PSO Based Linear Parameter Varying Model Predictive Control for Trajectory Tracking of Autonomous Vehicles

Chala Abdulkadir Kedir

A Thesis Submitted to

The Department of Electrical Power and Control Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfilment of the Requirement for the Degree of Master's in Electrical Power and Control Engineering (Control Engineering)

Office of Graduate Studies

Adama Science and Technology University

July, 2023

Adama, Ethiopia

PSO Based Linear Parameter Varying Model Predictive Control for Trajectory Tracking of Autonomous Vehicles

Chala Abdulkadir Kedir

Advisor: Prof. Sam Sun Ma

A Thesis submitted to

The Department of Electrical Power and Control Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Electrical Power and Control engineering (Control Engineering)

Office of Graduate Studies

Adama Science and Technology University

July, 2023

Adama, Ethiopia

DECLARATION

I hereby declare that this Master Thesis entitled "**PSO Based Linear Parameter Varying Model Predictive Control for Trajectory Tracking of Autonomous Vehicles**" is my original work. That is, it has not been submitted for the award of any academic degree, diploma or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation.

Chala Abdulkadir Kedir	_________________	_________________
Name of the student	Signature	Date

RECOMMENDATION OF ADVISOR

I, the advisor of this thesis, hereby certify that I have read the revised version of the thesis entitled “**PSO Based Linear Parameter Varying Model Predictive Control for Trajectory Tracking of Autonomous Vehicles**” prepared under my guidance by **Chala Abdulkadir Kedir**, submitted in partial fulfillment of the requirements for the degree of Mater's of Science in **Control Engineering**. Therefore, I recommend the submission of a revised version of the thesis to the department following the applicable procedures.

________________________	________________________	____________
Advisor	Signature	Date

APPROVAL SHEET

I, the advisor of the thesis entitled **"PSO Based Linear Parameter Varying Model Predictive Control for Trajectory Tracking of Autonomous Vehicles"** and developed by **Chala Abdulkadir Kedir**, hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

______________________	______________________	____________
Advisor	Signature	Date

We, the undersigned, members of the Board of Examiners of the thesis by **Chala Abdulkadir Kedir** have read and evaluated the thesis work entitled "**PSO Based Linear Parameter Varying Model Predictive Control for Trajectory Tracking of Autonomous Vehicles**" and examined the candidate during open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in **Control Engineering**.

______________________	______________________	____________
Chairperson	Signature	Date
______________________	______________________	____________
Internal Examiner	Signature	Date
______________________	______________________	____________
External Examiner	Signature	Date

Finally, approval and acceptance of the thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

______________________	______________________	____________
Department Head	Signature	Date
______________________	______________________	____________
School Dean	Signature	Date
______________________	______________________	____________
Office of Postgraduate Studies, Dean	Signature	Date

ACKNOWLEDGEMENT

First and foremost, I would like to express my heartfelt gratitude to the Almighty Allah for granting me the resilience and determination to persevere through the challenging journey of completing this thesis. I am deeply grateful to my advisor, Prof. Sam Sun Ma, for his unwavering guidance, inspiration, and constant support throughout the entire process of my thesis work. His expertise, valuable insights and motivation have been instrumental in shaping the direction and success of this thesis. Additionally, I would like to extend my sincere appreciation and thanks to my brother, as well as all those who have provided assistance and support during this thesis endeavor. Their unwavering belief in me, encouragement, and blessings have been a constant source of strength and motivation. I am truly indebted to their love, support, understanding, and encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

Abbreviations	**Description**
AI	Artificial Intelligence
AV	Autonomous Vehicle
CG	Center of Gravity
CPU	Central Processing Unit
CTVE	Center of Transportation and Vehicle Engineering
DoF	Degree of Freedom
EKF	Extended Kalman Filter
GPS	Global Positioning System
ICE	Internal Combustion Engine
ICR	Instantaneous Center of Rotation
IMU	Inertial Measurement Unit
ISMC	Integral Sliding Mode Control
IT	Information Technology
LDI	Linear Differential Inclusion
LiDAR	Light Detection and Ranging
LPV	Linear Parameter Varying
LTV	Linear Time Variant
MIMO	Multiple Input Multiple Output
ML	Machine Learning
MPC	Model Predictive Control
NMPC	Nonlinear Model Predictive Control
NN	Neural Network
PID	Proportional Integral Derivative
PSO	Particle Swarm Optimization

ABSTRACT

Autonomous vehicles (AV) have gained significant attention and became prominent for research area in recent years due to their potential to revolutionize transportation systems. The AV system is characterized by instability, high nonlinearity and time-varying dynamics. Trajectory tracking control, plays a fundamental role in achieving the basic task of AVs to enable safe and efficient autonomous driving. In this research work, the design of a trajectory tracking control system for AVs using a specific variant of Model Predictive Control (MPC) called Linear Parameter Varying (LPV)-MPC which has emerged as an effective technique for trajectory tracking is designed. The time-varying LPV form of the state space representation is formulated from the mathematical model of the vehicle. This model is based on a nonlinear dynamic bicycle model, which considers constraints and is expressed in road-aligned coordinates. The LPV form allows for the incorporation of time-varying dynamics, providing a more accurate representation of the vehicle's behavior. The designed LPV-MPC controller for AVs is specifically designed to handle constraints in trajectory tracking. To enhance its performance, Particle Swarm Optimization (PSO) is employed as an optimization technique. PSO is used to tune the weighting matrices of the control parameters, optimizing the system's response and improving trajectory tracking performance. To evaluate the effectiveness of the LPV-MPC system, extensive simulation and testing are conducted. The results are compared with the outcomes of a Linear MPC controller. The simulation and testing demonstrate that the LPV-MPC controller outperforms the Linear MPC controller in accurately tracking the desired trajectory, particularly when dealing with nonlinear reference roads. This highlights the capability of the LPV-MPC controller to effectively handle the challenges posed by nonlinear reference trajectories.

Key words: *Autonomous Vehicle, Dynamic model, Model Predictive Control, Linear Parameter Varying, Constraints, Particle Swarm Optimization, Trajectory tracking*

CHAPTER ONE

1 INTRODUCTION

1.1 Background

Autonomous vehicles (AVs), commonly referred to as self-driving cars, are vehicles that can operate and navigate without human intervention. Due to the high frequency of traffic accidents and congestion brought on by drivers' mistakes, the creation of AVs has long been a goal of humanity, and they are likely to become inevitable and important development trends and research areas, playing key roles in intelligent modes of transport. AVs can sense their surroundings and drive on its own. To understand its surroundings, AVs use a combination of sensors, such as LiDAR, radar, sonar, GPS, odometry, and IMUs. Control systems examine sensory data to decide the optimal paths to go, as well as obstacles and essential signs. AVs have the potential to revolutionize transportation in cities, reducing traffic congestion by providing a safe means of transportation (Zhu et al., 2023).

AVs have become a prominent subject in the field of control engineering and are driving the growth of the automotive industry. The transportation sector has experienced a significant transformation due to the rapid advancements in AV technology, prompting widespread interest and investment from academic institutions, technology companies, and automotive manufacturers. However, the journey towards developing AVs has presented its fair share of challenges. As early as 1925, an Electrical Engineer named Francis Houdina introduced the concept of AVs with the remote-control car laying the foundation for the AV revolution. In 1939, General Motors showcased the Faturama at the World's Fair, an AV designed for transporting passengers on urban roads, providing a glimpse into a future where vehicles could navigate autonomously, enhancing convenience and mobility (Liu et al., 2020).

In the 1960s, significant research and development efforts were made to advance AVs. At The Ohio State University, Professor Robert Fenton and his team led the way in developing vehicles for testing steering and speed controllers. Similarly, in Japan, the Mechanical Engineering Laboratory and the National Institute of Advanced Industrial Science and Technology, that were at the forefront of this research developed an AV that employed inductive cables and pickup coils for lateral control, creating a cooperative system between vehicles and infrastructure. Rear-end collision avoidance systems based on roadway vehicle sensors were also developed (Ozguner et al., 2011).

In 1980s, a German researcher Ernst Dickmanns, converted Mercedes-Benz into AV guided by an integrated computer, which played a significant role in AVs advancement. A significant milestone in the development of AVs was the Defense Advanced Research Projects Agency (DARPA) Grand Challenge in 2005. With the racing challenge of a 212 km off-road course, and it marked a crucial moment in the advancement of AV technology. The winning robot, named Stanley and created by Stanford University's team, achieved successful steering control using a kinematic motion model and classical control theory. After that, several companies, including established automakers, tech companies, and startups, are now testing and developing AVs. Although the technology has not yet reached its full potential and not widely used, it is anticipated to have a big impact on transportation in the future (Babu et al., 2019).

In recent years, AVs have drawn a lot of interest for their ability to improve the convenience, safety, and efficiency of road transportation. The rapid development of advanced technologies in robotics, automation, and high-speed communication networks has led to the growth and evolution of AVs. There is a large market developing around the world for AVs, which are expected to be autonomous, connected, electrified, and shared. Longer range, safe and regulated technology and security of personal and vehicle are required for AVs to become more common (K. Nonami et al., 2013). A number of modules must operate in a sequential and structured manner in order to accomplish fully autonomous driving. First, the vehicle needs sensors such as GPS, IMU, cameras, LiDAR, thermographic cameras, odometry, sonar and radars that collect all the vehicle and environment information and are treated to extract information about the vehicle and its surroundings (Liu et al., 2020).

Trajectory tracking, which is the fundamental capability of AVs involves maintaining motion stability and accurately following a desired trajectory within a specified time frame. The primary objective of tracking control is to enable the AVs to closely follow a predefined reference. The tracking controller adjusts the control inputs in real-time to ensure that the system tracks the reference trajectory with minimal deviation. This trajectory is composed of global positions, orientations and vehicle velocities. The tracking controller continuously adjusts the control inputs in real-time based on the computed sequence of references and the vehicle's current position (Guo et al., 2020). The automatic control generates the control inputs for the actuators using the computed sequence of references and the position of the vehicle. The control is the last piece in the sequence of the AV and one of the most important tasks since it is in charge of guaranteeing its motion. With the rapid development and adoption of AVs, high-precision trajectory tracking control has become one of the issues for achieving the numerous capabilities. Accurate trajectory tracking can provide crucial theoretical and practical foundations for AV motion control (Pang et al., 2022).

The study of AVs poses a complex problem that demands interdisciplinary competence. This complexity arises from the dynamic nature of the environment in which AVs operate, the need to ensure passenger safety and comfort, in the presence of stochastic companion agents such as other vehicles, pedestrians, and moving objects. Achieving the full potential of AV technology will have profound implications for the field of control engineering. Furthermore, safety is a critical concern in AV technology (Samak et al., 2020). Ensuring that AVs can navigate and interact safely with their surroundings, obey traffic rules, and respond appropriately to unforeseen events is of utmost importance. This necessitates the development of robust and reliable control systems that can handle the complexities and uncertainties of real-world driving scenarios. The implications of successful AV technology implementation extend beyond engineering. They have the potential to revolutionize transportation, urban planning, and society as a whole (Xia et al., 2020).

