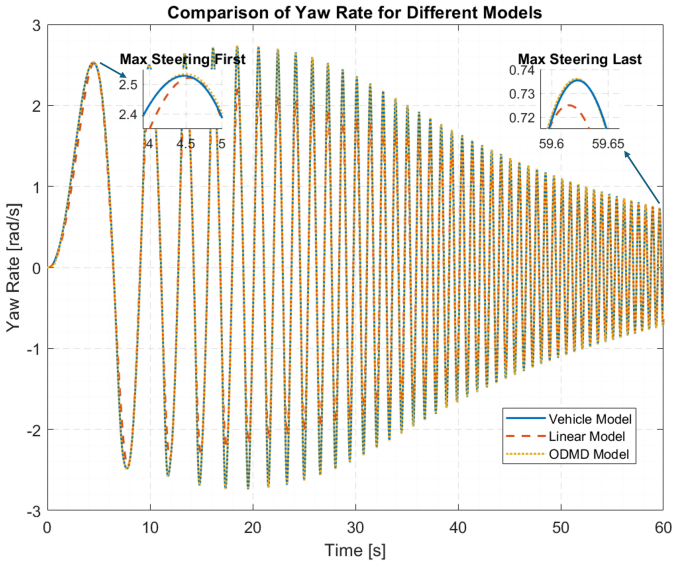


Figure 3.10: Comparison of yaw rate for chirp signal input at 15 m/s ($\mu = 0.9$).

15 m/s, Icy Road ($\mu = 0.3$)

Figures 3.11 and 3.12 display the results for a medium speed of 15 m/s on a low-traction surface ($\mu = 0.3$). The Linear Model shows overshoot initially and undershoot at the end, impacted significantly by the road condition. The ODMD model maintains its robustness and aligns closely with the actual vehicle behavior, unaffected by the reduced traction.

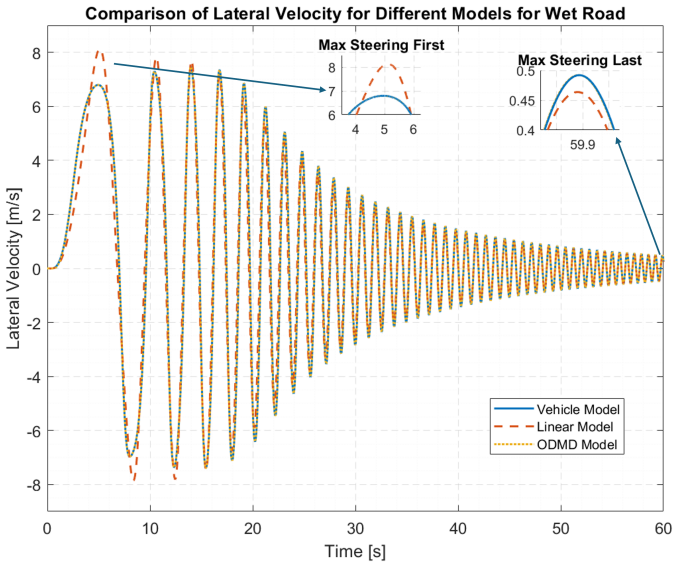


Figure 3.11: Comparison of lateral velocity for chirp signal input at 15 m/s ($\mu = 0.3$).

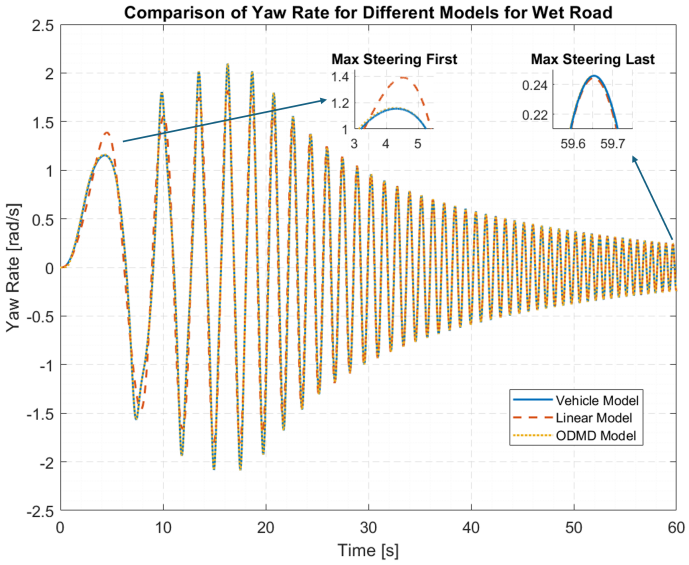


Figure 3.12: Comparison of yaw rate for chirp signal input at 15 m/s ($\mu = 0.3$).

3.4 Discussion

The simulation results presented in the previous sections demonstrate the comparative performance of the Online Dynamic Mode Decomposition (ODMD) and Linear models under various test conditions. Table 3.1 summarizes the evaluation metrics for lateral velocity, while Table 3.2 provides the corresponding results for yaw rate. These results offer valuable insights into the adaptability, accuracy, and robustness of each model.

3.4.1 Performance Across Different Scenarios

The ODMD model consistently outperforms the Linear model across all test cases, as evidenced by significantly lower RMSE, MAE, and MD values in Tables 3.1 and 3.2. This indicates the ability of the ODMD model to adapt to varying dynamics, including changes in vehicle speed, road surface conditions, and input characteristics.

Table 3.1: Evaluation metrics for lateral velocity.

Input	v_x **(m/s)**	μ	**Model**	**RMSE**	**MAE**	**MD**
Sinusoid	5	0.9	Linear	0.9171	0.7978	1.4076
			ODMD	0.0073	0.0045	0.0632
Sweep	5	0.9	Linear	0.5713	0.4779	1.3368
			ODMD	0.0063	0.0053	0.03
Sweep	15	0.9	Linear	2.1243	1.5812	5.678
			ODMD	0.0091	0.0075	0.0285
Sweep	5	0.3	Linear	0.1174	0.0863	0.3394
			ODMD	0.0048	0.0035	0.023
Sweep	15	0.3	Linear	0.6479	0.4203	2.0256
			ODMD	0.0092	0.0071	0.041

Table 3.2: Evaluation metrics for yaw rate.

Input	v_x **(m/s)**	μ	**Model**	**RMSE**	**MAE**	**MD**
Sinusoid	5	0.9	Linear	0.5803	0.5006	0.9311
			ODMD	0.007	0.0044	0.0458
Sweep	5	0.9	Linear	0.4	0.345	0.8112
			ODMD	0.0049	0.0038	0.0232
Sweep	15	0.9	Linear	0.77	0.6549	1.6211
			ODMD	0.0069	0.0055	0.0393
Sweep	5	0.3	Linear	0.1017	0.0682	0.335
			ODMD	0.0046	0.003	0.023
Sweep	15	0.3	Linear	0.1302	0.0871	0.3315
			ODMD	0.0048	0.0032	0.0254

Impact of Steering Inputs

For both sinusoidal and chirp signal steering inputs, the ODMD model demonstrates superior accuracy, particularly in capturing dynamic behaviors at higher frequencies and varying magnitudes. The Linear model exhibits noticeable offsets, especially at higher magnitudes of steering input, as it fails to account for nonlinearity and time-varying dynamics.

Impact of Vehicle Speed

The performance of the Linear model degrades significantly as vehicle speed increases. For instance, at 15 m/s (medium speed), the RMSE for lateral velocity increases more than three times compared to low-speed scenarios, as shown in Table 3.1. This limitation arises because the Linear model cannot adequately capture the increased dynamics and nonlinear effects at higher speeds. In contrast, the ODMD model maintains its accuracy, demonstrating its robustness to speed variations.

Impact of Road Surface Coefficients

The road surface coefficient (μ) has a considerable effect on the Linear model's performance. On icy roads ($\mu = 0.3$), the Linear model shows larger errors due to its inability to account for the reduced traction. For example, in the chirp signal test at 15 m/s and $\mu = 0.3$, the Linear model's MD for lateral velocity is 2.0256, compared to only 0.041 for the ODMD model. The ODMD model, relying solely on data, remains unaffected by road surface variations, consistently aligning with the actual vehicle's output. Also, ODMD enable the algorithm to adapt the model without actually knowing road surface coefficient.

In summary, the ODMD model demonstrated superior adaptability and accuracy in capturing vehicle lateral dynamics under varying conditions, outperforming the Linear Model in dynamic scenarios. Its robustness across different inputs and road conditions establishes a strong foundation for model-based controller designs. The next chapter will build on these results, employing ODMD with Linear Quadratic Regulator (LQR) and Adaptive Model Predictive Control (MPC) strategies to further explore its potential in enhancing autonomous vehicle control systems.

4

LQR and Adaptive MPC Design using Online DMD

In this work, a hierarchical control architecture is proposed for the lane-keeping system of an autonomous vehicle. The system integrates Online Dynamic Mode Decomposition (Online DMD) for real-time system identification and model updates, enabling adaptive control in dynamic environments.

The hierarchical structure consists of two levels of control:

- **Upper-Level Controller:** The upper-level controller is responsible for generating the reference signals for lateral velocity ($\dot{y}_{\text{ref}}$) and yaw rate ($\dot{\psi}_{\text{ref}}$) based on the tracking errors in lateral position (e_y) and yaw angle (e_ψ). These reference signals define the desired motion of the vehicle along the lane centerline.
- **Lower-Level Controller:** The lower-level controller uses the reference signals provided by the upper level to compute the required steering input (δ_f) for path tracking. This controller is designed based on the real-time updated system matrices ($\hat{A}$ and $\hat{B}$) provided by the Online DMD module.

The Online DMD module plays a critical role by continuously updating the linear model of the vehicle system based on real-time measurements of lateral velocity ($\dot{y}$), yaw rate ($\dot{\psi}$), and longitudinal velocity (v_x). This ensures that the controllers adapt to variations in road conditions, vehicle dynamics, and environmental uncertainties.

The overall control architecture, depicted in Figure 4.1, illustrates the interaction between the vehicle system, the hierarchical control layers, and the Online DMD module.

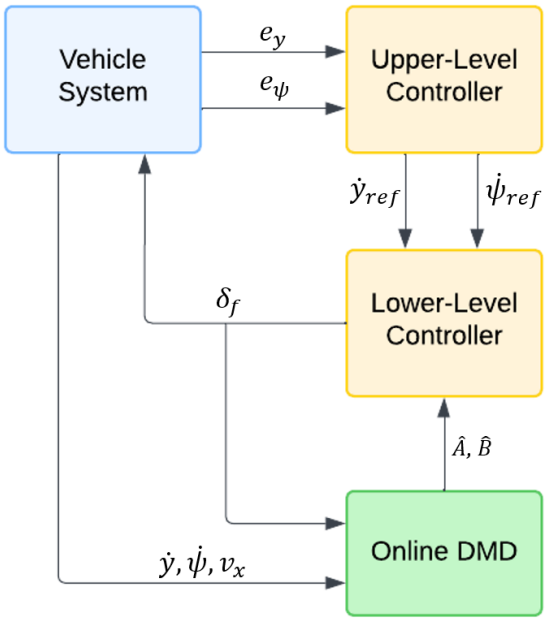


Figure 4.1: Hierarchical control system architecture for the lane-keeping system.

4.1 Upper-Level Controller

The upper-level controller is responsible for generating the desired reference signals for lateral velocity ($\dot{y}_{\text{ref}}$) and yaw rate ($\dot{\psi}_{\text{ref}}$) based on the vehicle's position and orientation relative to the lane center. This controller processes real-time data obtained from a camera sensor, which detects the vehicle's lateral distance and relative yaw angles to both the left and right lane markers, as illustrated in Figure 4.2.

To compute the error values for lane centering, the lateral distances ($e_{y,\text{left}}$ and $e_{y,\text{right}}$) and yaw angles ($e_{\psi,\text{left}}$ and $e_{\psi,\text{right}}$) relative to the lane markers are obtained from both lanes and averaged as follows:

$$e_y = \frac{e_{y,\text{left}} + e_{y,\text{right}}}{2}, \tag{4.1}$$

$$e_\psi = \frac{e_{\psi,\text{left}} + e_{\psi,\text{right}}}{2}. \tag{4.2}$$

These errors represent the vehicle's lateral deviation and relative orientation with respect to the lane center and serve as inputs to the upper-level controllers.

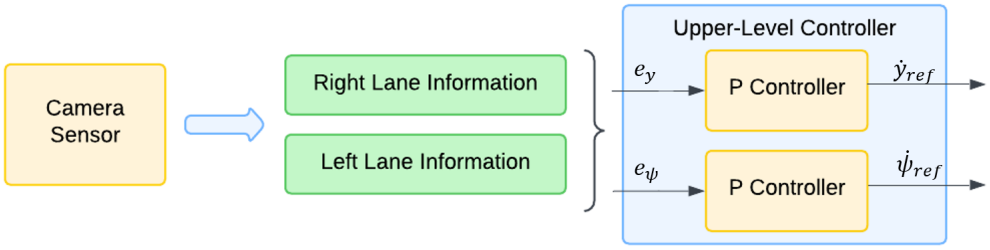


Figure 4.2: Upper-level controller design: lane information from both the left and right lane markers is used to compute the lane center.