1.2 Statement of the Problem

AVs are being developed by a variety of manufacturing companies and are close to being widely adopted in many applications. Ensuring that AVs can safely, accurately and smoothly follow a desired trajectory is crucial for their safe and efficient operation.

Although, there have been several studies dedicated to trajectory tracking of AVs, achieving precise and reliable tracking remains a significant challenge in the presence of uncertainties, especially in unpredictable or difficult road conditions. While conventional controllers have their own advantages, they experience a number of problems that reduce their efficiency in trajectory tracking for AVs. These problems include limited adaptability, difficulty in handling constraints, lack of predictive capability to anticipate future events and difficulty in handling complex trajectory profiles that involves complicated maneuvers. MPC is a promising trajectory tracking method for AVs, but improving tracking accuracy and addressing challenges in design and practical implementation are ongoing efforts. One key challenge is the dynamic nature of AVs and their varying models over time. The traditional fixed model approach may not sufficiently capture the evolving dynamics and uncertainties of AVs. Accurately tracking trajectories in dynamic environments requires considering the varying AV models and developing strategies that can dynamically adjust to these changes.

1.3 Objective

1.3.1 General Objective

The general objective of this thesis is to develop effective MPC-based trajectory tracking strategy for AVs to operate autonomously in a wide range of conditions.

1.3.2 Specific Objective

- ✓ Developing a kinematic and dynamic model of an AV considering constraints.
- ✓ Designing a MPC algorithm for trajectory tracking based on the dynamic model.
- ✓ Designing MPC controller optimization problem, that is solved online to compute the control inputs that guide the AV towards the desired trajectory.
- ✓ Analyzing and validating the performance of the MPC-based trajectory tracking strategy under different trajectories, and testing its performance in simulation.

1.4 Significance of the Study

AVs controlled by MPC can be used in a variety of applications, due to the rapid advancement of robotics and automation technologies. With the recent improvements in computer calculation capability, the complexity and CPU elapsed time of MPC with constraints are no longer major limitations. The proposed MPC controller design approach can be effectively used in research, development and practical applications for the ASTU Center of Transportation and Vehicle Engineering (CTVE) Center of Excellence.

1.5 Scope of the Thesis

The scope of this thesis includes developing a trajectory tracking controller for an AV with three DoF using MPC, which involves designing a controller for trajectory tracking that uses a nonlinear dynamic model, taking into account factors such as combined-slip tire forces, rolling resistance, and nonlinearities. The controller is designed to use the throttle and brake as longitudinal control input and steering angle as lateral control inputs, and the objective is to control the longitudinal motion of the vehicle and minimize the lateral distance between the vehicle and the predetermined path. Constraints are introduced in the design process to ensure that the controller meets desired performance requirements.

1.6 Motivation of the Study

Research into trajectory tracking for AVs using MPC has gained momentum due to the increasing demand for precise and safe operations for challenging road conditions. AVs must navigate through traffic, adapt to dynamic road conditions and avoid collisions with other vehicles and pedestrians for optimal safety. MPC-based trajectory tracking incorporates constraints such as vehicle dynamics, safety standards and traffic regulations to meet these requirements. Finally, MPC-based trajectory tracking for AVs research has the potential to promote the design of efficient and stable AV systems, which could have a significant impact on mobility, sustainability, and transportation.

1.7 Limitations of the Study

The thesis work is limited to designing the system model on a level road surface without considering the presence of inclined or uneven terrain. This simplifying assumption may not fully capture the challenges and dynamics introduced by driving on inclined surfaces. The research also focuses on handling trajectory uncertainties, although it might not cover the complete spectrum of uncertainty that AVs might experience in real-world scenarios, including things like bad weather, changing traffic patterns and sensor limits.

The research work primarily focused on the design and optimization of the control system in a simulated environment. While extensive evaluation and testing of AV systems can be performed using simulation, the practical implementation of these systems in hardware remains a challenge that has not been addressed.

1.8 Thesis Organization

This thesis is organized as follows:

- ✓ *Chapter 1:* The research work's history, statement of the problems, objectives, scope, limitations, and importance are all described.
- ✓ *Chapter 2:* The literature on AV and related research is reviewed in this chapter.
- ✓ *Chapter 3:* The kinematic and dynamic models of AV are described, as well as the materials and methods used.
- ✓ *Chapter 4:* The controller design is covered in this chapter.
- ✓ *Chapter 5:* The simulation result for proposed MPC controller is analyzed and discussed.
- ✓ *6:* Conclusions and recommendations are included in this chapter.

CHAPTER TWO

2 LITERATURE REVIEW

2.1 Overview

This chapter covers brief technical review, components and applications of AVs. The discussion of AVs' concept and architecture then becomes the main topic. In addition, the chapter reviews literatures related to research works in the subject of AV and discusses their benefits and gaps.

2.2 Brief Technical Review of AVs

With the help of modern technology, AVs are capable of sensing, localizing, perceiving, choosing and controlling their own movements. The specific tasks of AVs include steering and accelerating without human intervention. Additionally, AVs must recognize pedestrians, asses road conditions and traverse through traffic patterns. The ultimate goal of AV is to promote road safety, eliminate human error-related accidents, increase transportation efficiency, and provide options for people who are unable to drive. Many companies and researchers worldwide are working to develop and test AV technology in order for it to become totally autonomous (Liu et al., 2020).

AVs rely on a combination of key components that work together to enable autonomous driving. These components include

1. ***Sensors***: AVs use a range of sensors, including radar, LiDAR, sonar, GPS, odometry, and IMUs, to understand their environment. These sensors provide the system information about its surroundings, including where other objects are located, the topography, and pertinent signs.

2. ***Computer Vision***: AVs process the data from their sensors using computer vision algorithms to extract relevant information about the environment. This includes identifying objects, recognizing patterns and detecting features such as road lanes and traffic lights.

3. ***Sensor Fusion***: AVs combine the data from multiple sensors to create a more comprehensive understanding of their surroundings. It involves combining the data from different sensors in a way that allows the system to make more accurate decisions about its environment.

4. ***Localization**:* AVs use localization algorithms to determine their position and orientation within their environment. This is typically done using a combination of sensors and map data, and can involve techniques such as simultaneous localization and mapping (SLAM).
5. ***Path Planning***: AVs use path planning algorithms to determine the best route to follow based on their current position destination and any constraints or objectives that they need to consider.
6. ***Control**:* AVs use advanced control algorithms to determine the appropriate control inputs (such as throttle, brake, and steering) based on the vehicle's current state and the desired trajectory.

These components work together to enable AVs to navigate safely and efficiently through their environment, and to make decisions about how to respond to changing conditions and obstacles (H. Wang et al., 2019).

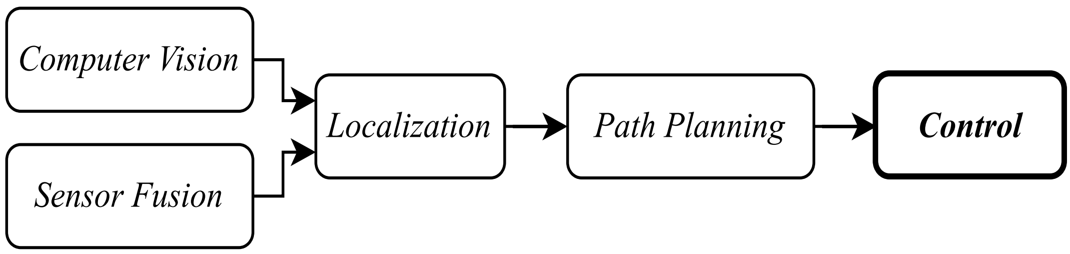

Figure 2.1: Architecture of AVs

The automatic control is the last piece in the sequence of the AV and one of the most important tasks since it is in charge of guaranteeing its motion.

The automotive industry uses the Society of Automotive Engineering’s autonomy scale to determine various levels of autonomous capacity. This scale serves as a valuable tool for understanding and categorizing the varying degrees of autonomy in vehicles. By utilizing this scale, we can gauge our current position in relation to the rapid advancements in autonomous technology (Ranjan & Senthamilarasu, 2020).

- ✓ ***Level 0 (No Automation):*** Both the steering and the acceleleration of the car are controlled by the driver. This level of autonomy may include issuing a warning to the driver, but the vehicle itself may not take any action.
- ✓ ***Level 1 (Driver Assistance):*** The driver assumes primary control over the car and is responsible for managing most of its features. This includes keeping an eye the environment as well as managing steering, breaking and accelerating.

- ✓ ***Level 2 (Partial Automation):*** The vehicle is capable of assuming control over certain aspects such as steering or acceleration, aiming to relieve drivers of a few basic tasks. However, the presence of the driver in the car is still necessary.
- ✓ ***Level 3 (Conditional Automation):*** The entire environmental monitoring is done by the vehicle. It can function in autonomous mode in some circumstances, but the driver must be ready to take over if the vehicle is put in conditions that are beyond its scope of operation.
- ✓ ***Level 4 (High Automation):*** The acceleration, braking, and steering are all under the vehicle's control. It can also keep an eye on the car, the pedestrians, and the general state of the road. The car can operate in autonomous mode for a large amount of the time, but in circumstances that are out of the system's control, such as congested urban areas and busy streets, a human driver must still take over.
- ✓ ***Level 5 (Complete Automation):*** The vehicle will be completely autonomous. No human driver is necessary as the vehicle handles all critical tasks including steering, braking, and pedal control. It actively monitors the environment, recognizing and responding to various driving conditions, including traffic congestion.

2.2.1 Applications of AVs

With the advancements in advanced control techniques, ML, and AI, AVs have the potential to revolutionize various sectors. Here are some examples:

- ✓ *Robotaxis:* AVs can be used as robotaxis, providing transportation services without the need for human drivers. It revolutionizes the taxi industry and enhance urban mobility.
- ✓ *Shuttles:* AVs can be used as autonomous shuttle services in airports, campuses, and urban areas, providing convenient and efficient transportation options.
- ✓ *Delivery Vehicles:* AVs can be used for autonomous delivery services, enabling the efficient and cost-effective transportation of goods and packages.
- ✓ *Trucking:* Autonomous trucks have the potential to transform the freight industry by improving safety and optimizing long-haul transportation operations.
- ✓ *Logistics:* AVs can be utilized in logistics, including warehouses storage and shipping centers, for tasks such as material handling, inventory management, and autonomous forklift operations.
- ✓ *Heavy Machinery:* AV technology can be applied to heavy machinery used in agriculture, mining and construction by improving productivity and safety.

Adoption of AVs in these industries has the potential to have a wide range of benefits, such as higher productivity, lower labor costs, improved safety, and enhanced environmental sustainability. However, it is crucial to address challenges related to regulations, infrastructure, public acceptance, and ensuring the reliability and robustness of AV technologies for widespread deployment in these sectors.

2.3 Recent Related Work on AVs

A review of earlier AV investigations is provided in this section. Numerous research publications have provided insight that has been used throughout the early stages of the design of AVs. In order to obtain optimal performance while keeping computational feasibility, these research works have investigated practical concerns for the real-world manufacture of AVs. In Table 2.1 a summary of the key research papers used in this thesis is presented.

In (Alcalá et al., 2020), a novel control approach for autonomous racing vehicles utilizing LPV theory and LPV-MPC is introduced in this paper. The practical implications of this research indicate that the proposed LPV-MPC strategy can be employed as a promising solution for autonomous driving control challenges in real-world scenarios and in real-time. The study incorporates an optimal off-line trajectory planner to compute the best trajectory considering the constraints of the racing circuit. Additionally, an identification of the system model based on optimization techniques is performed. The effectiveness of the proposed strategies is validated through simulations and experimental tests conducted on a physical platform. The controller is capable of generating a 20-step prediction at a frequency of 33Hz and successfully following the desired trajectory, albeit with some error attributed to unmodeled dynamics.

In (Alcalá et al., 2019), an innovative approach for trajectory tracking of AVs using a cascade control system is proposed. The proposed approach uses an external loop that employs an LPV-MPC technique for position control, while the internal loop uses an LPV-LQR technique for dynamic control. Both methods make use of an LPV representation of the kinematic and dynamic models of the vehicle. When compared to the non-linear variant, the LPV-MPC technique can produce extremely accurate results with reduced computational cost, allowing for real-time operation. It is also shown how well the LPV-LMI-LQR approach works to regulate the vehicle's dynamics. The findings show that the suggested strategy has the potential to improve both performance and safety.

In (Vu et al., 2021), a NMPC strategy for AVs that can monitor different trajectories is introduced. By taking into account both hard and soft constraints within the controller, the objective is to improve the stability and control abilities of AVs. While taking into consideration the dynamic constraints imposed by the vehicle's physical limitations, the environment, and surrounding obstacles, the NMPC controller uses an online objective function to compute the best control actions for vehicle speed and steering angular velocity. The possibility of identifying the best control actions and preserving system stability increases with the introduction of softer limitations. To assess the suggested controller's performance and efficacy, many parameters are used in thorough simulations and analyses.

In (Alcala et al., 2018), a control strategy for autonomous vehicle guidance is proposed, employing a non-linear kinematic Lyapunov-based technique with LQR-LMI tuning. The closed-loop system is reformulated in a LPV form to optimally adjust the parameters of the Lyapunov controller. An optimization algorithm is utilized to solve the LQR-LMI problem and determine the controller parameters. Additionally, a pole placement constraint is incorporated to ensure that the dynamic loop achieves the maximum achievable performance of the kinematic loop. The results show that the suggested controller successfully and comfortably steers the vehicle from a beginning point to a finish position. The control strategy successfully addresses both the lateral and longitudinal control problems of AVs. The optimization algorithm used for parameter determination is efficient and provides optimal results, while the inclusion of the pole placement constraint enhances system performance.

In (Pang et al., 2022), a LTV-MPC design for AVs based on a two DoF kinematic vehicle model is presented. The primary contributions of this study are the reduction in computational complexity, improved control accuracy, and real-time performance compared to existing MPC methods, thereby facilitating the implementation and development of similar MPC controllers. The proposed LTV-MPC controller ensures AVs in real driving situations high-precision trajectory tracking, hence boosting their dependability and safety. The findings show that the suggested LTV-MPC technique monitors the desired reference road trajectories for AVs with outstanding accuracy and stability across a variety of driving circumstances.

In (Jeong & Yim, 2021), a four-wheel independent steering and driving algorithm based on MPC for AVs is proposed. The reference state decision and MPC-based vehicle motion controller modules make up the algorithm. It efficiently distributes control effort among a number of actuators, improving reference tracking performance as well as the efficiency and safety of autonomous driving. By more effectively controlling the body slip angle, enhancing path and velocity tracking, and actively allocating driving and braking torques to specific actuators, simulation results show that the proposed approach outperforms base algorithms. The algorithm is simple modifiable when the actuator design changes and is responsive to various road conditions and vehicle systems. Overall, the suggested method appears to have potential for improving the efficiency and security of AVs on the road.

In (M. Wang et al., 2022), the solution for path tracking for autonomous cars proposed in the paper combines MPC and GA. The vehicle's future movement is predicted using a nonlinear predictive model, and an objective function is created based on the predicted movement and the intended path. Instead of employing a convex quadratic programming solver, GA is used to tackle the optimization problem, allowing for greater flexibility and the ability to solve any nonlinear problem. Simulations and field testing are used to validate the proposed MPC-GA method, demonstrating its effectiveness in comparison to more conventional approaches.

In (Tang et al., 2020), in order to increase tracking accuracy and mitigate road curvature disturbances, this paper proposes an alternative path tracking control system for AVs that combines kinematic MPC with PID feedback control of yaw rate and a vehicle sideslip angle compensation. This technique enhances tracking precision and manages road curvature disturbances. Although the proposed solution requires for a vehicle sideslip angle sensor, which may not be present in all AVs, it considerably increases tracking accuracy and manages road curvature disturbances. Even at high speeds and the handling limits, the proposed method is effective and robust, as shown by the modeling and field trial results.

In (Y. Wang et al., 2021), the paper proposes a reliable trajectory tracking control strategy for AVs that accounts for the nonlinearity, uncertainty, and fluctuating longitudinal velocity of the vehicle. In order to improve vehicle handling stability, the proposed system contains a novel target longitudinal velocity planning, a lateral control objective, a reliable active disturbance rejection control route tracking controller, and a feedforward-feedback control method. In order to effectively deal with fluctuating velocity and unpredictable lateral disturbance, the article created a reliable active disturbance rejection control route tracking controller and an extended state observer. It also employed a feedforward-feedback control method to regulate the overall tire torque. Through simulations of single-lane change maneuver and double-lane change maneuver, the suggested controller's robustness under velocity-varying conditions and rapid lateral disruption was also assessed.

In (Faulwasser & Findeisen, 2011) a NMPC approach for trajectory tracking of constrained continuous-time systems with a known and asymptotically constant reference trajectory. The proposed approach differs from standard methods by explicitly considering end penalties and terminal regions that depend on explicit times. The authors introduce the concept of time-varying level sets of Lyapunov functions as terminal regions, which can be efficiently computed using a quadratic Lyapunov function. These terminal regions handle the time-varying nature of the tracking problem and ensure positive invariance. The proposed approach also accounts for input and state constraints and guarantees recursive feasibility. In contrast, standard approaches may not explicitly consider end penalties and terminal regions that depend on explicit times, and may not guarantee positive invariance or recursive feasibility. The proposed NMPC scheme explicitly addresses input and state constraints while ensuring recursive feasibility.

In (Lin et al., 2019) proposed an adaptive MPC-based route tracking method for AV that enhances tracking stability and accuracy in comparison to general MPC. The recursive least squares approach is first utilized by the proposed controller to online estimate the road friction coefficient and tire cornering stiffness. The vehicle dynamics model is then updated using the predicted tire cornering stiffness, and the road adhesion constraint is updated using the estimated road friction coefficient. The main contribution of this research lies in the proposed Adaptive MPC path tracking controller, which not only ensures high accuracy and stability in path tracking but also exhibits excellent adaptability to changing working conditions, particularly variations in speed and road conditions. This is achieved through the incorporation of estimators for tire cornering stiffness and road friction coefficient, as well as a control parameter selection module.

In (Hu, Wang, et al., 2019) the ISMC was used to examine the path-tracking control problem for AVs while taking temporary performance improvements into account. System uncertainties, state immeasurability's, and steering angle constraint are considered, when designing a reliable path-tracking controller. On the basis of the MME-EKF observer, a novel, all-encompassing ISMC technique is created, which is utilized to estimate errors caused by unmodeled dynamics and system uncertainties. With the measurement of the yaw rate and roll rate, the lateral velocity and roll angle are estimated. A robust EKF is created to achieve yaw stabilization and improve the transient tracking performance while taking input saturation of the vehicle states into consideration. This increased estimation accuracy for the vehicle states in the presence of parametric uncertainties caused by the lateral and roll dynamics.

Table 2.1: Summary of literature review

Title, Authors & year published	Controller Used	Achievement	Gap
Autonomous racing using LPV-MPC (Alcalá et al., 2020)	LPV-MPC	Proposed LPV-MPC strategy can be used as a novel approach to solve autonomous driving control problems under realistic conditions in real-time.	Didn't consider inertial frame coordinates resulting in less accurate AV model and less precise control
LPV-MPC control for autonomous vehicles (Alcalá et al., 2019)	LPV-MPC NLMP	LPV-MPC provides optimal response it isn't linearized and the technique works very well compared to the NMPC problem but in a much faster way (more than 50 times faster).	Only kinematic model is used for simulation, which results in limited accuracy, Lack of stability and inability to handle complex maneuvers.
MPC for Autonomous Driving Vehicles (Vu et al., 2021)	NMPC	The capacity of AVs to maintain system stability and identify the best control actions is improved by the NMPC controller being subject to hard and softer constraints.	Slower when compared to other MPC. And the hard constraints have more difficulty driving the vehicle tracking to the trajectory
Path tracking Method based on MPC and GA algorithm for AV (M. Wang et al., 2022)	NMPC	The NMPC is used to predict the future movement of the AV, GA is applied to solve the optimization problem instead of using a convex quadratic programming solver.	Computationally very heavy and complex It is sensitive to model uncertainty.
Path tracking of AV based on adaptive MPC (Lin et al., 2019)	Adaptive MPC	Proposed Adaptive MPC path tracking controller ensures good path tracking accuracy and stability and also has a great adaptive ability to changing working conditions, especially to speed and road conditions.	Tracking accuracy is low at high speed, due absence complex dynamic model. Tuning of the Adaptive parameters is online and very challenging

CHAPTER THREE

3 METHODOLOGY AND AV SYSTEM MODELING

3.1 Overview

This chapter discusses the materials and procedures employed in the research project. It involves the development of a mathematical model for the AV system, incorporating both complex dynamic modeling methods and kinematic modeling approaches. The chapter also includes an analysis of the AV's open-loop response, offering insights into its behavior without the presence of a controller.

3.2 Materials

The system model in the thesis is simulated and tested using the MATLAB R2022a software package. MATLAB is a versatile computing tool with various toolboxes that facilitate numerical solutions and graphical user interface design. Microsoft Office v.2021 is utilized for editing the thesis documentation, while diagrams and flow charts are created using the draw.io website.

3.3 Method

In order to attain the objectives of the thesis, the following methods and techniques are followed.

- ✓ Comprehensive review of existing literature and research paper related to trajectory tracking of AVs was conducted.
- ✓ Mathematical model (kinematics and complex dynamics) of the AV was developed, considering constraints, uncertainties and combined slip tire forces.
- ✓ The AV system was analyzed to understand its behavior and characteristics.
- ✓ MPC controller is designed for the AV to achieve it's the desired objectives.
- ✓ The performance of the controller is simulated and tested using MATLAB under different scenarios.
- ✓ The designed controller is optimized and tuned, then after evaluated and analyzed by comparing to different controller.

The block diagram in Figure 3.1 summarizes the method followed throughout the research work.