Proportional (P) controllers are employed to regulate the lateral velocity and yaw rate references based on these error values:

$$\dot{y}_{\text{ref}} = K_y e_y, \tag{4.3}$$

$$\dot{\psi}_{\text{ref}} = K_\psi e_\psi, \tag{4.4}$$

where K_y and K_ψ are the proportional gains for the lateral and yaw control loops, respectively. These gains are tuned to ensure stable and responsive tracking of the lane center under varying conditions.

The upper-level controller uses a proportional gain to adjust the vehicle's lateral response by utilizing system error data. This approach is similar to methods from earlier studies [30, 32, 33], which use lane error to produce the target state for lower-level control.

4.2 Lower-Level Controller

This section outlines the design and implementation of two lower-level control strategies: Linear Quadratic Regulator (LQR) and Model Predictive Control (MPC). These controllers process the reference signals generated by the upper-level controller to produce steering inputs for accurate lane-keeping. The specific controller parameters will be detailed in the Results section.

4.2.1 Overview of Lower-Level Control Strategies

The lower-level controllers are tasked with minimizing lateral and yaw errors by considering vehicle dynamics under varying speeds and road conditions. The LQR was selected for its optimal trade-off between tracking performance and control effort, while MPC was chosen for its ability to handle constraints and to test the prediction capability of the system updates.

4.2.2 Linear Quadratic Regulator (LQR) Design

The Linear Quadratic Regulator (LQR) is implemented as the lower-level control strategy to minimize tracking errors in lateral position and yaw rate. This control strategy uses the linearized state-space representation of the vehicle dynamics, updated dynamically using the Online Dynamic Mode Decomposition (DMD) algorithm.

Control Law Formulation

The LQR control law is designed as:

$$u = -K_f x_k + K_r u_c$$

where:

- u: Control input (steering angle).
- K_f: Feedback gain matrix.
- K_r: Reference gain matrix.
- x_k: Current state vector.
- u_c: Reference signal for the desired state.

Gain Calculation

The feedback gain K_f is computed by solving the discrete-time Algebraic Riccati Equation (ARE) using MATLAB's `dlqr` function. The weighting matrices for the cost function are chosen as:

$$Q = \begin{bmatrix} Q_{11} & 0 \\ 0 & Q_{22} \end{bmatrix}, \quad R = R_1$$

where:

- Q: State weighting matrix, penalizing deviations from the desired state.
- R: Control effort weighting, penalizing excessive control input.

Reference Gain Calculation

The reference gain K_r is computed to ensure the control law appropriately incorporates the reference signal. This is achieved using:

$$M = B^T S B + R, \quad N = I - (A - BK_f)^T, \quad O = C^T Q$$

where S is the solution to the Riccati equation. The reference gain is determined as:

$$K_r = M^{-1}(B^T N^{-1} O)$$

This ensures the control signal drives the system toward the reference trajectory effectively.

Implementation Details

The control input is calculated in real-time using the updated A and B matrices provided by the Online DMD algorithm. The process involves:

- Dynamically updating the system matrices (A, B).
- Solving for the feedback and reference gains (K_f, K_r).
- Computing the control input (u) using the current state (x_k) and reference signal (u_c).

This approach ensures adaptability to changing vehicle dynamics, improving the controller's performance in dynamic and uncertain environments.

4.2.3 Adaptive Model Predictive Control Design

Model Predictive Control is implemented as the lower-level control strategy to optimize tracking performance while respecting system constraints. An adaptive version of MPC is employed, where the system model is dynamically updated using the Online DMD algorithm. The implementation is carried out using MATLAB's *Model Predictive Control Toolbox*, which provides built-in functionalities for real-time optimization and adaptive system modeling [21, 1].

Control Strategy Overview

MPC uses a predictive model to optimize the control inputs over a defined prediction horizon. At each control step, the control signal is obtained by solving an optimization problem that minimizes a predefined cost function, subject to system constraints. The control law can be expressed as:

$$\min_{u_{0:N-1}} \sum_{k=0}^{N-1} \left(x_k^T Q x_k + u_k^T R u_k \right)$$

subject to:

$$x_{k+1} = Ax_k + Bu_k, \quad u_{\min} \leq u_k \leq u_{\max}$$

where:

- N: Prediction horizon, the total number of future steps considered for optimizing the control strategy.
- C: Control horizon, the number of future control moves optimized at each step.
- x_k: State vector at step k.
- u_k: Control input at step k.
- Q, R: Weighting matrices for states and control effort, respectively.
- $u_{\min}, u_{\max}$: Bounds on the control input (steering angle).

Adaptive Model Integration

To ensure the MPC controller adapts to changing vehicle dynamics, the system matrices (A, B) are updated in real-time using the Online DMD algorithm. This adaptive approach allows the controller to maintain accuracy under varying operating conditions, such as changes in speed or road surface properties. The adaptive model updates enable the controller to incorporate the latest vehicle dynamics, ensuring precise predictions and control actions.

Optimization and Constraints

The optimization problem is solved at each control step. The constraints ensure the control input remains within physically allowable limits and the predicted states respect system dynamics. For instance:

$$u_k \in [u_{\min}, u_{\max}], \quad x_k \in \mathcal{X}$$

where $\mathcal{X}$ represents the allowable state space. The use of a control horizon (C) ensures that only the first M control moves are optimized within the prediction horizon, reducing computational complexity while maintaining performance.

Implementation Details

The implementation process involves:

- Model Updates: Dynamically updating the A and B matrices using Online DMD.
- Prediction Horizon (N): Forecasting system behavior over a fixed horizon for optimized control actions.

- Control Horizon (M): Limiting the number of control moves optimized to balance computational load and performance.
- Cost Function Weights: Employing Q and R matrices to balance tracking accuracy and control effort.
- Real-Time Optimization: Solving the optimization problem iteratively at each control step to compute the control input.

This adaptive MPC framework combines the MPC with the Online DMD module, ensuring robust path tracking performance under dynamic and uncertain conditions.

4.3 Integration of Online DMD in Control Design

The Online DMD methodology, detailed in Chapter 3, provides a framework for modeling vehicle dynamics. By extracting state-space matrices (A, B) from real-time data, Online DMD captures the system behavior under varying operating conditions, such as changes in vehicle speed, road surface properties, and steering inputs.

In this chapter, Online DMD plays a pivotal role in enabling the real-time adaptability of both the LQR and MPC strategies. The updated state-space matrices are used to:

- Recalculate feedback gains for the LQR, ensuring optimal control performance under dynamic conditions.
- Update the predictive model for the Adaptive MPC, enhancing its ability to predict system behavior and optimize control inputs.

This integration of Online DMD with control design will allow that the controllers remain accurate, even in the presence of significant system variations. For further details on the modeling process using Online DMD, refer to Chapter 3.

5

Simulation Results and Discussion

In this chapter, the performance of the proposed control strategies is evaluated under different simulation scenarios using a high-fidelity simulator in Matlab. These scenarios aim to test the robustness and adaptability of the control system under varying conditions. The evaluation includes:

- **Constant Curvature:** The vehicle tracks the lane center on a circular road with a constant curvature to evaluate steady-state performance.
- **Sharp Turn:** Lane tracking performance is tested under a sharp turning scenario, representing a road with abrupt changes in curvature.
- **City-Like Road:** The vehicle follows a road with varying curvature profiles, including smooth transitions, constant curvatures, and sharp turns, resembling realistic urban driving conditions.

5.1 Simulation Setup

This section provides an overview of the simulation environment, including the vehicle parameters, sensor configurations, and controller tuning strategies used to evaluate the performance of the proposed control methods.

5.1.1 Vehicle Parameters

The vehicle dynamics model is implemented using MATLAB's Vehicle Dynamics Blockset, which provides a high-fidelity simulation environment for vehicle dynamics. The control strategies, including the LQR and Adaptive MPC controllers, are designed and tested within the Simulink framework. Disturbances and measurement noise are incorporated into the simulation to evaluate the robustness and adaptability of the proposed controllers under realistic conditions.

Simulation parameters for the vehicle are summarized in Table 5.1. These parameters reflect a typical passenger car and are used consistently across all simulation scenarios.

Table 5.1: Simulation Parameters for Vehicle Dynamics

Parameter	Value	Description
m	1575 kg	Mass of the vehicle
I_z	2875 kg $\cdot$ m^2	Moment of inertia about Z-axis
l_f	1.2 m	Distance from CoG to front axle
l_r	1.6 m	Distance from CoG to rear axle
C_f	19000 N/rad	Cornering stiffness of front tires
C_r	33000 N/rad	Cornering stiffness of rear tires
C_d	0.30	Drag resistance coefficient
A_f	2.2 m^2	Frontal area of the vehicle
C_{rr}	0.01	Rolling resistance coefficient
h	0.55 m	Height of the CoG
g	9.81 m/s^2	Gravitational acceleration
μ	0.9	Road friction coefficient (Asphalt Road)
v_x	10 m/s	Longitudinal velocity

These parameters ensure consistency and provide a foundation for comparing controller performance across different scenarios. The inclusion of measurement noise and road disturbances adds complexity, allowing for a thorough assessment of the controllers' robustness and adaptability under realistic operating conditions.

A constant longitudinal speed of $v_x = 10\,\mathrm{m/s}$ is chosen to balance between system linearity and physical feasibility. Since a longitudinal controller is not implemented in this study, maintaining a higher constant speed may not be physically possible due to the limitations imposed by sharp curvatures and road friction. A medium-speed value is selected to avoid very low speeds, where the vehicle dynamics exhibit near-linear behavior, while also ensuring that the vehicle does not operate near its physical limits at high speeds. This choice simplifies the study's scope while maintaining the focus on lateral control performance.

5.1.2 Sensor Setup

The simulation employs a camera sensor to detect and provide real-time information about the vehicle's position relative to the lane markers. The sensor is configured and placed using MATLAB's *Automated Driving Toolbox*, which allows to simulate camera behavior in predefined dynamic driving environments. The camera is positioned at $(X, Y, Z) = (1.2, 0, 1)$ in the vehicle's coordinate frame, providing sufficient field of view for lane detection.

The camera detects the lateral distances to both the left and right lane markers, rather than directly providing the distance to the lane center. The lane center error is subsequently calculated based on the detected distances, as described in Section 4.1. This process ensures precise alignment of the vehicle within the lane boundaries.

The sensor's configuration is summarized below:

- **Lane Update Interval:** 100 ms
- **Detection Probability:** 0.9
- **Maximum Range:** 100 m
- **Bounding Box Accuracy:** 5 pixels
- **Process Noise Intensity:** 5 m/s^2
- **Camera Noise:** Enabled

These settings are adjust to simulate the camera behavior that is accounting for noise and measurement uncertainties.

5.1.3 Controller Parameters

The controller parameters are calibrated for the given vehicle dynamics and simulation scenarios through an iterative process to ensure stable and optimal performance. These parameters are specifically tuned to achieve accurate tracking and robust control under varying conditions, including changes in road friction and vehicle speed. The following sections detail the parameters used for both the upper-level and lower-level controllers.

Upper-Level Controller Parameters

The upper-level controllers are designed to calculate the desired lateral velocity and yaw rate based on the error values derived from the camera sensor data. Proportional (P) controllers are employed for both lateral deviation and yaw deviation, and their gains are tuned to balance responsiveness and stability.

Table 5.2: Upper-Level Controller Parameters

Controller	**Proportional Gain (K)**
Lateral Deviation (P_{e_y})	2.8
Yaw Deviation (P_{e_ψ})	4.0

These parameters are fine-tuned to ensure that the reference signals for the lower-level controllers are generated with sufficient error and responsiveness, as described in Section 4.1.

Lower-Level Controller Parameters

The lower-level controllers are designed to compute the control inputs (steering angle) required to minimize the tracking errors for lateral velocity and yaw rate. Two control strategies are employed: Adaptive MPC and LQR. The parameters for these controllers are calibrated for the given vehicle dynamics and scenarios, as detailed in the tables below.

Adaptive MPC Parameters Two sets of MPC parameters are tested to evaluate the impact of the prediction horizon on control performance. The parameters are summarized in Table 5.3.

Table 5.3: Adaptive MPC Parameters

Parameter	**MPC 1**	**MPC 2**
Prediction Horizon (N_p)	10	30
Control Horizon (N_c)	4	10
Manipulated Variable Limit ($u_{\min}, u_{\max}$)	$[-1, 1]$ rad	$[-1, 1]$ rad
Output Weights ($Q_{\dot{y}}, Q_{\dot{\psi}}$)	12, 20	12, 20
Manipulated Variable Rate Weight ($w_{\Delta u}$)	0.01	0.01

LQR Parameters The LQR controller is designed with parameters consistent across all scenarios. The weighting matrices for the state variables and control effort are provided in Table 5.4. Additionally, the same steering limit as the MPC controllers ($[-1, 1]$ radians) is applied to the output of the LQR controller as a saturation limit, ensuring the physical constraints of the steering system are respected.

Table 5.4: LQR Parameters

Parameter	**Value**
State Weights ($Q_{\dot{y}}, Q_{\dot{\psi}}$)	12, 20
Control Effort Weight (R_u)	0.01
Manipulated Variable Limit ($u_{\min}, u_{\max}$)	$[-1, 1]$ rad

Both controllers are designed to respect the physical constraints of the steering system while minimizing tracking errors. The MPC parameters are varied to assess the trade-offs between computational complexity and performance, whereas the LQR parameters remain fixed for consistent evaluation across scenarios. The applied

saturation ensures that both controllers operate within the same physical bounds for fair comparison.