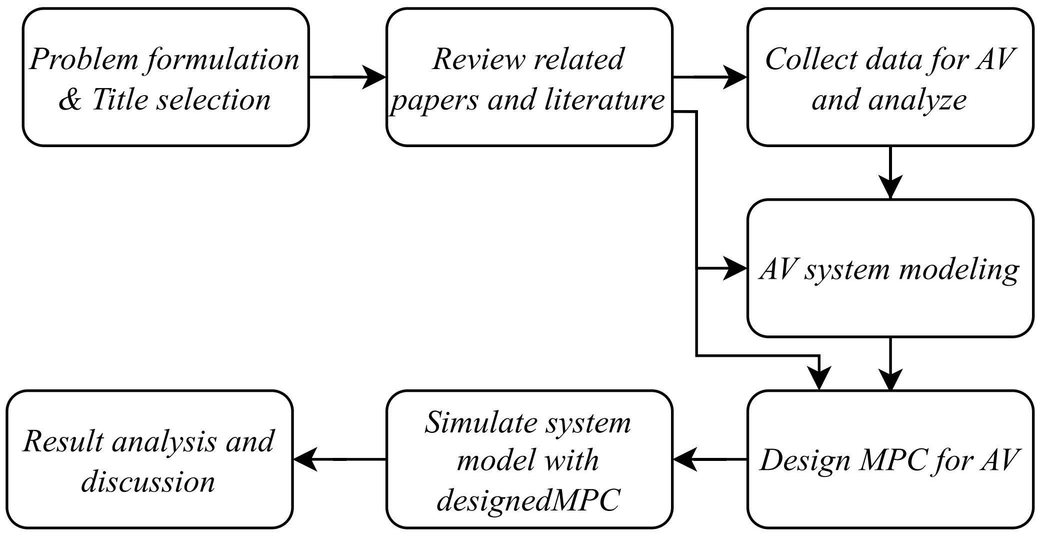

Figure 3.1: Flowchart of MPC Design Methodology for AVs

3.4 Mathematical Modeling of AV and Description

System modeling of an AV's platform in this thesis work includes kinematic and dynamic modeling. Kinematic modeling for a system focuses on the geometric relationships and physics of motion without taking influencing forces into account. Dynamic modeling, on the other hand, is the study of motion that incorporates and represents many factors, such as energy, friction and aerodynamic forces. A derivation of an effective system model makes a system more understandable and controllable.

A control system with inputs, outputs, and state variables are used to model the AV system. The steering angle and acceleration command are inputs to the system, and the position, velocity, orientation, and angular velocity of the vehicle are all state variables. A group of nonlinear differential equations that link the inputs to the state variables and outputs explain the dynamic behavior of the system.

The thesis study focuses on one specific vehicle architecture, the front-wheel steering paradigm for AVs. This type of vehicle has four wheels, with the front and rear wheels lumped so that they resemble a bicycle. In this design, the front wheel acts as the primary steering and navigational mechanism in this design.

3.4.1 Representing AV's Coordinate

The position of the AV is primarily represented by the coordinate systems. For the development of system models, two coordinate systems Cartesian and Polar coordinates are frequently utilized. In this thesis work, system modeling is carried out in Cartesian coordinates due to their simplicity and ease of representation. To model and operate AV, the following cartesian coordinate systems are employed:

- ✓ ***The inertial reference frame (X, Y):*** Refers to a reference frame that is fixed in space in the coordinate system in the plane of the AV. It defines a random inertial base on the plane as the global reference frame from some origin.
- ✓ ***Body frame (x, y):*** Refers to a frame of reference that is attached to the AV itself, and moves along with the vehicle. It is also known as the vehicle-attached or local frame. The body frame is typically defined such that the *x*-axis points forward along the longitudinal axis of the vehicle and the *y*-axis points to the right along the lateral axis of the vehicle.

Kinematic modeling and dynamic modeling are the two main methods for modeling a vehicle's motion. Kinematic modeling accurately represents the motion of the vehicle at slow speeds and with little acceleration. It, however, might not completely capture the complexity and dynamism of real-world environments. Dynamic modeling, on the other hand, considers all of the forces and moments acting on the vehicle. It gives a better understanding of how the vehicle will behave under varied driving circumstances, such faster speeds and difficult maneuvers. Considering all of the forces and moments that affect the vehicle, dynamic modeling can offer a more complete and accurate representation of the vehicle's behavior in real-world scenarios. Dynamic modeling is therefore more suited for effectively simulating AVs (Reda et al., 2020).

3.4.2 Kinematic Bicycle Model of the AV Motion

In this section, a kinematic model of AV motion is introduced. This model is frequently used to guide at low speeds or with little lateral acceleration. Because it is straightforward and can accurately depict nonholonomic restrictions. The bicycle model has been employed to illustrate the kinematics of the vehicle based on (Rajamani, 2012). The kinematic bicycle model is a condensed version of the kinematic four-wheel model, in which the two rear wheels and the two front wheels are each lumped into one (imaginary) rear wheel and front wheel. The kinematic model isolates the motion of the vehicle from the forces acting on it (Boyali et al., 2018).

Since the front wheel's orientation is determined by the direction the vehicle is traveling, the kinematic bicycle model is also known as the front wheel steering model. This model is particularly useful for vehicles with adjustable front wheel angles in relation to the vehicle's direction. The front wheel of the suggested bicycle model reflects the front right and left wheels of the vehicle, while the rear wheel represents the rear right and left wheels. The center of rotation for the vehicle, or CG, is where the ICR is situated. As shown in Figure 3.2, it is the point around which the vehicle turns when doing a turning maneuver.

The kinematic bicycle model is analyzed using the vehicle coordinates in the body frame represented by x_c and y_c, positioned at the CG of the vehicle. The velocity vector of the vehicle is represented by v and aligns with the direction of the front wheel.

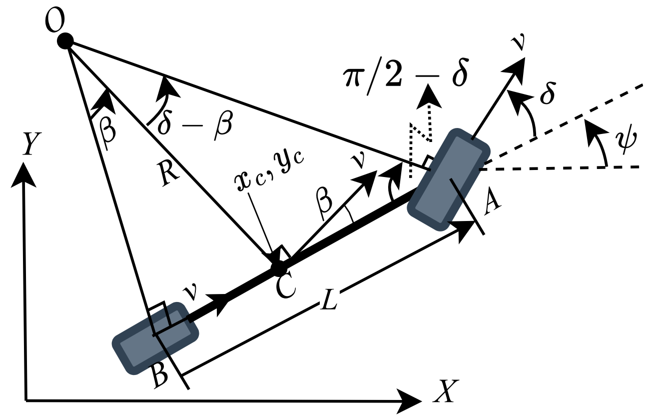

Figure 3.2: Kinematic Bicycle model

The motion at the CG of the vehicle deviates slightly from both the bicycle's heading and the direction in which either wheel is moving forward. This deviation is known as the slip angle (β). The steering angle for the front wheel is denoted by δ and is measured relative to the longitudinal direction of the bicycle (Rajamani, 2012). The vehicles heading angle or yaw angle is represented by ψ. The distance from the center of the rear wheels to the CG is denoted as l_r, and the distance from the CG to the center of the front wheel is denoted as l_f. The total wheelbase of the vehicle is given by $L = l_f + l_r$.

In this inertial frame model, the variable ψ represents the yaw angle, which indicates the orientation of the vehicle with respect to the reference frame. Yaw rate ($\dot{\psi}$) represents the rate at which the vehicle's orientation is changing. The direction which the vehicle is traveling or course angle for the vehicle is defined as:

$$\gamma = \psi + \beta \tag{3.1}$$

By applying sine rule to triangle OCA:

$$\frac{\sin(\delta-\beta)}{l_f}=\frac{\sin\left(\frac{\pi}{2}-\delta\right)}{R} \tag{3.2}$$

Also, by applying sine rule to triangle OCB:

$$\frac{sin(\beta)}{l_r}=\frac{sin\left(\frac{\pi}{2}\right)}{R}=\frac{1}{R} \tag{3.3}$$

Whereas R represents the radius of the vehicle's path, it is defined as the distance from the ICR to the CG of the vehicle.

By expanding equation (3.2), and multiplying both sides by ${l_f}/{cos(\delta)}$, the following can be obtained:

$$tan(\delta)\cos(\beta)-\sin(\beta)=\frac{l_f}{R} \tag{3.4}$$

And by multiplying both sides of equation (3.3) by l_r and adding to equation (3.4) gives:

$$tan(\delta)\cos(\beta)=\frac{l_f+l_r}{R} \tag{3.5}$$

Under the assumption that the radius of the vehicle's path changes gradually, primarily due to the vehicle's low speed, the rate at which the vehicle's orientation changes can be considered equivalent to its angular velocity (Rajamani, 2012). This assumption indicates that the vehicle's rotational motion is primarily controlled by its angular velocity, which is directly related to the rate at which its orientation changes.

$$\dot{\psi}=\frac{v}{R} \tag{3.6}$$

By using the equations (3.6) in the equation (3.5), the yaw rate can be obtained as:

$$\dot{\psi}=\frac{vcos(\beta)}{l_f+l_r}\ tan(\delta) \tag{3.7}$$

From Figure 3.2, the rate of inertial frame coordinates can be obtained as:

$$\dot{X}=vcos(\psi+\beta) \tag{3.8}$$

$$\dot{Y}=vsin(\psi+\beta) \tag{3.9}$$

Hence, the equations of motion for the vehicle in the kinematic model can be described by equations (3.7), (3.8) and (3.9). Where the slip angle β can be obtained as:

$$\beta=tan^{-1}\left(\frac{l_r tan\delta}{L}\right) \tag{3.10}$$

3.4.3 Dynamic Bicycle Model of the AV Motion

The dynamic vehicle model is presented in this section and is based on the bicycle model technique shown in Figure 3.3. The longitudinal direction, which corresponds to the heading direction, and the lateral direction, which is perpendicular to the heading, are the two components of motion that are taken into account by the model (Schramm et al., 2014).

In the dynamic bicycle model, the following assumptions are made.

- ✓ The left and right wheels of both the front and rear axles are lumped into a single wheel, simplifying the model to a two-wheel bicycle representation.
- ✓ Nonlinear effects such as suspension movement, road inclination, and aerodynamic influences are neglected.

The full dynamic bicycle model encompasses three DoF. These include the vehicle's longitudinal velocity (V_x), lateral velocity (V_y), and yaw angle (ψ) motion. The lateral position of the vehicle is measured along the vehicle's lateral axis, specifically to the point that serves as the vehicle's center of rotation. The yaw angle is measured relative to the inertial X axis (Zhu et al., 2023).

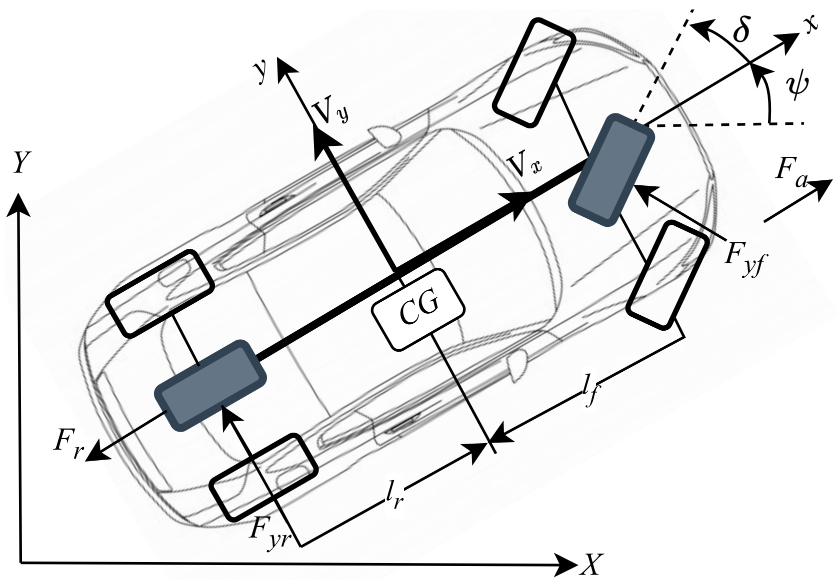

Figure 3.3: Dynamic Bicycle model

By applying Newton's second law of motion and analyzing net acceleration in a rotating body frame,

$$\sum F_x = ma_x = F_a - F_r - F_{yf} sin(\delta) \quad (3.11)$$

$$\sum F_y = ma_y \quad (3.12)$$

$$\sum M_z = I_z\ddot{\psi} \tag{3.13}$$

$$F_a = ma \tag{3.14}$$

$$F_r = \mu mg \tag{3.15}$$

F_a is applied force in longitudinal direction provided by the vehicle engine (acceleration) or brakes (deceleration). F_a is positive when applying throttle and negative when applying brakes. The rolling resistance (friction force) is denoted by F_r. Where, the rolling resistance coefficient, μ, quantifies the magnitude of this resistance.

When analyzing Forces in Lateral Direction, we can obtain:

$$F_{yflat} = F_{yf}cos(\delta) \tag{3.16}$$

$$F_{yflong} = F_{yf}sin(\delta) \tag{3.17}$$

The distance from the vehicle CG to the front and rear tires are represented by l_f and l_r respectively.

By using equations from (3.11) to (3.15) the following equations of motion can be obtained.

$$ma_x = ma - \mu mg - F_{yf}sin(\delta) \tag{3.18}$$

$$ma_y = F_{y_r} + F_{y_f}cos(\delta) \tag{3.19}$$

$$I_z\ddot{\psi} = F_{yf}cos(\delta)l_f - F_{yr}l_r \tag{3.20}$$

Since the longitudinal acceleration $a_x = \dot{V}_x - \dot{\psi}V_y$ and the lateral acceleration at the center of mass of the vehicle is $a_y = \dot{V}_y + \dot{\psi}V_x$. The equations (3.18) and (3.19) can be further derived as:

$$ma - \mu mg - F_{yf}sin(\delta) = m\dot{V}_x - m\dot{\psi}V_y \tag{3.21}$$

$$\dot{V}_x = a - \mu g - \frac{F_{yf}sin(\delta)}{m} + \dot{\psi}V_y \tag{3.22}$$

$$F_{yr} + F_{yf}cos(\delta) = m\dot{V}_y + m\dot{\psi}V_x \tag{3.23}$$

$$\dot{V}_y = \frac{F_{yr}}{m} + \frac{F_{yf}}{m}cos(\delta) - \dot{\psi}V_x \tag{3.24}$$

From equation (3.20), the angular acceleration $\ddot{\psi}$ can be calculated as:

$$\ddot{\psi} = \frac{F_{yf}l_f}{I_z}cos(\delta) - \frac{F_{yr}l_r}{I_z} \tag{3.25}$$

3.4.4 Tire Modelling

The tires are among the most crucial elements in modeling vehicle dynamics. Since tires are the only part in contact with the road surface during normal operation, modeling tire forces is the essential part of a vehicle motion model. Tire forces are typically difficult to predict precisely and tires have nonlinear and empirically derived characteristics. Under normal driving conditions, it is possible to use a simple linear approximation to estimate the tire forces based on the research by (Kissai et al., 2017). However, this approximation is accurate only when the slip angles are small and the tire forces exhibit a linear relationship with the slip angles. The slip angles refer to the angles between the velocity vectors and the direction of the tire's orientation vector for both the front and rear wheels, denoted as α_f and α_r respectively. The tire forces are modeled and represented in the respective tire fixed coordinate frames, following the convention shown in Figure 3.4.

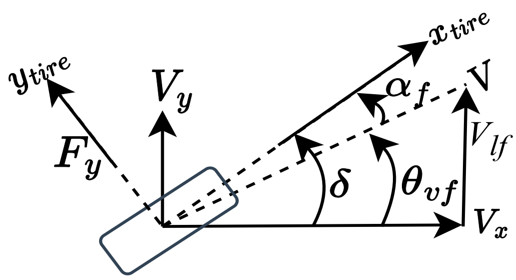

Figure 3.4: Tire frame coordinates

At higher speeds, the velocity direction of the front wheel and the orientation of the front wheel itself deviate due to tire deformation. The angle between the inertial X-axis and the velocity vector of the front wheel is denoted as θ_{vf}. When a vehicle is turning, the orientation of the wheel and the velocity of the tire contact patch result in an angle called the tire slip angle, represented as α. This angle plays a significant role in modeling the lateral force generated by the tire. In the single-track model, the tire slip angle α_f and α_r for the front and rear tires are defined as:

$$\alpha_f = \delta - \theta_{vf} \tag{3.26}$$

$$\alpha_r = \delta_r - \theta_{v_r} \tag{3.27}$$

The lateral velocity of the vehicle and the yaw rate of the vehicle multiplied by the distance from the CG to the front axle determine the velocity of the front left tire in the vehicle's body frame.

$$V_{lf} = V_y + \dot{\psi} l_f \tag{3.28}$$

$$tan(\theta_{vf}) = \frac{V_y + \dot{\psi} l_f}{V_x} \tag{3.29}$$

The control input δ is constrained between $-\pi/_6$ and $\pi/_6$, to ensure that the steering angle of the vehicle remains within a reasonable and feasible range during control actions. So, it is assumed that the θ_{v_f} will be small. Then, $tan\left(\theta_{v_f}\right) \approx \theta_{v_f}$, due to small angle approximation. Thus, equation (3.29) becomes:

$$\theta_{vf} = \frac{V_y}{V_x} + \frac{\dot{\psi} l_f}{V_x} \tag{3.30}$$

Similarly, the longitudinal velocity at the rear of the vehicle with respect to the body frame relates the velocity at the rear of the vehicle to the lateral velocity at the center of mass of the vehicle.

$$V_{lr} = V_y - \dot{\psi} l_r \tag{3.31}$$

$$tan(\theta_{vr}) = \frac{V_y - \dot{\psi} l_r}{V_x} \tag{3.32}$$

The angle θ_{v_r} will be very small which results in, $tan(\theta_{vr}) \approx \theta_{v_r}$, Then equation (3.32) becomes,

$$\theta_{vr} = \frac{V_y}{V_x} - \frac{\dot{\psi} l_r}{V_x} \tag{3.33}$$

The lateral force exerting on the vehicle at the tire contact patch of the front and rear tires are calculated as:

$$F_{yf} = C_{\alpha f} \alpha_f \tag{3.34}$$

$$F_{yr} = C_{\alpha r} \alpha_r \tag{3.35}$$

$C_{\alpha f}$ and $C_{\alpha r}$ represent the cornering stiffness coefficients of the front and rear wheels, respectively. The cornering stiffness of a tire refers to its ability to resist deformation when the vehicle undergoes rotational or turning maneuvers.

Since the bicycle dynamic model is being considered, the rear wheels are assumed to be fixed and not steerable. i.e., $\delta_r = 0$, thus, equation (3.27) is changed to $\alpha_r = -\theta_{vr}$. And the lateral forces exerting on the vehicle defined in equation (3.34) and (3.35) are defined as:

$$F_{yf} = C_{\alpha f} \left(\delta - \frac{V_y}{V_x} - \frac{\dot{\psi} l_f}{V_x} \right) \tag{3.36}$$

$$F_{yr} = C_{\alpha r}\left(-\frac{V_y}{V_x} + \frac{\dot{\psi} l_r}{V_x}\right) \tag{3.37}$$

3.5 State Space Representation of the AV

As in the previous section, the motion of the vehicle is assumed planar; the vehicle is assumed to be a rigid body with mass m and moment of inertia I_z with respect to the axis normal to the horizontal plane at the center of mass. The left and right tires are lumped into one, representing the whole axle. This approximation allows for a more manageable model and disregards certain complexities associated with the individual behavior of each tire. By doing so, the roll dynamics of the suspension and lateral weight transfer are neglected, and a large turning radius is assumed, such that the left and right tires undergo approximately the same lateral deflection as the virtual tire in the middle. This assumption facilitates the mathematical treatment of the vehicle's dynamics and allows for a more straightforward analysis. The coordinate frame and angular variables used in the dynamic bicycle model are shown in Figure 3.3.

The bicycle dynamic model considers both the inertial frame and the body frame to accurately describe the motion of the vehicle. By connecting these two frames, the absolute position and orientation of the vehicle can be captured.

From Figure 3.5 the relation between the inertial frame and the body frame is obtained as:

$$\dot{X} = V_x cos(\psi) - V_y sin(\psi) \tag{3.38}$$

$$\dot{Y} = V_x sin(\psi) - V_y cos(\psi) \tag{3.39}$$

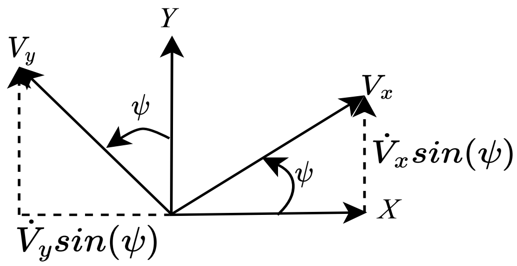

Figure 3.5: Relations between inertial frame and body frame of AV

The state space has six states which are $\begin{bmatrix} V_x & V_y & \psi & \dot{\psi} & X & Y \end{bmatrix}^T$ and two control inputs which are $[\delta \quad a]^T$. When inserting the value of F_{yf} and F_{yr} from equations (3.36) and (3.37) into $\dot{V}_x$, $\dot{V}_y$ and $\ddot{\psi}$, from equation (3.22), (3.24) and (3.25) the following rigid body constrained to the planar movement that has three degrees of freedom which its motion is governed by Newton-Euler equations can be formed, which describe the relationship between forces, moments, and motion. And by rearranging equations (3.38) and (3.39), the following equation of motion of the AV can be obtained.

$$\begin{aligned}
\dot{V_x} &= -\mu g + \frac{C_{\alpha f} sin(\delta)}{mV_x} V_y + \left(\frac{C_{\alpha f} sin(\delta) l_f}{mV_x} + V_y\right)\dot{\psi} - \frac{C_{\alpha f} sin(\delta)}{m}\delta + a \\
\dot{V_y} &= -\left(\frac{C_{\alpha r} + C_{\alpha f} cos(\delta)}{mV_x}\right) V_y + \left(\frac{C_{\alpha r} l_r - C_{\alpha f} cos(\delta) l_f}{mV_x} - V_x\right)\dot{\psi} + \frac{C_{\alpha f} cos(\delta)}{m}\delta \\
\ddot{\psi} &= \left(\frac{C_{\alpha r} l_r - C_{\alpha f} cos(\delta) l_f}{I_z V_x}\right) V_y - \left(\frac{C_{\alpha f} cos(\delta) l_f^2 + C_{\alpha r} l_r^2}{I_z V_x}\right)\dot{\psi} + \frac{C_{\alpha f} cos(\delta) l_f}{I_z}\delta \\
\dot{X} &= cos(\psi) V_x - sin(\psi) V_y \\
\dot{Y} &= sin(\psi) V_x + cos(\psi) V_y
\end{aligned} \tag{3.40}$$

where V_x, V_y are components of the velocity in the body-fixed coordinate frame and $\dot{\psi}$ is the yaw rate, the angular velocity of the rigid body with respect to the normal axis at the center of mass. Only the front wheel is assumed steered with δ denoting the steering angle.