5.1.4 Scenario Design

The evaluation scenarios listed at the beginning of this section are designed using the Driving Scenario Designer App from MATLAB's Automated Driving Toolbox [20]. This app enables the creation of realistic driving environments, incorporating road geometries, vehicle maneuvers, and external disturbances to simulate various conditions for testing.

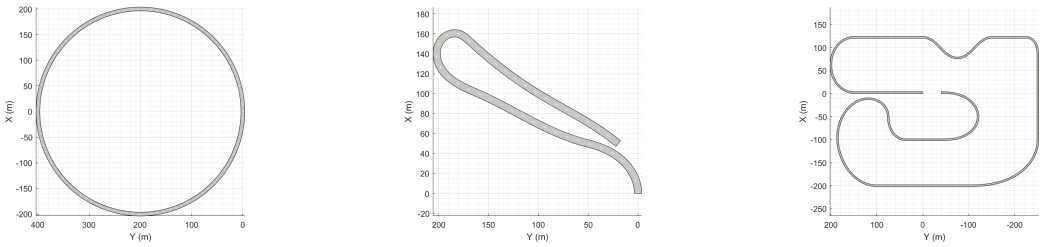
To test the robustness of the proposed controllers and the Online DMD model, wind disturbance is added to the scenarios with the following properties:

- **External Lateral Wind Forces:** White noise with a constant noise power added as a force simulating lateral wind disturbances applied to the vehicle.

Three different road scenarios are designed to evaluate the controllers' performance:

- **Constant Curvature:** A circular road to test lane-keeping performance under steady-state conditions.
- **Sharp Turn:** A road with a sharp U-turn to assess controller adaptability during sudden curvature changes.
- **City-Like Road:** A complex road with varying curvatures to mimic urban driving conditions.

The visualizations of these scenarios are shown in Figure 5.1, highlighting the diversity of test cases used to evaluate the proposed control strategies.



(a) Constant Curvature **(b)** Sharp Turn **(c)** City-Like Road

Figure 5.1: Driving scenarios designed for evaluation: (a) Constant Curvature, (b) Sharp Turn, and (c) City-Like Road.

5.1.5 Initialization

To ensure accurate system identification and stable control performance from the start of each simulation, an initialization procedure is implemented. During the ini-

tial phase, the controllers rely on a predefined constant linear model, as described in Chapter 3, until the system matrices (A, B) generated by the Online Dynamic Mode Decomposition (DMD) algorithm become nonzero. This constant model provides a baseline for the controllers to operate while the Online DMD adapts to the actual system dynamics.

Additionally, an initial yaw rate of 0.01 rad/s is applied to the vehicle. This small excitation ensures measurable lateral dynamics, which are essential for the Online DMD algorithm to compute meaningful initial estimates of the system matrices. Without this initialization, the absence of lateral movement would result in degenerate system matrices, compromising the control and identification processes.

This initialization strategy ensures the robustness and stability of the controllers during the early stages of simulation while allowing the Online DMD model to adapt dynamically as the vehicle begins to interact with the road and environment. The approach minimizes transient effects and provides a smooth transition to fully adaptive control based on the evolving system matrices. In every scenario, it is made sure that the controllers uses the Online DMD model and not the constant model so that the interpreted results are based on controllers with adaptive system model.

5.1.6 Disturbance Force Configuration

To introduce realistic challenges, a fixed-seed Band-Limited White Noise is used to simulate lateral wind forces acting on the vehicle. This noise is applied consistently across all scenarios, ensuring comparability of results.

The noise is configured with the following parameters:

- **Noise Power:** 100, representing the height of the Power Spectral Density (PSD) of the white noise.
- **Fixed Seed:** Ensures reproducibility of the disturbance force across simulations.

Figure 5.2 shows the applied lateral disturbance force over time for the city-like road scenario, which encapsulates the range of disturbances experienced across all scenarios.

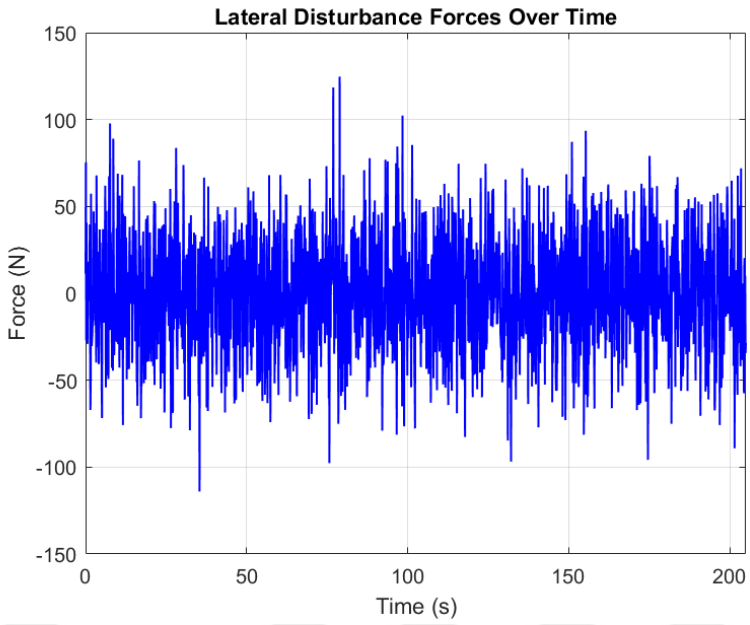


Figure 5.2: Lateral disturbance force applied during the city-like road scenario.

The disturbance force provides an additional layer of complexity, allowing for a comprehensive evaluation of the robustness and adaptability of the proposed controllers.

5.2 Case 1: Constant Curvature Road

This section evaluates the performance of the proposed controllers on a circular road representing a constant curvature scenario. In this scenario, vehicle follow the lane on a circular road and the scenario ends when vehicle reaches the starting point.

5.2.1 Trajectory Visualization

Figure 5.3 shows the vehicle's trajectories compared to the desired path on the constant curvature road. The trajectory visualization highlights the controllers' tracking performance, with deviations indicating potential shortcomings in maintaining the desired path.

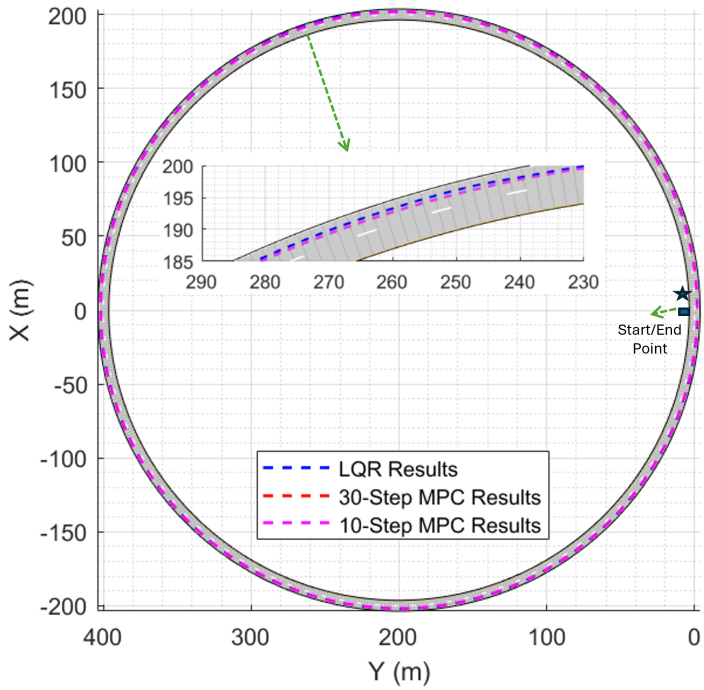


Figure 5.3: Trajectory visualization for the constant curvature road scenario.

The trajectory plot compares the results of the proposed controllers. Both the 10-step MPC and 30-step MPC exhibited very similar and superior performance compared to the LQR controller. While the LQR controller successfully kept the vehicle within the lane, there were noticeable deviations in certain sections, likely due to the influence of disturbance factors. The following sections will go through model evaluation and controller evaluation to better assess the results.

5.2.2 Model Evaluation

The accuracy of the Online DMD model is evaluated in terms of its ability to estimate the lateral velocity and yaw rate of the vehicle. The modeling errors are analyzed using RMSE and maximum deviation values.

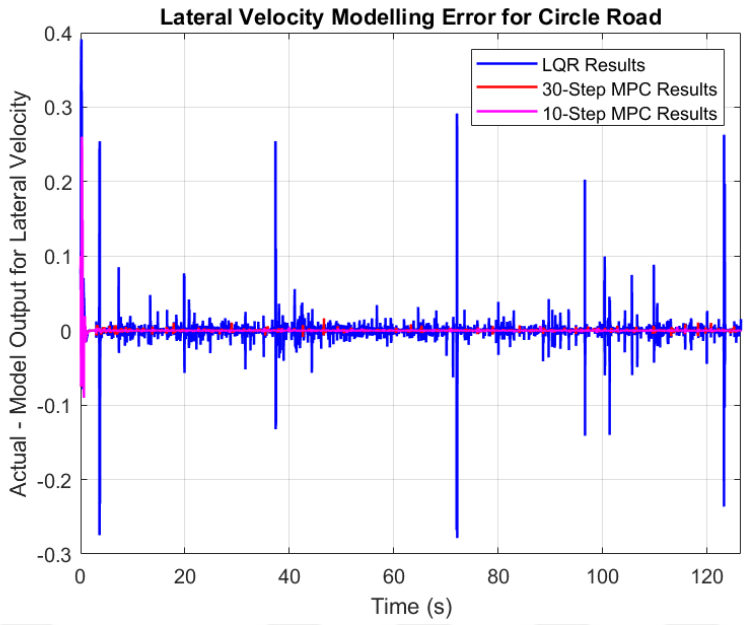


Figure 5.4: Real-time modeling error for lateral velocity on the constant curvature road.

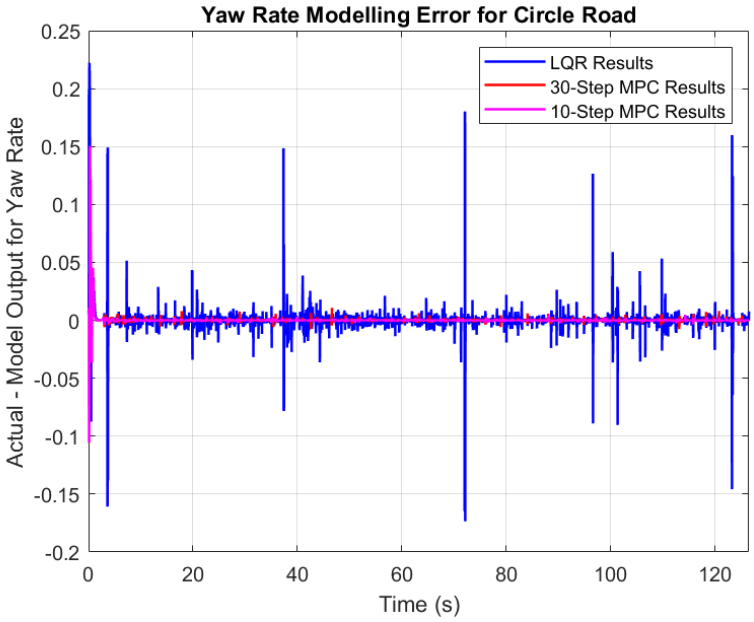


Figure 5.5: Real-time modeling error for yaw rate on the constant curvature road.

Table 5.5 summarizes the RMSE and maximum deviation values for lateral velocity and yaw rate modeling errors.

Table 5.5: Modeling errors for the constant curvature road scenario.

Metric	LQR	30-Step MPC	10-Step MPC
Lateral Velocity RMSE	0.0182	0.0071	0.0071
Lateral Velocity MD	0.3912	0.2599	0.2594
Yaw Rate RMSE	0.0111	0.0053	0.0052
Yaw Rate MD	0.2222	0.1506	0.1506

The results demonstrate the accuracy of the Online DMD model under constant curvature conditions, with variations observed based on the controller used. The modeling results for the MPC controllers remain consistent with the findings presented in Section 3, highlighting their accuracy. Although the LQR results are still commendable, they are notably worse compared to the MPC controllers. Since all inputs to the Online DMD, except the control input, are identical, the observed differences suggest that the control input plays a significant role in the model's performance. Additionally, the minimal difference between the 10-step and 30-step MPC results could be attributed to the disturbance effects, though it remains uncertain if the prediction horizon influenced this outcome.

5.2.3 Control Evaluation

The control performance is assessed based on the **lateral deviation**, **relative yaw angle errors**, and **steering angle input**. Figures 5.6 and 5.7 illustrate the lateral deviation and yaw angle errors for each controller, while Figure 5.8 shows the steering angle input used by the controllers. Table 5.6 summarizes the RMSE and Maximum Deviation (MD) values for lateral deviation and yaw angle, providing a quantitative comparison.