To formulate the nonlinear state space equation using equation (3.40):

$$\dot{x} = f(x, u) \tag{3.41}$$

Where, $x = \begin{bmatrix} V_x & V_y & \psi & \dot{\psi} & X & Y \end{bmatrix}^T$, $\dot{x} = [\dot{x_1} \quad \dot{x_2} \quad \dot{x_3} \quad \dot{x_4} \quad \dot{x_5} \quad \dot{x_6}]^T$, $u = [\delta \quad a]^T$

$$\begin{cases}
\dot{x_1} = f_1(x, u) \\
\dot{x_2} = f_2(x, u) \\
\dot{x_3} = x_4 \\
\dot{x_4} = f_4(x, u) \\
\dot{x_5} = f_5(x, u) \\
\dot{x_6} = f_6(x, u)
\end{cases} \tag{3.42}$$

Where the value of $f(x, u)$ are:

$$\begin{aligned}
f_1(x, u) &= -\mu g + \frac{C_{\alpha f} sin(u_1)}{m x_1} x_2 + \left(\frac{C_{\alpha f} sin(u_1) l_f}{m x_1} + x_2\right) x_4 \\
&\quad - \frac{C_{\alpha f} cos(u_1)}{m} u_1 + u_2 \\
f_2(x, u) &= \left(-\frac{C_{\alpha r} + C_{\alpha f} cos(u_1)}{m x_1}\right) x_2 + \left(\frac{C_{\alpha r} l_r - C_{\alpha f} cos(u_1) l_f}{m x_1} - x_1\right) x_4 \\
&\quad + \frac{C_{\alpha f} cos(u_1)}{m} u_1
\end{aligned} \tag{3.43}$$

$$f_4(x,u) = \left(\frac{C_{\alpha r} l_r - C_{\alpha f} cos(u_1) l_f}{I_z x_1}\right) x_2 - \left(\frac{C_{\alpha f} cos(u_1) l_f^2 + C_{\alpha r} l_r^2}{I_z x_1}\right) x_4 + \frac{C_{\alpha f} cos(u_1) l_f}{I_z} u_1$$

$$f_5(x,u) = cos(x_3) x_1 - sin(x_3) x_2$$

$$f_6(x,u) = sin(x_3) x_1 + cos(x_3) x_2$$

The relationship between the body frame velocities and the inertial frame positions can be obtained as follows:

$$\begin{aligned} V_x &= cos(\psi)\dot{X} + sin(\psi)\dot{Y} \\ V_y &= -sin(\psi)\dot{X} + cos(\psi)\dot{Y} \end{aligned} \tag{3.44}$$

3.6 Open-Loop System Modeling Simulation Analysis

The open-loop simulation is a method of testing the derived kinematic and dynamic model of the AV to check if the model correctly captures the vehicle's behavior.

Table 3.1: AV's parameter specification

Parameters	**Description**	**Value**
m	Mass of the AV	1575 *Kg*
I_z	Yaw moment of inertia	2875 *Kg.m*^{2}
$C_{\alpha f}$	Front tire cornering stiffness	19000 *N/rad*
$C_{\alpha r}$	Rear tire cornering stiffness	33000 *N/rad*
l_f	Distance from the front axle to CG	1.2 *m*
l_r	Distance from the rear axle to CG	1.4 *m*
μ	Coefficient of rolling resistance	0.02
g	Gravity	*9.81 m/s*^{2}

Table 3.1 shows the AV's parameters specification used for this thesis and are based on the Chevrolet SUV car as in (Alcalá et al., 2019). A dynamic model of an AV in equation (3.41) is simulated based on parameters given in Table 3.1. It explains the consequence of input on the output state by using initial velocity of 1*m/s* for calculating input δ and a which are the steering wheel angle and the longitudinal acceleration.

The simulation result illustrated in Figure 3.6 reveals that the response of the states of the AV at initial speed of 1*m/s*. The lateral velocity V_y, which represents the vehicle's sideways movement, shows unpredictable behavior contrary to the expected value of 0 m/s. Ideally, the vehicle should not have any sideways motion.

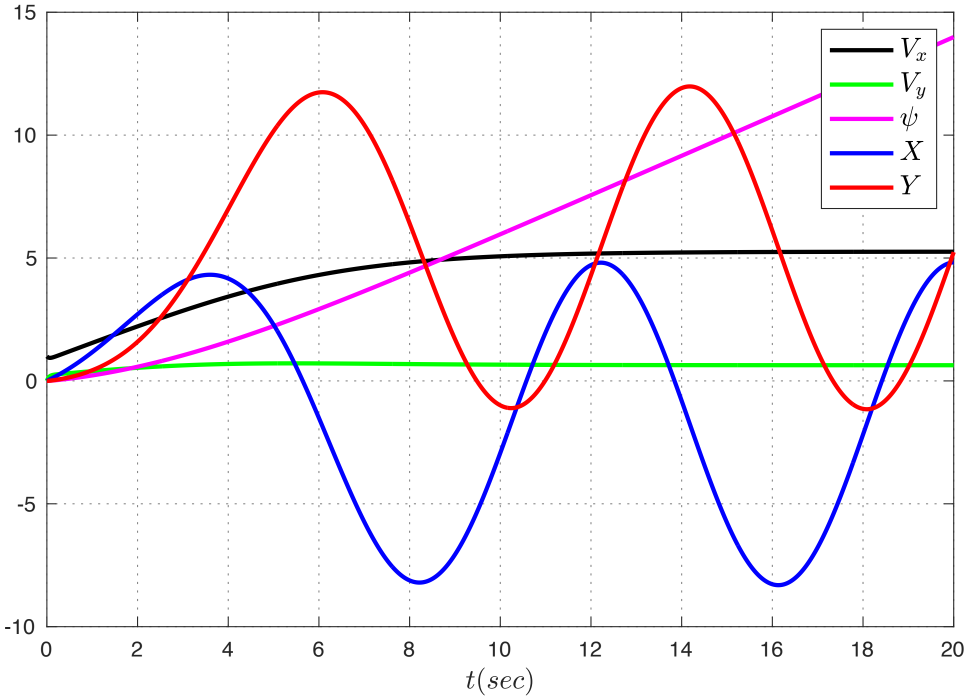

Figure 3.6: Open-loop response of AV

The yaw angle, suggesting instability in the vehicle's turning behavior and lead to compromised vehicle control and stability. It can result in unpredictable and erratic movements, making it difficult for the vehicle to accurately navigate and maintain its intended trajectory. The instability, nonlinearity, erratic and unpredictable behavior observed in Figure 3.6 of the X and Y positions of the inertial frame of the AV shows the need for a controller. A controller is required to track the trajectory of the AV's motion, ensuring accurate and stable positioning in the X and Y directions.

CHAPTER FOUR

4 CONTROLLER DESIGN

4.1 Overview

The main focus of this chapter is the design of the MPC controller, which is a key component of the technique suggested in this thesis and is essential for accomplishing the required control objectives. The theoretical underpinnings and fundamental ideas of AV trajectory tracking utilizing Linear MPC and LPV-MPC are covered in this chapter, providing an understanding of their underlying concepts. It also looks into the particular design criteria and development processes used in this thesis to create a successful MPC controller.

4.2 Control System Architecture for AVs

In this research work, MPC controller is designed for controlling the nonlinear model of AV. The objective of using MPC controller is to control the steering wheel angle and the longitudinal acceleration of the AV, in order to follow the given trajectory.

The entire control system of an AV is fundamentally broken down into the following two components:

- ✓ ***Longitudinal Control:*** This component controls the longitudinal motion of the AV, considering its longitudinal dynamics. In this case, the throttle (acceleration) and brake (deceleration) inputs to the AV control its longitudinal motion.
- ✓ ***Lateral Control*:** This component controls the lateral motion of the AV, considering its lateral dynamics. In this case the steering input, which controls the steering angle and the heading, is the input to AV.

4.3 Model Predictive Control

Since its origin in the late 1970s, Model Predictive Control (MPC) has seen significant development. The term MPC refers to a wide variety of control techniques that explicitly use a model of the process to derive the control signal by minimizing an objective function, rather than a specific control approach. MPC is an optimal control technique, that uses system dynamics to predict future behavior and optimizes these predictions over a finite time horizon. This flexible and model-based approach to control system design enables the optimization of control inputs across a future time horizon to meet desired objectives while considering system constraints. (Raković & Levine, 2019).

The MPC receives the current measurements of the system's past outputs at each time step. The current measurement serves as feedback to the MPC, allowing to compare the actual system outputs with the predicted outputs obtained from the AV system. The MPC evaluates tracking performance and pinpoints any deviations or errors in the system's behavior by comparing predicted and measured outputs. MPC operates in a receding horizon way by measuring or estimating the system's current state at each time step and computing a series of future control inputs based on model predictions. The process is repeated using the current measurements and predictions at the subsequent time step, with only the first control input in the sequence being executed (Brunton & Kutz, 2022).

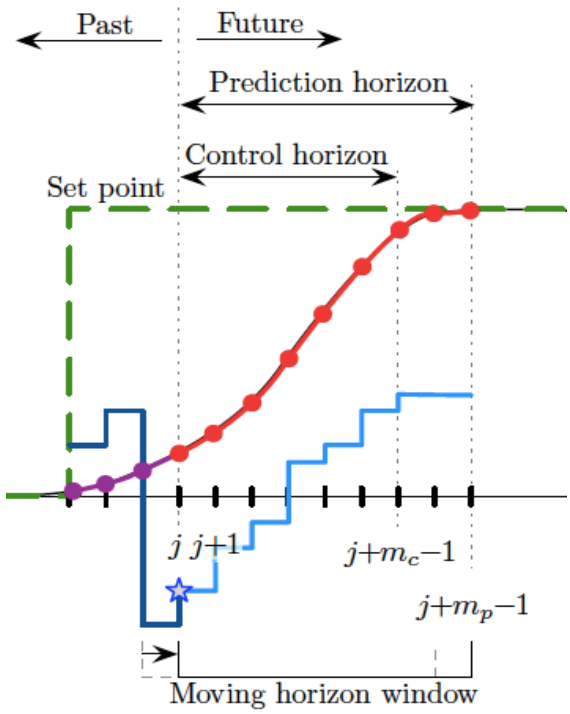

Figure 4.1: MPC Scheme *(Brunton & Kutz, 2022)*

MPC is frequently used interchangeably with the term "receding horizon control". Receding Horizon Control solves a fixed size optimization at each time step to determine the best control inputs to use from the current time to the end of the horizon based on the objective's constraints and the vehicle's current condition (Rossiter, 2018; M. Wang et al., 2022). In order to implement MPC, the basic structure shown in Figure 4.2 is used. Based on past, current and suggested optimal future control actions, the AV model is used to predict plant outputs in the future. The suggested optimal future control actions are obtained by solving an optimization problem using the AV model and considering the desired objectives and constraints. The optimizer determines these actions by calculating the cost function, which takes constraints and future tracking errors into consideration (Camacho & Bordons, 2007).

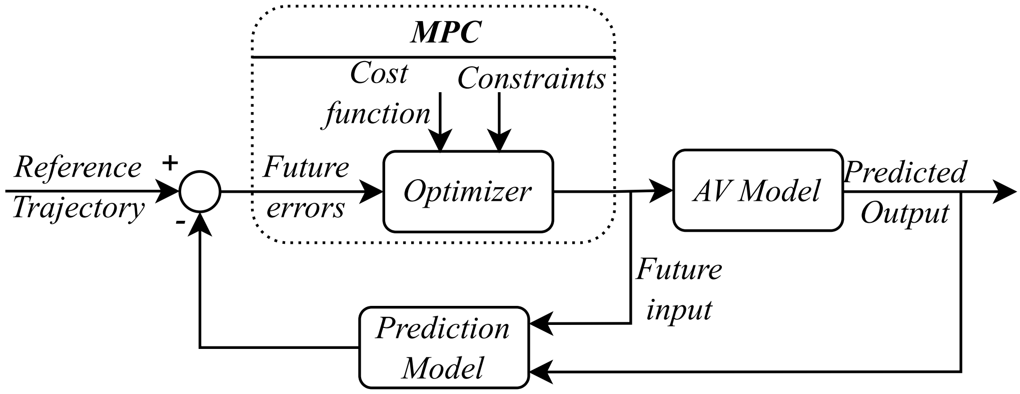

Figure 4.2: Block diagram of MPC AV model

Some of the key Advantages of MPC for AVs are:

- ✓ Ability to handle constraints.
- ✓ Ability to handle MIMO systems.
- ✓ Ability to predict future based on the system model.
- ✓ MPC is optimization technique, which means MPC is designed to find optimal control input by considering system dynamics, constraints and objectives.
- ✓ MPC is a totally open methodology, which allows for future extension.

Thus, MPC is a powerful control technique that is well-suited for trajectory tracking of AVs. A significant drawback of MPC is its computational intensity. The process of generating an optimal control action with MPC involves major computational requirements. The controller needs to solve an optimization problem repeatedly at each time step, considering constraints and system dynamics. The computational burden increases as a result of the need to analyze and optimize a cost function over a predicted horizon, which consists of multiple decision variables (Besselmann & Morari, 2009).

There are many types of MPC each subjected to specific system characteristics and control objectives. Some of them are Linear MPC, Nonlinear MPC, Robust MPC, Adaptive MPC, LPV-MPC and stochastic MPC. The controller used in this thesis work uses linearized MPC as a benchmark for performance comparison and LPV-MPC to account for time-varying AV system characteristics. By exploring these two approaches, the thesis aims to evaluate their respective performances in achieving the desired control objectives. The main focus of this thesis work is the LPV-MPC method which takes into account the system dynamics' dependence on varying parameters. In this approach, the controller design takes into account the time-varying nature of the system parameters and adjusts the control actions accordingly.

4.3.1 Linear MPC Controller Design for AV

The Linear MPC creates the optimal control actions by using a linearized model of the system. To make use of Linear MPC for trajectory tracking of an AV, the linearized continuous-time state space equation from equation (3.41) of the system needs to be discretized and augmented. To discretize the continuous-time state space equation, Euler's discretization method is used.

$$\begin{aligned} x(k+1) &= A_d x(k) + B_d u(k) \\ y(k) &= C_d x(k) \end{aligned} \tag{4.1}$$

Where, $A_d = I + A \cdot T_s$, $B_d = B \cdot T_s$ and $C_d = C$

In tracking problems, in order to maintain the tracking error zero or close to zero, the control input must be non-zero. This particular approach which eliminates steady-state error and make the system follow the desired trajectory is known as control signal increment. The control input is adjusted using the control signal increment in accordance with the difference between the desired trajectory and current system output. The MPC algorithm determines the increment in control signal required to bring the AV system to the desired trajectory. Therefore, in a tracking problem, the changes of input $\Delta u(k)$ are used, which are defined as follows:

$$\Delta u(k) = u(k) - u(k-1) \tag{4.2}$$

In order to capture the desired trajectory and tracking errors, additional variables or states are added to the state space. To give the MPC controller the ability to track the reference trajectory and account for deviations, the following augmented states are implemented (Han & Gao, 2022).

$$\begin{cases} \tilde{x}(k+1) = \tilde{A}\tilde{x}(k) + \tilde{B}\Delta u(k) \\ y(k) = \tilde{C}\tilde{x}(k) \end{cases} \tag{4.3}$$

$$\tilde{x}(k) = \begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix} = \begin{bmatrix} V_{x_k} & V_{y_k} & \psi_k & \dot{\psi}_k & X_k & Y_k & \delta_{k-1} & a_{k-1} \end{bmatrix}^T$$

$$\begin{cases} \tilde{A} = \begin{bmatrix} A_d & B_d \\ 0_{m\times n} & I_m \end{bmatrix} \\ \tilde{B} = \begin{bmatrix} B_d \\ I_m \end{bmatrix} \\ \tilde{C} = \begin{bmatrix} C_d & 0_{n\times m} \end{bmatrix} \end{cases} \tag{4.4}$$

Where, n is the dimension of state vector and m is the dimension of the control vector.

In tracking MPC, new states that capture the discrepancy between the system's actual states and the desired reference trajectory must be added when augmenting. The difference between the system's behavior and the reference trajectory is represented by these additional states, also known as tracking states. The controller can monitor the reference trajectory by reducing the error between the tracking states and the target trajectory by including these states in the MPC formulation. Each MPC iteration updates and controls the tracking states in order to steer the system toward the reference trajectory while satisfying constraints.

By defining the prediction horizon as N_p and the control horizon as N_c, the future state variables are calculated sequentially using the set of future control parameters as in (Liuping, 2009):

$$\begin{aligned} x(k+1) &= \tilde{A}x(k) + \tilde{B}\Delta u(k) \\ x(k+2) &= \tilde{A}^2 x(k) + \tilde{A}\tilde{B}\Delta u(k) + \tilde{B}\Delta u(k+1) \\ &\vdots \\ x(k+N_p) &= \tilde{A}^{N_p} x(k) + \tilde{A}^{N_p-1}\tilde{B}\Delta u(k) + \cdots + \tilde{A}^{N_p-N_c}\tilde{B}\Delta u(k+N_c-1) \end{aligned} \tag{4.5}$$

From the predicted state variables of equation (4.5), the predicted output variables are:

$$\begin{aligned} y(k+1) &= \tilde{C}\tilde{A}x(k) + \tilde{C}\tilde{B}\Delta u(k) \\ y(k+2) &= \tilde{C}\tilde{A}^2 x(k) + \tilde{C}\tilde{A}\tilde{B}\Delta u(k) + \tilde{C}\tilde{B}\Delta u(k+1) \\ &\vdots \\ y(k+N_p) &= \tilde{C}\tilde{A}^{N_p} x(k) + \tilde{C}\tilde{A}^{N_p-1}\tilde{B}\Delta u(k) + \cdots + \tilde{C}\tilde{A}^{N_p-N_c}\tilde{B}\Delta u(k+N_c-1) \end{aligned} \tag{4.6}$$

In compact matrix form equation (4.6) can be written as:

$$Y = \Gamma x(k) + \Phi \Delta U(k) \tag{4.7}$$

Where,

$$Y = \begin{bmatrix} y(k+1) & y(k+2) & \ldots & y(k+N_p) \end{bmatrix}^T$$

$$\Delta U = [\Delta u(k) \quad \Delta u(k+1) \quad \ldots \quad \Delta u(k+N_c-1)]^T$$

$$\Gamma = \begin{bmatrix} \tilde{C}\tilde{A} \\ \tilde{C}\tilde{A}^2 \\ \tilde{C}\tilde{A}^3 \\ \vdots \\ \tilde{C}\tilde{A}^{N_p} \end{bmatrix}, \Phi = \begin{bmatrix} \tilde{C}\tilde{B} & 0 & 0 & \ldots & 0 \\ \tilde{C}\tilde{A}\tilde{B} & \tilde{C}\tilde{B} & 0 & \ldots & 0 \\ \tilde{C}\tilde{A}^2\tilde{B} & \tilde{C}\tilde{A}\tilde{B} & \tilde{C}\tilde{B} & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \tilde{C}\tilde{A}^{N_p-1}\tilde{B} & \tilde{C}\tilde{A}^{N_p-2}\tilde{B} & \tilde{C}\tilde{A}^{N_p-3}\tilde{B} & \ldots & \tilde{C}\tilde{A}^{N_p-N_c}\tilde{B} \end{bmatrix}$$

4.3.2 LPV-MPC Controller Design for AV

This section presents a novel formulation for the MPC technique that makes use of the non-linear AV model's representation. LPV is an LTV plant with finite dimensions, if the state space matrices are fixed functions of a vector of measurable scheduling variables. The LPV model is employed by the MPC approach when it predicts future vehicle states. This suggests that representation of the non-linear vehicle model computed with a known scheduling vector is necessary at each time instant (Alcalá et al., 2020).

LPV-MPC, lies between the nonlinear and LTI formalisms. While LPV systems are nonlinear in their parameter, they exhibit linearity in the state space. This approach allows for the consideration of nonlinearities in the system by representing it as a collection of linear models, each valid within a specific range of operating conditions. In LPV-MPC, the system model is parameterized by varying parameters, and the control law is determined by solving an optimization problem at each time step. This method involves solving multiple optimization problems, one for each operating condition, which can lead to increased computational complexity (Morato et al., 2020).

LPV-MPC is well-suited for systems that exhibit nonlinearities and experience variations in operating conditions. LPV-MPC, predicts the future behavior of the system, employing an optimization algorithm to compute the control input that minimizes a predefined cost function. The advantage of the LPV approach is that it allows the representation of a nonlinear model as a combination of linear models, where the parameters of these models vary with certain scheduling variables. This approach eliminates the need for linearization. This provides a more accurate representation of the system dynamics, especially when the system exhibits significant nonlinear behavior as the AV (Besselmann & Morari, 2009).