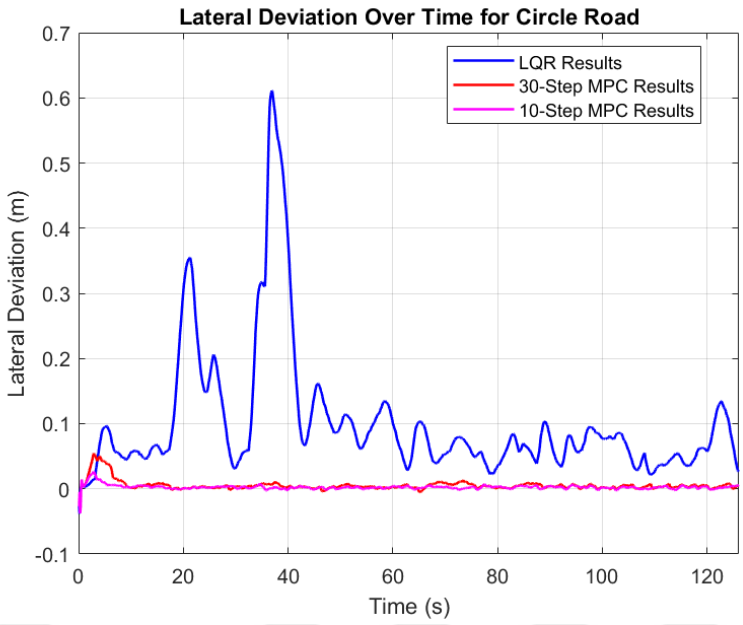


Figure 5.6: Lateral deviation for the constant curvature road scenario.

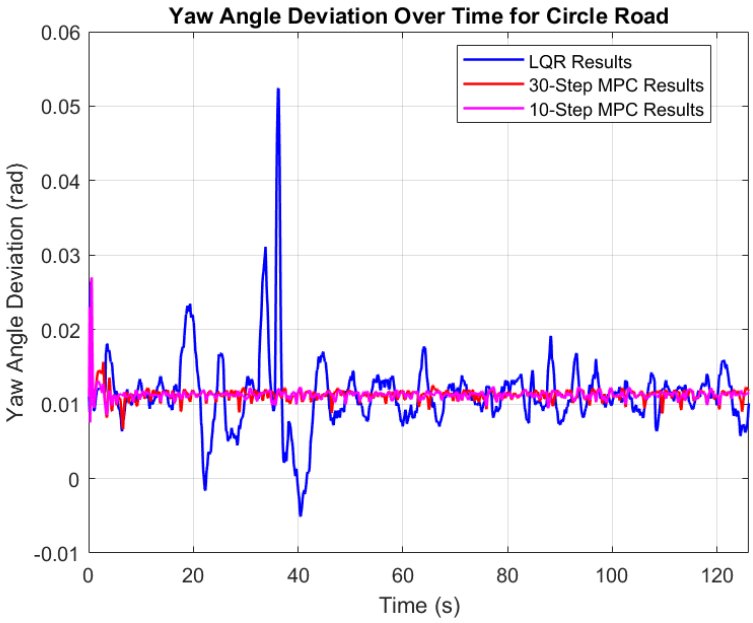


Figure 5.7: Relative yaw angle for the constant curvature road scenario.

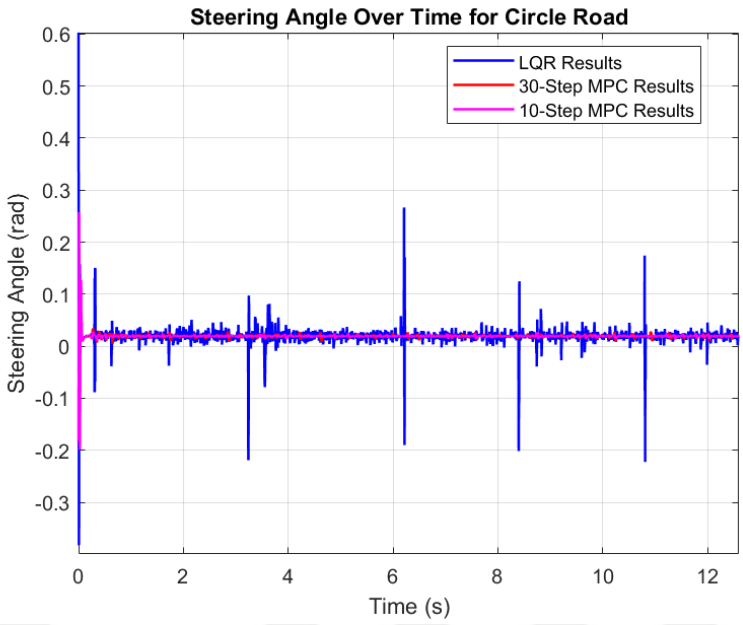


Figure 5.8: Steering angle input over time for the constant curvature road scenario.

Table 5.6: Lateral deviation and yaw angle errors for the constant curvature road.

Metric	LQR	30-Step MPC	10-Step MPC
Lateral Deviation RMSE	0.1428	0.0096	0.0044
Lateral Deviation MD	0.611	0.0543	0.0375
Yaw Deviation RMSE	0.0123	0.0113	0.0113
Yaw Deviation MD	0.0524	0.027	0.027

The **lateral deviation** plot evaluates the controllers' ability to maintain the lane center, while the **yaw angle** plot assesses their ability to align the vehicle's orientation with the road. The **steering angle** plot, on the other hand, provides insights into the control effort required by each controller to achieve the desired trajectory.

The results indicate the following:

- **Lateral deviation** and **relative yaw angle** errors are significantly larger in the **LQR controller** compared to the MPC controllers, as shown in the figures and supported by the Maximum Deviation values in Table 5.6.
- The **modelling performance** for the LQR controller was observed to be worse compared to the MPC controllers, which correlates with its poorer control performance.
- For both **MPC controllers**, the lateral deviation fluctuates closely around zero, with the 10-step MPC performing slightly better in certain sections of the simulation.

- In the **relative yaw angle** plot, all three controllers show fluctuations around 0.01, with the LQR exhibiting more aggressive fluctuations. These fluctuations align with the sections where the modelling errors are highest, confirming the impact of model inaccuracies on control performance.
- The **steering angle plot** (Figure 5.8) reveals significant oscillations in the LQR controller's steering angle, particularly during steady-state conditions. These oscillations are indicative of less efficient control effort, which may result in driver discomfort in real-world applications. In contrast, the MPC controllers exhibit smoother steering behavior with reduced oscillation magnitude, showing better handling of control effort.
- Although the relative yaw angle does not perfectly reach zero, the results remain acceptable. A more careful tuning of the upper-level **P controller** could further improve performance.

These observations highlight the accuracy of the **MPC controllers**, with the 10-step MPC achieving slightly better results while offering reduced computational complexity for a constant curvature road. The analysis also underscores the importance of accurate system modeling in ensuring controller performance, as evident in the performance discrepancies between LQR and MPC. Furthermore, the evaluation of steering angle inputs emphasizes the need for efficient control effort to maintain stability and comfort, where the MPC controllers excelled compared to the LQR controller.

5.3 Case 2: Sharp Turn

This section evaluates the performance of the proposed controllers in a sharp turn scenario, representing roads with abrupt changes in curvature. In this scenario, the vehicle navigates a U-shaped turn, which challenges the controllers to maintain lane tracking accuracy while responding to sudden changes in road geometry. The scenario concludes once the vehicle successfully exits the sharp turn.

5.3.1 Trajectory Visualization

Figure 5.9 shows the vehicle's trajectories compared to the desired path on the sharp turn scenario. The trajectory visualization demonstrates how the controllers handle abrupt changes in road curvature, with deviations highlighting their ability to respond to sudden geometry shifts.

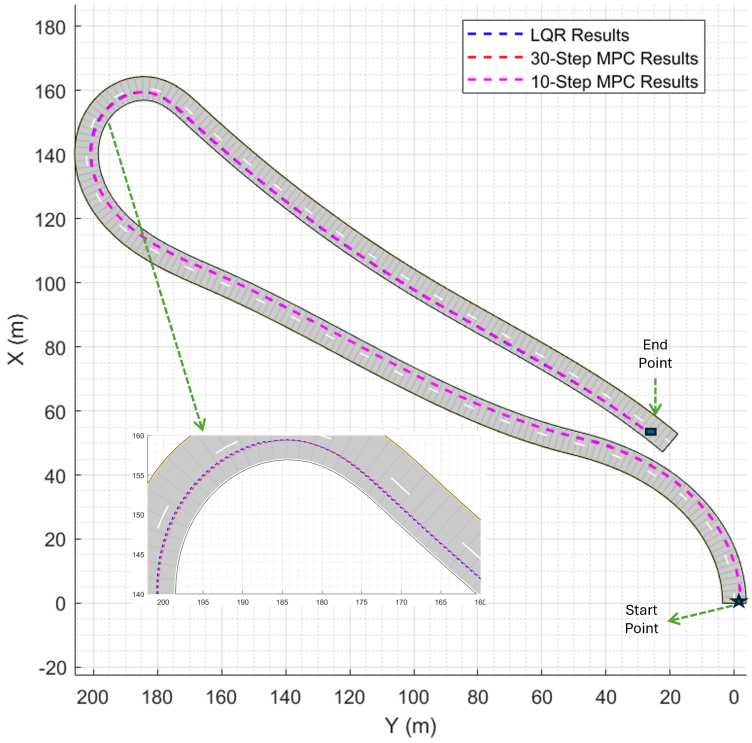


Figure 5.9: Trajectory visualization for the sharp turn scenario.

For lower curvatures and flat road sections, all controllers perform similarly and maintain good tracking accuracy. However, at the sharp turn, as illustrated in the close-up section of the figure, all controllers exhibit some deviation from the lane center. The LQR controller, in particular, has the hardest time recovering back to the centerline and demonstrates the greatest deviation compared to the MPC controllers.

The 10-step MPC and 30-step MPC controllers show superior performance in minimizing deviations, with both controllers quickly stabilizing after the turn. The following sections will provide a detailed analysis of the modeling accuracy and control performance to better assess these results.

5.3.2 Model Evaluation

The accuracy of the Online DMD model is evaluated in terms of its ability to estimate the lateral velocity and yaw rate of the vehicle. The modeling errors are analyzed using RMSE and maximum deviation values.

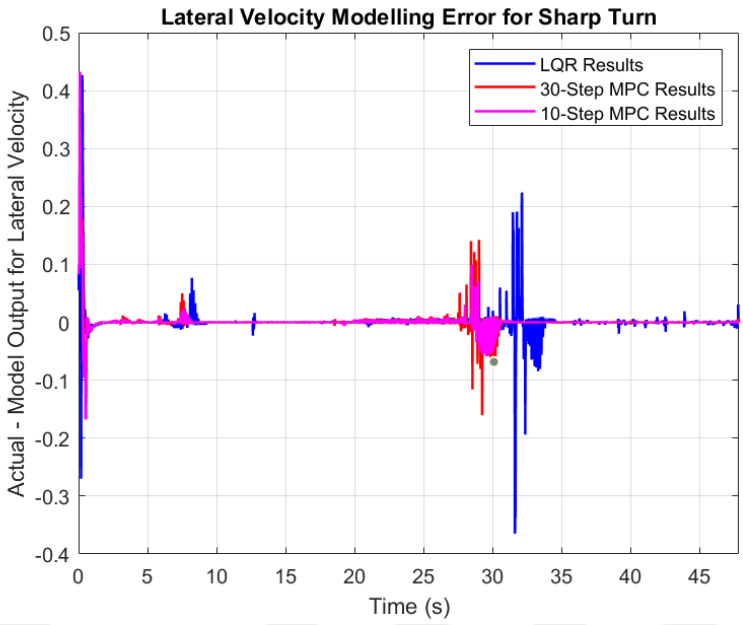


Figure 5.10: Real-time modeling error for lateral velocity on the sharp turn scenario.

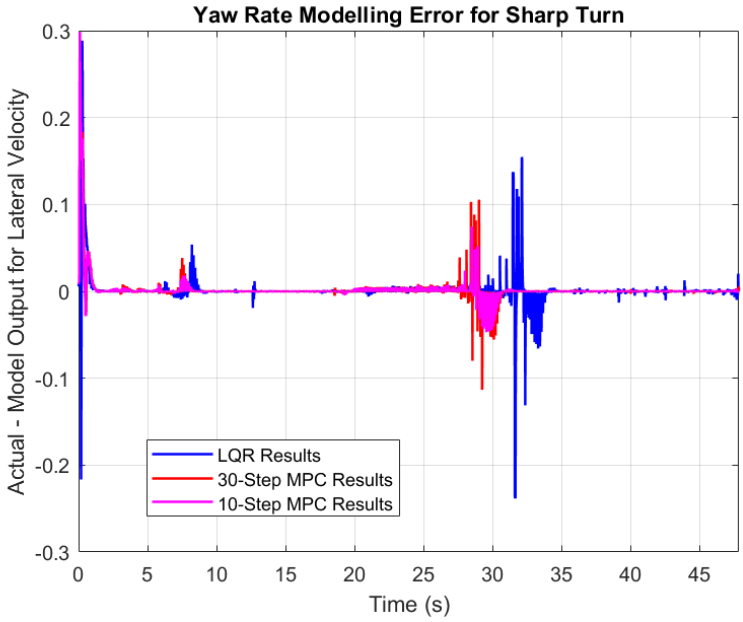


Figure 5.11: Real-time modeling error for yaw rate on the sharp turn scenario.

Table 5.7 summarizes the RMSE and maximum deviation values for lateral velocity and yaw rate modeling errors.

Table 5.7: Modeling errors for the sharp turn scenario.

Metric	LQR	30-Step MPC	10-Step MPC
Lateral Velocity RMSE	0.0264	0.0197	0.0186
Lateral Velocity MD	0.427	0.4326	0.04327
Yaw Rate RMSE	0.0186	0.0163	0.0153
Yaw Rate MD	0.2885	0.2986	0.2984

The results highlight the performance of the Online DMD model under sharp turn conditions, where oscillatory modeling errors are observed for all controllers, particularly during the U-turn. While the magnitude of these oscillations is not very high, making the results acceptable, it is important to note that real-life applications in noisier environments may amplify these effects.

Despite the oscillations, the maximum deviation values indicate that the errors do not deviate significantly from the actual results, suggesting reasonable accuracy. Notably, the 10-step MPC controller again delivers better performance compared to the 30-step MPC and LQR controllers. This consistent improvement highlights the influence of the prediction and control horizon parameters on the Online DMD identification process.

5.3.3 Control Evaluation

The control performance is evaluated based on the **lateral deviation**, **relative yaw angle errors**, and **steering angle input**, which collectively measure the controllers' ability to maintain the lane center, align the vehicle with the road curvature, and regulate control effort. Figures 5.12, 5.13, and 5.14 present these metrics for each controller, while Table 5.8 summarizes the RMSE and Maximum Deviation (MD) values for quantitative analysis.

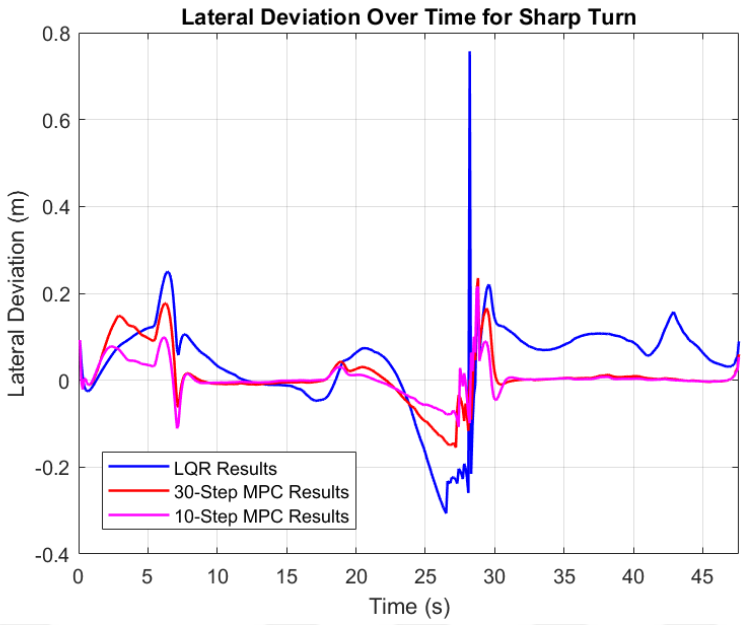


Figure 5.12: Lateral deviation for the sharp turn scenario.

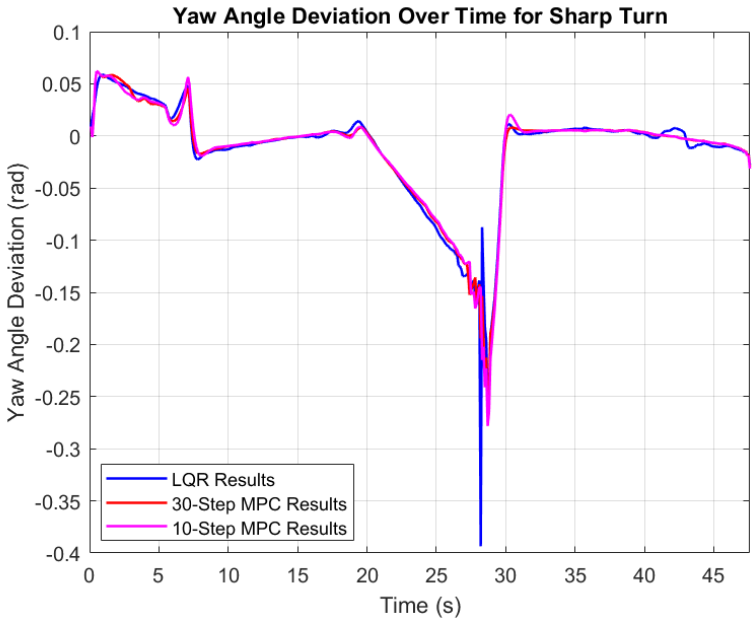


Figure 5.13: Relative yaw angle for the sharp turn scenario.

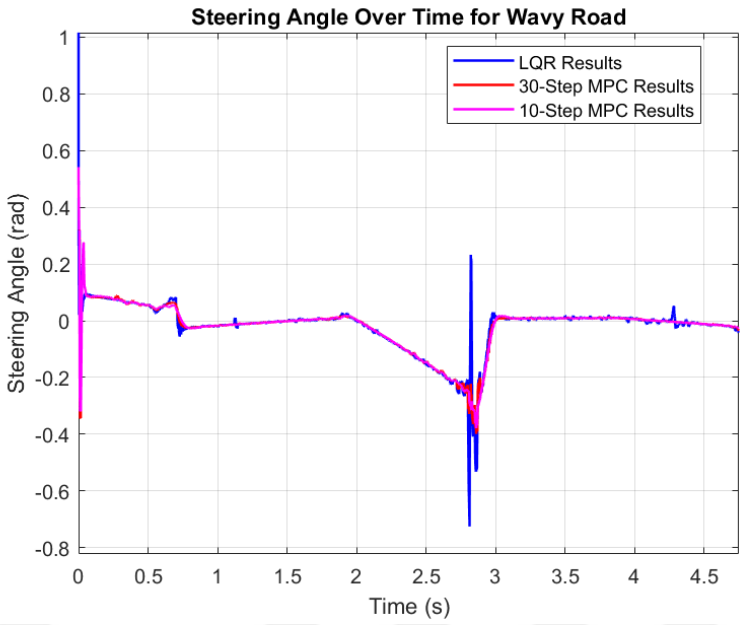


Figure 5.14: Steering angle input over time for the sharp turn scenario.

Table 5.8: Lateral deviation and yaw angle errors for the sharp turn scenario.

Metric	LQR	30-Step MPC	10-Step MPC
Lateral Deviation RMSE	0.0264	0.0197	0.0186
Lateral Deviation MD	0.427	0.4326	0.04327
Yaw Deviation RMSE	0.0186	0.0163	0.0153
Yaw Deviation MD	0.2885	0.2986	0.2984

The results highlight the following key observations:

- During the **sharp turn**, the LQR controller struggles to recover quickly after the turn, exhibiting significantly larger lateral deviation and yaw angle errors compared to the MPC controllers. This is evident in both the figures and the table, where the lateral deviation MD for the LQR reaches 0.427.
- The **30-step MPC** and **10-step MPC** controllers perform much better, with lateral deviation values remaining relatively small throughout the turn. However, the maximum deviation for lateral deviation in the 30-step MPC (0.4326) is slightly higher compared to the 10-step MPC (0.04327). This suggests that the longer prediction horizon may introduce some instability in handling sudden sharp edges.
- The **relative yaw angle** errors are more oscillatory for the LQR controller, particularly during and immediately after the turn. Both MPC controllers, on the other hand, demonstrate smoother yaw alignment, with the 10-step MPC achieving the smallest RMSE of 0.0153.

- The close-up sections of the trajectory visualization confirm that the **10-step MPC** recovers faster and more accurately, particularly during the exit from the sharp turn, whereas the LQR exhibits a longer settling time.
- The **steering angle input** (Figure 5.14) shows oscillations for all controllers during the peak curvature of the sharp turn. However, the magnitude of these oscillations is considerably larger in the LQR controller, reflecting less efficient control effort. The MPC controllers also display oscillations, but their magnitude is much smaller, particularly in steady-state conditions when there is no turning.
- Minor oscillations in the LQR controller are also observed during steady-state conditions, highlighting inefficiency in maintaining a stable control input. These oscillations could lead to reduced passenger comfort in real-world scenarios.

These findings underscore the challenges posed by abrupt changes in curvature and highlight the advantages of the **10-step MPC** controller in achieving faster recovery and improved tracking performance. The evaluation of steering angle inputs further emphasizes the importance of efficient control effort, with the MPC controllers demonstrating superior handling of sharp turn conditions compared to the LQR controller.

5.4 Case 3: City-Like Road

This section evaluates the performance of the proposed controllers on a city-like road, which incorporates a combination of road features encountered in real-life driving scenarios. The road includes smooth curvatures, sharp edges, abrupt turns, varying curvatures, and flat road sections.

5.4.1 Trajectory Visualization

Figure 5.15 shows the vehicle's trajectories compared to the desired path on the city-like road scenario. The trajectory visualization highlights the controllers' performance across four critical sections designed to assess their response to varying road conditions:

1. Section 1: Near-circular curvature transition.
2. Section 2: Exit of a V-shaped road.
3. Section 3: A sharp-edge 90-degree turn.

4. Section 4: Exit of a smooth-to-sharp curvature transition.

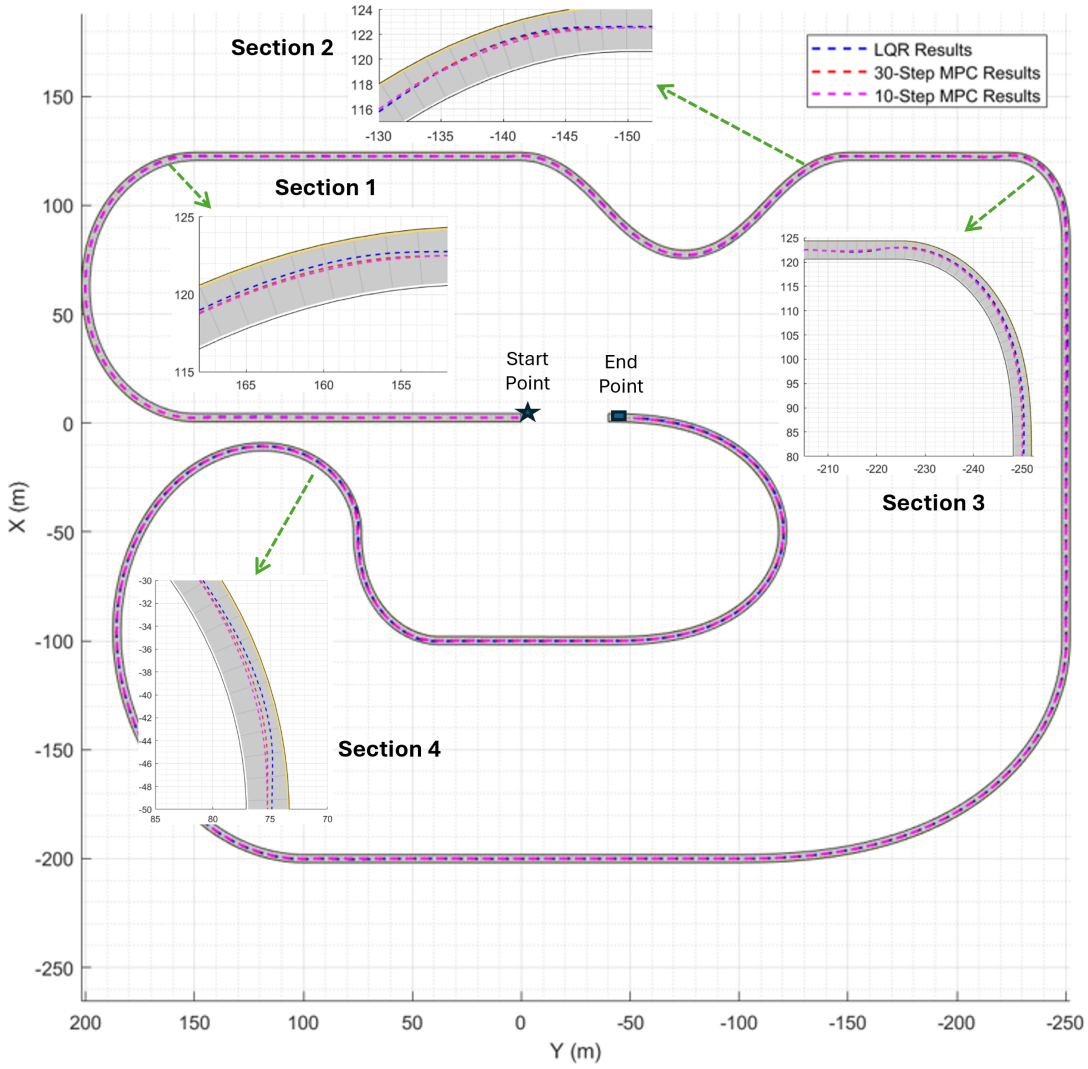


Figure 5.15: Trajectory visualization for the city-like road scenario.

In all sections, the vehicle successfully follows its lane, demonstrating the controllers' ability to adapt to varying road geometries. However, the close-up sections reveal notable differences in tracking performance:

- **LQR Controller:** The LQR controller exhibits visible deviations from the lane center in all sections, particularly at sharper turns. After Section 3 (sharp-edge 90-degree turn), the LQR controller shows a longer settling time, requiring more time to stabilize and return to the lane center.
- **MPC Controllers:** Both the 10-step MPC and 30-step MPC controllers demonstrate superior tracking accuracy compared to the LQR controller. Deviations are smaller across all sections, particularly in sharp transitions and exits.

- **10-Step vs 30-Step MPC:** While the performance difference between the MPC controllers is subtle, careful observation reveals that the 10-step MPC outperforms the 30-step MPC in maintaining tighter tracking accuracy. This difference becomes slightly more evident during sharp transitions and exits, as we will further analyze in the following sections.

The trajectory visualization demonstrates the robustness of the MPC controllers in handling real-world driving conditions, including abrupt geometry changes and transitions, while the LQR controller struggles to match their performance in sharper sections.

5.4.2 Model Evaluation

The accuracy of the Online DMD model is evaluated in terms of its ability to estimate the lateral velocity and yaw rate of the vehicle. The modeling errors are analyzed using RMSE and maximum deviation values.

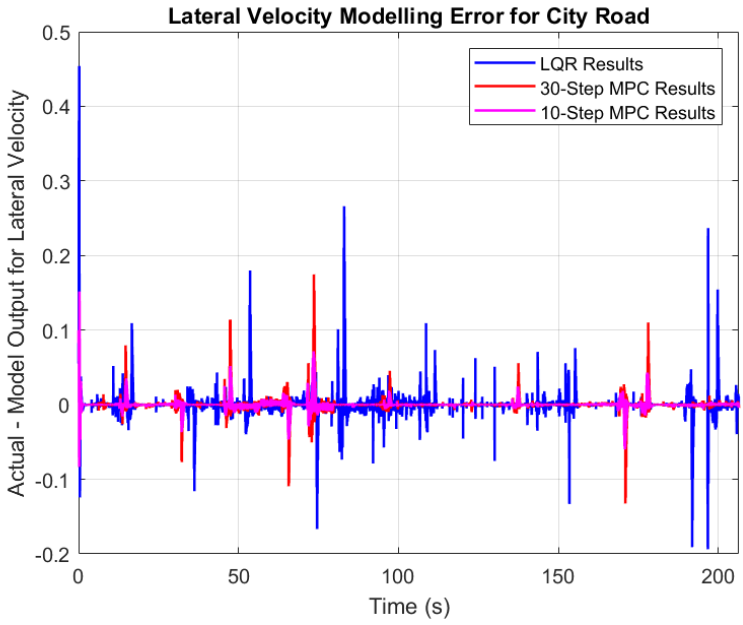


Figure 5.16: Real-time modeling error for lateral velocity on the city-like road scenario.

Figure 5.17: Real-time modeling error for yaw rate on the city-like road scenario.

Table 5.9 summarizes the RMSE and maximum deviation values for lateral velocity and yaw rate modeling errors.

Table 5.9: Modeling errors for the city-like road scenario.

Metric	LQR	30-Step MPC	10-Step MPC
Lateral Velocity RMSE	0.0136	0.0065	0.0047
Lateral Velocity MD	0.4541	0.1748	0.1519
Yaw Rate RMSE	0.0092	0.0049	0.0036
Yaw Rate MD	0.2982	0.1299	0.1042

The results highlight the performance of the Online DMD model under the city-like road scenario, which combines smooth curves, sharp turns, and straight sections. Modeling errors are observed to vary across these road profiles, with sharp curvature transitions leading to slightly higher deviations.

The LQR controller consistently exhibits the largest modeling errors for both lateral velocity and yaw rate, particularly at sharp edges and transitions. This aligns with its limited ability to adapt quickly to changing road conditions. In contrast, the MPC controllers demonstrate superior modeling accuracy, with the 10-step MPC again outperforming the 30-step MPC.

The RMSE and maximum deviation values in Table 5.9 further emphasize the advantage of the MPC controllers. The 10-step MPC achieves the smallest errors, suggesting that its shorter prediction and control horizons provide better real-time adaptation to dynamic conditions. While the errors remain within acceptable limits

for all controllers, the results reinforce the sensitivity of the Online DMD identification process to the choice of controller parameters.

5.4.3 Control Evaluation

The control performance is assessed based on the **lateral deviation**, **relative yaw angle errors**, and **steering angle input**, which measure the controllers' ability to maintain the lane center, align the vehicle with the road's curvature, and regulate control effort. Figures 5.18, 5.19, and 5.20 present these metrics for each controller, while Table 5.10 summarizes the RMSE and Maximum Deviation (MD) values for quantitative analysis.

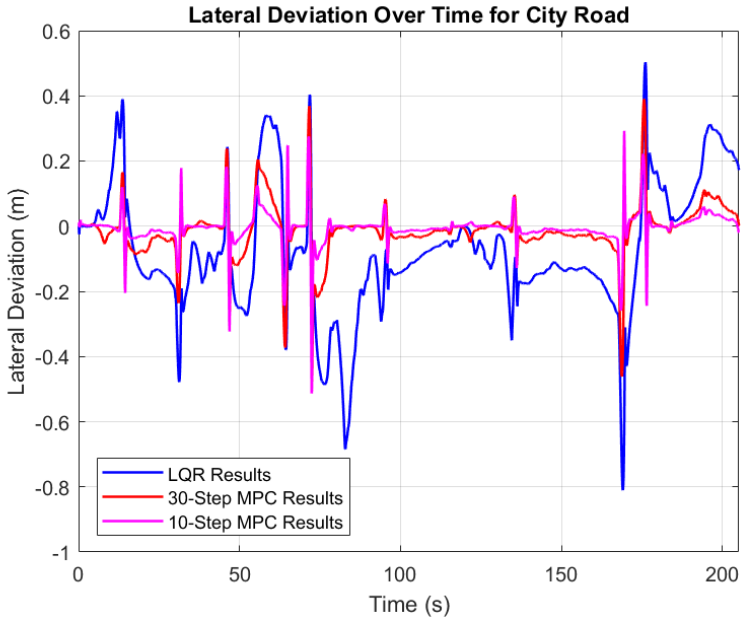


Figure 5.18: Lateral deviation for the city-like road scenario.

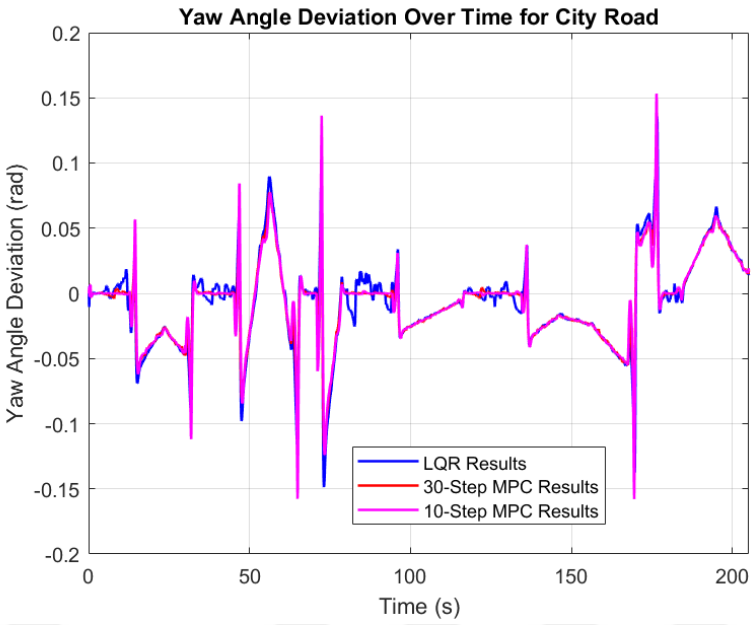


Figure 5.19: Relative yaw angle for the city-like road scenario.

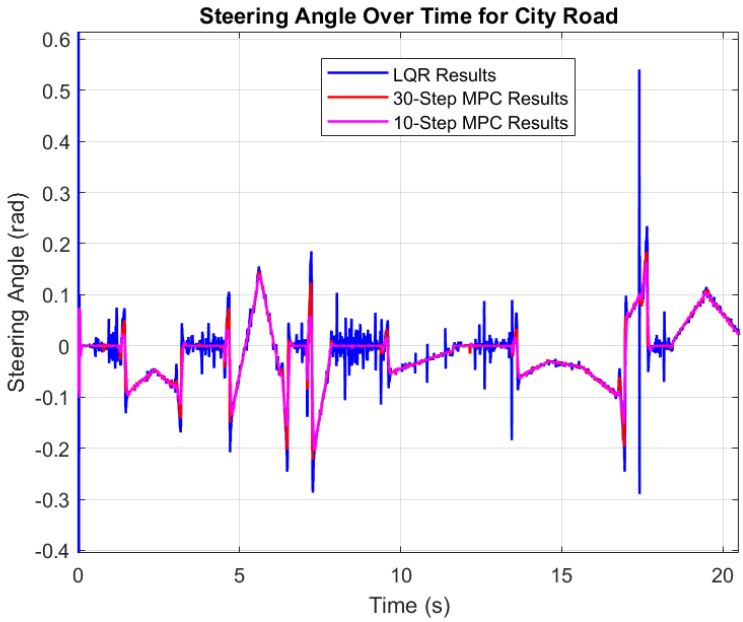


Figure 5.20: Steering angle input over time for the city-like road scenario.

Table 5.10: Lateral deviation and yaw angle errors for the city-like road.

Metric	LQR	30-Step MPC	10-Step MPC
Lateral Deviation RMSE	0.211	0.0818	0.0526
Lateral Deviation MD	0.8103	0.4894	0.5133
Yaw Deviation RMSE	0.0346	0.0312	0.032
Yaw Deviation MD	0.1487	0.1346	0.1578

The results highlight the following key observations:

- In the **city-like road** scenario, which incorporates smooth curves, sharp edges, and transitions, the LQR controller exhibits the largest **lateral deviation** and **relative yaw angle** errors. This is evident in both the figures and Table 5.10, where the Maximum Deviation (MD) values for lateral deviation reach 0.8103, highlighting its struggle to handle abrupt changes in curvature.
- For the **MPC controllers**, the lateral deviation remains significantly lower. The **10-step MPC** shows the smallest RMSE value of 0.0526, performing slightly better overall compared to the 30-step MPC, which has an RMSE of 0.0818.
- Interestingly, the Maximum Deviation (MD) for lateral deviation in the 10-step MPC is slightly higher than in the 30-step MPC. This could indicate small overshoots during sharp transitions, despite the controller maintaining overall better tracking accuracy.
- In terms of **relative yaw angle**, all controllers exhibit fluctuations, with the LQR controller showing the highest RMSE of 0.0346. Both MPC controllers perform comparably, with the 10-step MPC again showing minor improvements over the 30-step MPC.
- The **steering angle input** (Figure 5.20) shows significant oscillations for the LQR controller, particularly after experiencing curvatures. While the MPC controllers display a similar pattern of control input (increasing and decreasing steering angles), the LQR controller exhibits visible oscillations in both transient and steady-state conditions.
- The larger magnitude of oscillations in the LQR controller highlights inefficiency in maintaining smooth control effort. This could lead to reduced passenger comfort in real-world scenarios. The MPC controllers demonstrate better efficiency, with smaller oscillations and more stable behavior overall.

These findings confirm the robustness and adaptability of the **MPC controllers** in handling complex, real-world driving scenarios. The analysis of steering angle inputs further emphasizes the superior control effort of the MPC controllers compared to the LQR controller, making them more suitable for practical applications.

5.5 RLS and Online DMD for Sharp Turn

The sharp turn scenario is selected to compare the performance of controllers using Recursive Least Squares and Online Dynamic Mode Decomposition. This aggressive

maneuver challenges both system identification and control strategies.

Figures 5.21, 5.22, and 5.23 illustrate the lateral deviation, yaw angle deviation, and steering angle inputs, respectively, for the sharp turn scenario. Table 5.11 summarizes the RMSE and Maximum Deviation values for lateral and yaw deviations for both Online DMD and RLS-based controllers.

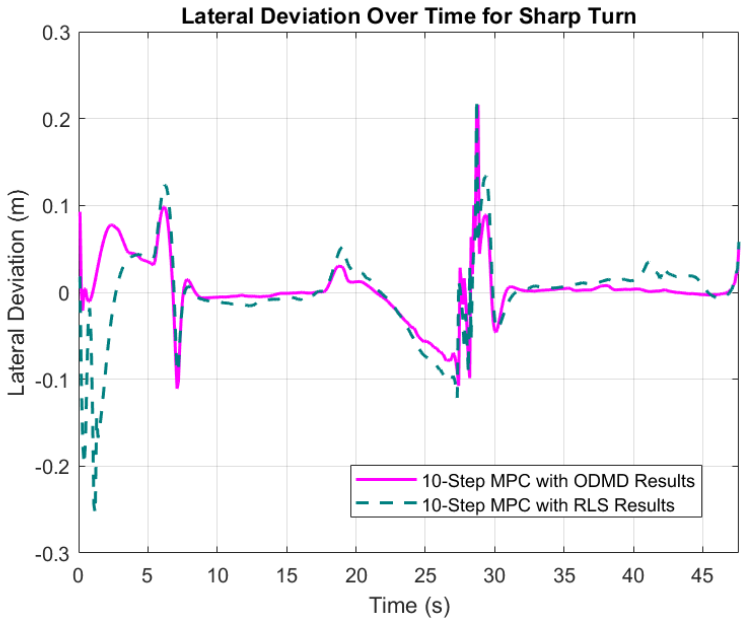


Figure 5.21: Lateral deviation for the sharp turn scenario using Online DMD and RLS.

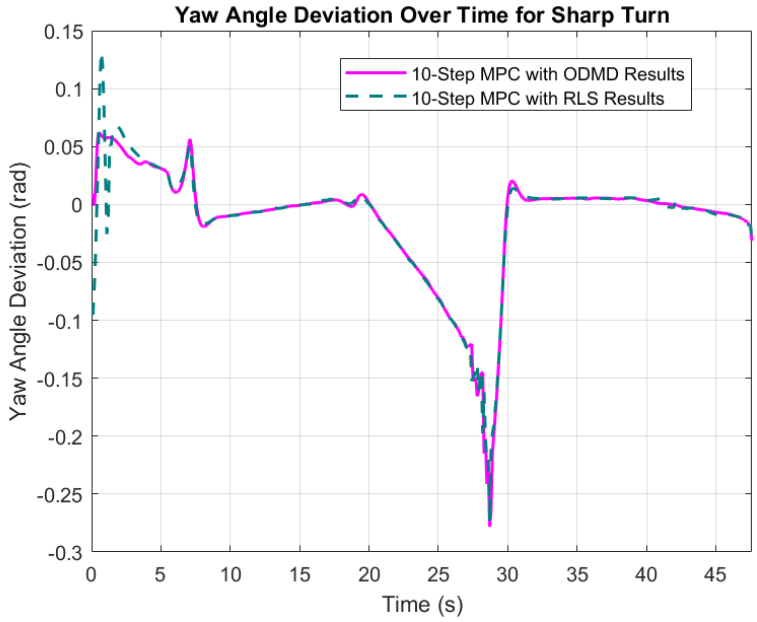


Figure 5.22: Yaw angle deviation for the sharp turn scenario using Online DMD and RLS.

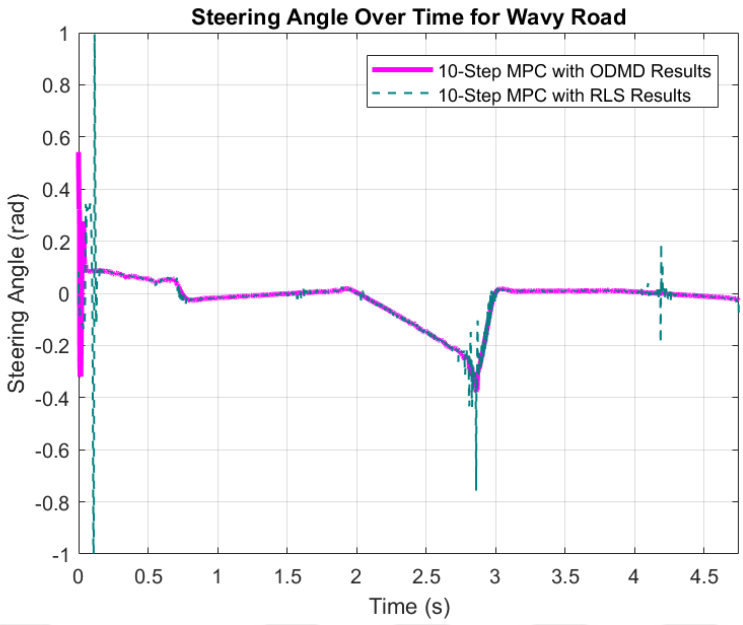


Figure 5.23: Steering angle input for the sharp turn scenario using Online DMD and RLS.

Table 5.11: Lateral and yaw angle errors for the sharp turn scenario using Online DMD and RLS.

Metric	10-Step MPC (ODMD)	10-Step MPC (RLS)
Lateral Deviation RMSE	0.0356	0.0487
Lateral Deviation MD	0.2157	0.2543
Yaw Deviation RMSE	0.0512	0.0513
Yaw Deviation MD	0.2782	0.2734

The results indicate the following key observations:

- The **lateral deviation** errors are slightly higher for the RLS-based controller compared to the Online DMD-based controller. The RMSE increases from 0.0356 to 0.0487, and the Maximum Deviation (MD) increases from 0.2157 to 0.2543.
- The **yaw angle deviation** results are comparable between the two methods, with minimal differences in RMSE (0.0512 vs. 0.0513) and MD (0.2782 vs. 0.2734), suggesting similar performance.
- The **steering angle inputs** (Figure 5.23) reveal visible oscillations in the RLS-based controller during both sharp turn and steady-state situtations. These oscillations may indicate delayed adaptation in the parameter estimation, as RLS requires convergence over time.

Additional Observations and Tuning Requirements for RLS

The implementation of RLS required careful tuning of parameters and preprocessing steps to ensure stability and adaptability during the simulation. Key challenges and modifications included:

- **Initialization Sensitivity:** RLS is sensitive to the initial values of the covariance matrix (P) and the parameter vector (θ). In this implementation, P was initialized with small values to avoid instability, and θ was initialized with small non-zero values to ensure meaningful parameter updates.
- **Input Excitation:** RLS relies on sufficiently excited input signals to ensure accurate parameter estimation. A low-pass filter with a smoothing factor was applied to the inputs to handle noise and stabilize the estimation process.
- **Forgetting Factor (λ):** The forgetting factor was dynamically adjusted to balance parameter convergence and adaptability. When input excitation was low or the error exceeded a threshold, λ was increased to slow down updates and improve stability.
- **Parameter Update Limitation:** The parameter updates were constrained to a maximum allowable change to prevent divergence during abrupt changes in dynamics. This helped mitigate the inherent convergence delays in RLS.
- **Comparison with ODMD:** Unlike Online DMD, RLS assumes a predefined model structure, which introduced additional constraints on the parameter estimation process. This assumption likely contributed to the observed oscillations in the steering input, as RLS struggled to adapt rapidly to sharp turns.

5.6 Discussion

One key finding across all scenarios is the observed difference in performance between the **10-Step MPC** and **30-Step MPC** controllers. While both controllers significantly outperform the LQR controller, the results indicate that the **prediction capability** of the **Online DMD** model plays a critical role in determining the effectiveness of the prediction horizon. Additionally, the analysis of **control effort** reveals key differences in how each controller manages steering inputs to achieve lane tracking.

- The **Online DMD** model continuously updates the system matrices (A, B) at each time step using the most recent data. However, this real-time update

process does not account for future states or disturbances, resulting in a **short-term predictive capability**.

- As a consequence, controllers with a **shorter prediction horizon**, such as the 10-Step MPC, are able to adapt more effectively to the **immediate system dynamics**, leveraging the most current model updates without relying on long-term predictions.
- On the other hand, the **30-Step MPC**, which optimizes over a longer prediction horizon, depends heavily on the accuracy of the current DMD model to predict system behavior further into the future. Due to the **limited foresight** of the Online DMD model, inaccuracies in the system identification process can accumulate over the prediction horizon, particularly during abrupt changes in curvature or external disturbances.
- The **LQR controller**, which does not utilize any prediction horizon, reacts directly to the current system state using the most recent DMD model. However, its **sensitivity to frequent model updates** and inability to anticipate dynamic changes make it more vulnerable to inaccuracies and external disturbances, leading to poorer performance in scenarios with sharp turns or rapidly changing conditions.
- The **control effort analysis** reveals that the LQR controller exhibits visible oscillations in the steering angle, particularly during steady-state operation and after sharp turns. These oscillations, while small in magnitude, reflect the controller's inability to dampen frequent updates from the Online DMD model. In contrast, the MPC controllers demonstrate smoother control inputs, with the 10-Step MPC showing the least oscillations.
- The analysis of the **RLS-based approach** for system identification in the sharp turn scenario reveals that RLS exhibits a delayed response in parameter estimation compared to Online DMD. This delay is reflected in higher lateral deviation RMSE and visible oscillations in steering inputs, particularly during aggressive maneuvers. These findings emphasize that RLS, while effective in stable and well-defined conditions, struggles to adapt as quickly as Online DMD under dynamic and aggressive scenarios.

This behavior demonstrates the strong influence of the Online DMD model's **short-term predictive limitations** on MPC performance. A shorter prediction horizon inherently aligns better with the real-time structure of the Online DMD updates. This tradeoff shows the importance of selecting prediction horizons that match the capabilities of the updating system model.

The results further highlight that:

- The **LQR controller** struggles in scenarios with sharp turns and varying curvatures. Its inability to adapt to changes in the road dynamics or external disturbances results in larger lateral deviation and yaw errors compared to the MPC controllers. Additionally, its oscillatory steering behavior can lead to higher control effort, potentially causing instability in challenging scenarios.
- The **10-Step MPC controller**, leveraging the most recent Online DMD updates, consistently demonstrates superior performance by maintaining better lane tracking and yaw stability in all test cases. This result aligns with the short-term accuracy of the Online DMD model and its reduced sensitivity to model inaccuracies. Its smoother control inputs further reduce unnecessary oscillations, improving both control stability and efficiency.
- The **30-Step MPC controller**, while performing well in most scenarios, exhibits higher sensitivity to inaccuracies in the Online DMD updates during scenarios with abrupt changes in curvature. This sensitivity is reflected in slightly larger lateral deviation values compared to the 10-Step MPC.
- The **RLS-based approach**, while providing reasonable performance, demonstrates limitations in its ability to adapt quickly to dynamic conditions. The slightly higher lateral deviation RMSE and observed oscillations in steering input shows the delayed parameter convergence related to RLS.

The analysis of control effort reinforces these findings. The smoother steering inputs provided by the MPC controllers not only improve lane tracking and yaw stability but also reduce unnecessary oscillations, ensuring more efficient and stable control. The LQR controller, while functional, shows more aggressive oscillations in steering input, highlighting its limitations in handling dynamic updates from the Online DMD model. The RLS-based approach introduces additional oscillations in steering inputs, further emphasizing its slower adaptability compared to Online DMD.

Overall, the findings emphasize the importance of selecting appropriate prediction horizons when using adaptive models like Online DMD. While longer horizons can theoretically improve control performance, their effectiveness is limited by the model's predictive accuracy. Furthermore, minimizing control effort through smoother steering inputs plays a crucial role in enhancing vehicle stability and driver comfort, with the 10-Step MPC providing the most balanced and effective performance in this regard. The comparison with RLS further underscores the advantages of Online DMD for scenarios requiring rapid adaptation to dynamic changes.

6

Conclusion and Future Works

This study explored the integration of the Online Dynamic Mode Decomposition (DMD) algorithm with advanced control strategies for autonomous vehicle lane-keeping applications. The proposed approach was evaluated under three distinct driving scenarios—constant curvature, sharp turn, and city-like roads—designed using a high-fidelity simulator. Controllers, including Linear Quadratic Regulator (LQR) and Model Predictive Control (MPC), were assessed based on their ability to maintain lane tracking, adapt to varying road conditions and disturbances, and manage control effort efficiently.

Conclusion

Several important insights into the performance of the proposed control strategies and the Online Dynamic Mode Decomposition (DMD) model are as follows:

- The **Online DMD model** effectively updated the system matrices (A, B) in real-time, enabling adaptive control in dynamic environments. However, its **limited predictive capability** influenced controller performance, particularly for controllers relying on longer prediction horizons.
- The **10-step MPC controller** outperformed the 30-step MPC and LQR controllers across all scenarios, showcasing superior tracking performance and adaptability. Its shorter prediction horizon allowed it to respond more effectively to immediate system dynamics, while the 30-step MPC struggled with accumulated errors due to the model's predictive limitations.
- The **LQR controller** exhibited the poorest performance in handling high-curvature scenarios. Its reliance on instantaneous model updates, combined with the absence of predictive capability, limited its ability to adapt to sharp changes in road geometry. Additionally, it demonstrated noticeable **oscilla-**

tions in steering input, which can contribute to higher control effort and reduced stability in certain scenarios.

- The addition of **lateral wind disturbances** demonstrated the robustness of the MPC controllers, as they maintained minimal performance degradation under such challenging conditions.
- The analysis of **control effort** showed differences between the controllers. The LQR controller exhibited frequent oscillations in steering input, also in steady-state conditions, which can lead to higher energy consumption and potential wear on the vehicle's steering system. In contrast, the MPC controllers demonstrated smoother control inputs, with the 10-step MPC achieving the best balance between performance and control effort.

The study provides several notable contributions:

- An integrated approach combining **Online DMD** and **data-driven control strategies** for autonomous vehicle applications.
- A comprehensive **evaluation framework** incorporating realistic scenarios, disturbances, and model errors to assess controller performance effectively.
- New insights into the effects of **prediction horizon**, model accuracy, and control effort on overall control performance, contributing to the design of more robust adaptive controllers.

Despite the promising results, this study has certain limitations:

- The **predictive accuracy** of the Online DMD model is constrained to short-term dynamics, limiting its effectiveness for controllers requiring longer prediction horizons.
- All experiments were conducted in a **simulation-based environment**, which does not fully replicate real-world complexities.
- **Noise and disturbance effects** were limited to controlled scenarios, which may not fully represent the challenges of real-world applications.
- The study did not explicitly evaluate the long-term impact of **control effort** on energy efficiency or system wear, which would be important for real-world implementations.

These findings highlight the potential of integrating Online DMD with adaptive controllers while emphasizing the importance of control effort analysis and the need for further validation through real-world testing and refinement to address the observed limitations.

Future Works

Future research directions include:

- Enhancing the predictive accuracy of **Online DMD** through adaptive filtering or hybrid modeling approaches that combine data-driven and physics-based methods. Additionally, implementing **windowed approaches** or techniques like forgetting factors can improve the model's adaptability and robustness in dynamic environments.
- Expanding the evaluation framework to include multi-vehicle interactions, traffic dynamics, and varying environmental conditions such as rain, fog, and icy roads.
- Conducting real-world experiments to validate the robustness and scalability of the proposed methods under practical conditions.
- Exploring alternative control strategies, such as reinforcement learning or neural network-based controllers, to complement the capabilities of Online DMD.
- Studying the effects of varying **longitudinal speeds** on the proposed strategy to assess its performance under different vehicle velocities and dynamic operating conditions. This will provide further insights into the applicability and limitations of the current approach.

By addressing these limitations and building upon the findings of this study, future research can further advance the development of robust and adaptive control systems for autonomous vehicles, paving the way for safer and more efficient transportation solutions.

Bibliography

[1] Victor Adetola, Dale DeHaan, and Martin Guay. "Robust adaptive MPC for constrained uncertain nonlinear systems". In: *International Journal of Adaptive Control and Signal Processing* 25 (2011), pp. 155–168. DOI: 10.1002/acs.1206.

[2] N.H. Amer et al. "Adaptive modified Stanley controller with fuzzy supervisory system for trajectory tracking of an autonomous armoured vehicle". In: *Robotics and Autonomous Systems* 105 (2018), pp. 94–111.

[3] P. J. Baddoo et al. "Physics-informed dynamic mode decomposition". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 479.2271 (2023). DOI: 10.1098/rspa.2022.0576. URL: https://doi.org/10.1098/rspa.2022.0576.

[4] L. Bascetta and G. Ferretti. "LFT-Based identification of lateral vehicle Dynamics". In: *IEEE Transactions on Vehicular Technology* 71.2 (2021), pp. 1349–1362. DOI: 10.1109/tvt.2021.3134591.

[5] I. Batkovic et al. "Real-time constrained trajectory planning and vehicle control for proactive autonomous driving with road users". In: *IEEE International Conference*. 2019, pp. 256–262.

[6] F. Borrelli et al. "MPC-based approach to active steering for autonomous vehicle systems". In: *International Journal of Vehicle Autonomous Systems* 3.2 (2005), pp. 265–291.

[7] S. L. Brunton and J. N. Kutz. *Data-Driven Science and Engineering*. Cambridge University Press, 2019. DOI: 10.1017/9781108380690. URL: https://doi.org/10.1017/9781108380690.

[8] D. M. Buede and W. D. Miller. *The Engineering Design of Systems: Models and Methods*. John Wiley & Sons, 2024.

[9] M. Buehler, K. Iagnemma, and S. Singh. *The 2005 DARPA Grand Challenge: The Great Robot Race*. 1st. New York: Springer, 2007.

[10] S.F. Campbell. "Steering control of an autonomous ground vehicle with application to the DARPA Urban Challenge". [Online]. Available: http://dspace.mit.edu/handle/1721.1/42301. PhD thesis. Massachusetts Institute of Technology, 2007.

[11] C.Y. Chen, T.H.S. Li, and Y.C. Yeh. "Ep-based kinematic control and adaptive fuzzy sliding-mode dynamic control for wheeled mobile robots". In: *Information Sciences* 179.1 (2009), pp. 180–195.

[12] European Commission. *Roadmap to a Single European Transport Area*. Tech. rep. Retrieved April 12, 2018. 2011. URL: http://ec.europa.eu/transport/strategies/facts-and-figures/index_en.htm.

[13] European Road Safety Observatory. *Traffic Safety Basic Facts – Heavy Goods Vehicles and Buses*. Tech. rep. Technical report. 2016.

[14] P. Falcone et al. "MPC-based yaw and lateral stabilisation via active front steering and braking". In: *Vehicle System Dynamics* 46.sup1 (2008), pp. 611–628.

[15] P. Falcone et al. "Predictive Active Steering Control for Autonomous Vehicle Systems". In: *IEEE Transactions on Control Systems Technology* 15.3 (2007), pp. 566–580.

[16] F. Hamerlain et al. "Trajectory tracking of a car-like robot using second order sliding mode control". In: *European Control Conference (ECC)*. Kos, 2007.

[17] G.M. Hoffmann et al. "Autonomous automobile trajectory tracking for off-road driving: controller design, experimental validation and racing". In: *American Control Conference*. New York, 2007.

[18] C. Hu et al. "Robust H output-feedback control for path following of autonomous ground vehicles". In: *Mechanical Systems and Signal Processing* 70-71 (2016), pp. 414–427.

[19] X. Huang et al. "Robust Weighted Gain-Scheduling H_∞ Vehicle Lateral Motion Control With Considerations of Steering System Backlash-Type Hysteresis". In: *IEEE Transactions on Control Systems Technology* 22.5 (2014), pp. 1740–1753.

[20] The MathWorks Inc. *Automated Driving Toolbox version: 23.2 (R2023b)*. Natick, Massachusetts, United States, 2023. URL: https://www.mathworks.com.

[21] The MathWorks Inc. *Model Predictive Control Toolbox version: 23.2 (R2023b)*. Natick, Massachusetts, United States, 2023. URL: https://www.mathworks.com.

[22] The MathWorks Inc. *Vehicle Dynamics Blockset version: 23.2 (R2023b)*. Natick, Massachusetts, United States, 2023. URL: https://www.mathworks.com.

[23] Rolf Isermann and Marco Munchhof. *Identification of Dynamic Systems: An Introduction with Applications*. Springer Science, 2010. ISBN: 978-3-540-78878-2.

[24] G. E. Karniadakis et al. "Physics-informed machine learning". In: *Nature Reviews Physics* 3 (2021), pp. 422–440. DOI: 10.1038/s42254-021-00314-5. URL: https://doi.org/10.1038/s42254-021-00314-5.

[25] A. Katriniok et al. "Optimal vehicle dynamics control for combined longitudinal and lateral autonomous vehicle guidance". In: *European Control Conference (ECC)*. Zurich, 2013.

[26] A. Katriniok and D. Abel. "LTV-MPC approach for lateral vehicle guidance by front steering at the limits of vehicle dynamics". In: *50th IEEE Conference on Decision and Control and European Control Conference*. Orlando, 2011.

[27] T. Keviczky et al. "Predictive control approach to autonomous vehicle steering". In: *American Control Conference*. Minneapolis, 2006.

[28] H.K. Khalil and J.W. Grizzle. *Nonlinear Systems*. Vol. 3. Upper Saddle River: Prentice Hall, 2002.

[29] D. Kim, J. Kang, and K. Yi. "Control strategy for high-speed autonomous driving in structured road". In: *14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. Washington, DC, 2011.

[30] G. Kim et al. "Vehicle's lateral motion control using dynamic mode decomposition model predictive control for unknown model". In: *International Journal of Automotive Technology* (2024). DOI: 10.1007/s12239-024-00074-y.

[31] J. S. Kim, Y. S. Quan, and C. C. Chung. "Data-Driven Modeling and Control for Lane Keeping System of Automated Driving Vehicles: Koopman Operator Approach". In: *2022 22nd International Conference on Control, Automation and Systems (ICCAS)*. 2022. DOI: 10.23919/iccas55662.2022.10003764.

[32] Y. Liang et al. "Yaw rate tracking-based path-following control for four-wheel independent driving and four-wheel independent steering autonomous vehicles considering the coordination with dynamics stability". In: *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 235.1 (2021), pp. 260–272.

[33] R. Marino, S. Scalzi, and M. Netto. "Nested PID steering control for lane keeping in autonomous vehicles". In: *Control Engineering Practice* 19.12 (2011), pp. 1459–1467.

[34] N. Mohajer et al. "Evaluation of the path tracking performance of autonomous vehicles using the universal motion simulator". In: *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. Miyazaki, 2018.

[35] H. Mouri and H. Furusho. "Automatic path tracking using linear quadratic control theory". In: *Proceedings of Conference on Intelligent Transportation Systems*. Boston, 1997.

[36] R.M. Murray et al. "Optimization-Based Control". In: *Dynamic Systems*. California: California Institute of Technology, 2009, pp. 111–128.

[37] J. Ni, J. Hu, and C. Xiang. "Robust Path Following Control at Driving/Handling Limits of an Autonomous Electric Racecar". In: *IEEE Transactions on Vehicular Technology* 68.6 (2019), pp. 5518–5526.

[38] F. Noé and F. Nuske. "A variational approach to modeling slow processes in stochastic dynamical systems". In: *Multiscale Modeling & Simulation* 11.2 (2013), pp. 635–655.

[39] A. Ollero and G. Heredia. "Stability analysis of mobile robot path tracking". In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*. Pittsburgh, 1995.

[40] H. B. Pacejka. *Tyre and Vehicle Dynamics*. Elsevier Science, 2005. ISBN: 9780080543338.

[41] J. L. Proctor, S. L. Brunton, and J. N. Kutz. "Dynamic Mode Decomposition with Control". In: *SIAM Journal on Applied Dynamical Systems* 15.1 (2016), pp. 142–161. DOI: 10.1137/15m1013857. URL: https://doi.org/10.1137/15m1013857.

[42] R. Rajamani. “Lateral vehicle dynamics”. In: *Vehicle Dynamics and Control*. Boston: Springer, 2012, pp. 15–46.

[43] Rajesh Rajamani. *Vehicle Dynamics and Control*. 1st. Springer US, 2005.

[44] A.L. Rankin, C.D. Crane III, and D.G. Armstrong II. “Evaluating a PID, pure pursuit, and weighted steering controller for an autonomous land vehicle”. In: *Mobile Robots XII*. Ed. by D.W. Gage. Vol. 3210. Bellingham: SPIE, 1998.

[45] J. Rawlings. “Tutorial overview of model predictive control”. In: *IEEE Control Systems* 20.3 (2000), pp. 38–52. DOI: 10.1109/37.845037.

[46] L.L. Scharf, W.P. Harthill, and P.H. Moose. “A comparison of expected flight times for intercept and pure pursuit missiles”. In: *IEEE Transactions on Aerospace and Electronic Systems* AES-5.4 (1969), pp. 672–673.

[47] S. S. Shadrin and A. A. Ivanova. “Analytical Review of Standard SAE J3016 ”Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles” with Latest Updates”. In: *SAE International* 10 (2019). URL: https://www.sae.org/standards/content/j3016_201806/.

[48] J.J.E. Slotine and W. Li. *Applied Nonlinear Control*. Englewood Cliffs: Prentice Hall, 1991.

[49] J.M. Snider. *Automatic steering methods for autonomous automobile path tracking*. Tech. rep. CMU-RI-TR-09-08. Robotics Institute, Pittsburgh, PA, 2009.

[50] N. A. Spielberg, M. Brown, and J. C. Gerdes. “Neural network model predictive motion control applied to automated driving with unknown friction”. In: *IEEE Transactions on Control Systems Technology* 30.5 (2021), pp. 1934–1945. DOI: 10.1109/tcst.2021.3130225.

[51] P. Stano et al. “Model predictive path tracking control for automated road vehicles: A review”. In: *Annual Reviews in Control* 55 (2022), pp. 194–236. DOI: 10.1016/j.arcontrol.2022.11.001. URL: https://doi.org/10.1016/j.arcontrol.2022.11.001.

[52] S. Thrun et al. “Stanley: The robot that won the DARPA grand challenge”. In: *The 2005 DARPA Grand Challenge: The Great Robot Race*. Heidelberg: Springer, 2007, pp. 1–43.

[53] J. H. Tu et al. “On dynamic mode decomposition: Theory and applications”. In: *Journal of Computational Dynamics* 1.2 (2014), pp. 391–421. DOI: 10.3934/jcd.2014.1.391. URL: https://doi.org/10.3934/jcd.2014.1.391.

[54] United Nations Population Division. *World Population Prospects: The 2017 Revision*. Tech. rep. Technical report. 2017.

[55] B. A. H. Vicente, S. S. James, and S. R. Anderson. “Linear system identification versus Physical Modeling of Lateral–Longitudinal Vehicle Dynamics”. In: *IEEE Transactions on Control Systems Technology* 29.3 (2020), pp. 1380–1387. DOI: 10.1109/tcst.2020.2994120.

[56] R. Wallace et al. “First results in robot road-following”. In: *Proceedings of the 9th International Joint Conference on Artificial Intelligence*. San Francisco, 1985.

[57] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley. "A Data–Driven approximation of the Koopman operator: Extending dynamic mode decomposition". In: *Journal of Nonlinear Science* 25.6 (2015), pp. 1307–1346. DOI: 10.1007/s00332-015-9258-5. URL: https://doi.org/10.1007/s00332-015-9258-5.

[58] D. Wu et al. "Fast velocity trajectory planning and control algorithm of intelligent 4wd electric vehicle for energy saving using time-based MPC". In: *IET Intelligent Transport Systems* 13.1 (2019), pp. 153–159.

[59] J.M. Yang and J.H. Kim. "Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots". In: *IEEE Transactions on Robotics and Automation* 15.3 (1999), pp. 578–587.

[60] N. Zhang, J. Ni, and J. Hu. "Robust H_∞ state feedback control for handling stability of intelligent vehicles on a novel all-wheel independent steering mode". In: *IET Intelligent Transport Systems* 13.10 (2019), pp. 1579–1589.