There are several reasons for choosing LPV-MPC over other variants of MPC for trajectory tracking in AVs (Cisneros et al., 2016), Such as:

- ✓ Nonlinear models can be used with LPV-MPC to represent the dynamics of the AV, without the need for linearization.
- ✓ LPV-MPC is well-suited for handling time-varying systems, like AV systems that operate in dynamic environments where operation conditions change overtime.
- ✓ By considering the time-varying nature of the system and constraints, LPV-MPC can effectively balance stability and performance in trajectory tracking.

The continuous time-state space used for designing the LPV-MPC can be obtained from equation (3.41).

$$\dot{x}=\begin{bmatrix}\dot{V}_x\\ \dot{V}_y\\ \dot{\psi}\\ \ddot{\psi}\\ \dot{X}\\ \dot{Y}\end{bmatrix}=\begin{bmatrix}A_{11} & A_{12} & 0 & A_{14} & 0 & 0\\ 0 & A_{22} & 0 & A_{24} & 0 & 0\\ 0 & 0 & 0 & A_{34} & 0 & 0\\ 0 & A_{42} & 0 & A_{44} & 0 & 0\\ A_{51} & A_{52} & 0 & 0 & 0 & 0\\ A_{61} & A_{62} & 0 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}V_x\\ V_y\\ \psi\\ \dot{\psi}\\ X\\ Y\end{bmatrix}+\begin{bmatrix}B_{11} & B_{12}\\ B_{21} & 0\\ 0 & 0\\ B_{41} & 0\\ 0 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\delta\\ a\end{bmatrix} \tag{4.8}$$

Where,

$$A_{11}=-\frac{\mu g}{V_x},\ A_{12}=\frac{C_{\alpha f}sin(\delta)}{mV_x},\ A_{14}=\frac{C_{\alpha f}sin(\delta)l_f}{mV_x}+V_y$$

$$A_{22}=-\left(\frac{C_{\alpha r}+C_{\alpha f}cos(\delta)}{mV_x}\right),\ A_{24}=\frac{C_{\alpha r}l_r-C_{\alpha f}cos(\delta)l_f}{mV_x}-V_x$$

$$A_{34}=1, A_{42}=\frac{C_{\alpha r}l_r-C_{\alpha f}cos(\delta)l_f}{I_zV_x},\ A_{44}=-\left(\frac{C_{\alpha f}cos(\delta)l_f^2+C_{\alpha r}l_r^2}{I_zV_x}\right)$$

$$A_{51}=cos(\psi),\ A_{52}=-sin(\psi), A_{61}=sin(\psi),\ A_{62}=cos(\psi)$$

$$B_{11}=-\frac{C_{\alpha f}sin(\delta)}{m},\ B_{12}=1, B_{21}=\frac{C_{\alpha f}cos(\delta)}{m}, B_{41}=\frac{C_{\alpha f}cos(\delta)l_f}{I_z}$$

Thus, $A(V_x, V_y, \psi, \delta)$ and $B(\delta)$, the system is represented in LPV form.

LPV-MPC discretizes the continuous-time LPV formulation from equation (4.8), augments it as in equation (4.3), and enables trajectory tracking by iteratively computing future state variables using a set of control parameters. This allows real-time adjustments of control inputs considering changing system dynamics and operating conditions. The future state variables are sequentially calculated using the set of control parameters.

$$\begin{aligned}
x(k+1)&=\tilde{A}_k x(k)+\tilde{B}_k\Delta u(k)\\
x(k+2)&=\tilde{A}_{k+1}\tilde{A}_k x(k)+\tilde{A}_{k+1}\tilde{B}_k\Delta u(k)+\tilde{B}_{k+1}\Delta u(k+1)\\
&\vdots\\
x(k+N_p)&=\tilde{A}_{k+N_p-1}\tilde{A}_{k+N_p-2}\dots\tilde{A}_k x(k)+\tilde{A}_{N_p-1}\tilde{A}_{N_p-2}\dots\tilde{A}_{k+1}\tilde{B}_k\Delta u(k)\\
&\quad+\dots+\tilde{A}_{k+N_p-N_c}\tilde{B}_{k+1}\Delta u(k+1)+\dots\\
&\quad+\tilde{B}_{k+N_c-1}\Delta u(k+N_c-1)
\end{aligned} \tag{4.9}$$

By reducing the complexity of the equation (4.9) into matrix form:

$$\begin{bmatrix} x(k+1) \\ x(k+2) \\ x(k+3) \\ \vdots \\ x(k+N_p) \end{bmatrix} = \begin{bmatrix} \tilde{B}_k & 0 & \dots & 0 \\ \tilde{A}_{k+1}\tilde{B}_k & \tilde{B}_{k+1} & \dots & 0 \\ \tilde{A}_{k+2}\tilde{A}_{k+1}\tilde{B}_k & \tilde{A}_{k+1}\tilde{B}_{k+1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \prod_{i=0}^{N_p-1} \tilde{A}_{k+i}\tilde{B}_k & \prod_{i=1}^{N_p-1} \tilde{A}_{k+i}\,\tilde{B}_{k+1} & \dots & \tilde{B}_{k+N_c-1} \end{bmatrix}$$

$$* \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \Delta u(k+2) \\ \vdots \\ \Delta u(k+N_c-1) \end{bmatrix} + \begin{bmatrix} \tilde{A}_k \\ \tilde{A}_{k+1}\tilde{A}_k \\ \tilde{A}_{k+2}\tilde{A}_{k+1}\tilde{A}_k \\ \vdots \\ \prod_{i=0}^{N_p-1} \tilde{A}_{k+i} \end{bmatrix} x(k) \tag{4.10}$$

Together in compact matrix form it can be expressed as:

$$\chi = \xi \Delta U + \zeta x(k) \tag{4.11}$$

Where,

$$\chi = \begin{bmatrix} x(k+1) & x(k+2) & \dots & x(k+N_p) \end{bmatrix}^T$$
$$\Delta U = [\Delta u(k) \quad \Delta u(k+1) \quad \dots \quad \Delta u(k+N_c-1)]^T$$

$$\zeta = \begin{bmatrix} \tilde{A}_k \\ \tilde{A}_{k+1}\tilde{A}_k \\ \tilde{A}_{k+2}\tilde{A}_{k+1}\tilde{A}_k \\ \vdots \\ \prod_{i=0}^{N_p-1} \tilde{A}_{k+i} \end{bmatrix}, \ \xi = \begin{bmatrix} \tilde{B}_k & 0 & \dots & 0 \\ \tilde{A}_{k+1}\tilde{B}_k & \tilde{B}_{k+1} & \dots & 0 \\ \tilde{A}_{k+2}\tilde{A}_{k+1}\tilde{B}_k & \tilde{A}_{k+1}\tilde{B}_k & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \prod_{i=0}^{N_p-1} \tilde{A}_{k+i}\tilde{B}_k & \prod_{i=1}^{N_p-1} \tilde{A}_{k+i}\,\tilde{B}_{k+1} & \dots & \tilde{B}_{k+N_c-1} \end{bmatrix}$$

From the predicted state variables of equation (4.9), the predicted output variables are:

$$\begin{aligned} y(k+1) &= \tilde{C}\tilde{A}_k x(k) + \tilde{C}\tilde{B}_k \Delta u(k) \\ y(k+2) &= \tilde{C}\tilde{A}_{k+1}\tilde{A}_k x(k) + \tilde{C}\tilde{A}_{k+1}\tilde{B}_k \Delta u(k) + \tilde{C}\tilde{B}_{k+1}\Delta u(k+1) \\ &\vdots \\ y(k+N_p) &= \tilde{C}\tilde{A}_{N_p-1}\tilde{A}_{N_p-2} \dots \tilde{A}_k x(k) + \tilde{C}\tilde{A}_{N_p-1}\tilde{A}_{N_p-2} \dots \tilde{C}\tilde{A}_{k+1}\tilde{B}_k \Delta u(k) + \cdots \\ &\quad + \tilde{C}\tilde{A}_{N_p-N_c}\tilde{B}_{k+1}\Delta u(k+1) + \cdots + \tilde{C}\tilde{B}_{k+N_c-1}\Delta u(k+N_c-1) \end{aligned} \tag{4.12}$$

By simplifying the above equation (4.12) on matrix form:

$$\begin{bmatrix} y(k+1) \\ y(k+2) \\ y(k+3) \\ \vdots \\ y(k+N_p) \end{bmatrix} = \begin{bmatrix} \tilde{C}\tilde{B}_k & 0 & \dots & 0 \\ \tilde{C}\tilde{A}_{k+1}\tilde{B}_k & \tilde{C}\tilde{B}_{k+1} & \dots & 0 \\ \tilde{C}\tilde{A}_{k+2}\tilde{A}_{k+1}\tilde{B}_k & \tilde{C}\tilde{A}_{k+1}\tilde{B}_{k+1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{C}\prod_{i=0}^{N_p-1}\tilde{A}_{k+i}\tilde{B}_k & \tilde{C}\prod_{i=1}^{N_p-1}\tilde{A}_{k+i}\,\tilde{B}_{k+1} & \dots & \tilde{C}\tilde{B}_{k+N_c-1} \end{bmatrix}$$

$$* \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \Delta u(k+2) \\ \vdots \\ \Delta u(k+N_c-1) \end{bmatrix} + \begin{bmatrix} \tilde{C}\tilde{A}_k \\ \tilde{C}\tilde{A}_{k+1}\tilde{A}_k \\ \tilde{C}\tilde{A}_{k+2}\tilde{A}_{k+1}\tilde{A}_k \\ \vdots \\ \tilde{C}\prod_{i=0}^{N_p-1}\tilde{A}_{k+i} \end{bmatrix} x(k) \tag{4.13}$$

The system output expression in the prediction horizon where, $N_p \geq N_c$ would be as follows:

In compact matrix form equation (4.13) can be written as:

$$Y = \Gamma x(k) + \Phi \Delta U(k) \tag{4.14}$$

Where,

$$Y = \left[y(k+1) \quad y(k+2) \quad \dots \quad y(k+N_p)\right]^T$$
$$\Delta U = [\Delta u(k) \quad \Delta u(k+1) \quad \dots \quad \Delta u(k+N_c-1)]^T$$

$$\Gamma = \begin{bmatrix} \tilde{C}\tilde{A}_k \\ \tilde{C}\tilde{A}_{k+1}\tilde{A}_k \\ \tilde{C}\tilde{A}_{k+2}\tilde{A}_{k+1}\tilde{A}_k \\ \vdots \\ \tilde{C}\prod_{i=0}^{N_p-1}\tilde{A}_{k+i} \end{bmatrix}, \Phi = \begin{bmatrix} \tilde{C}\tilde{B}_k & 0 & \dots & 0 \\ \tilde{C}\tilde{A}_{k+1}\tilde{B}_k & \tilde{C}\tilde{B}_{k+1} & \dots & 0 \\ \tilde{C}\tilde{A}_{k+2}\tilde{A}_{k+1}\tilde{B}_k & \tilde{C}\tilde{A}_{k+1}\tilde{B}_k & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{C}\prod_{i=0}^{N_p-1}\tilde{A}_{k+i}\tilde{B}_k & \tilde{C}\prod_{i=1}^{N_p-1}\tilde{A}_{k+i}\,\tilde{B}_{k+1} & \dots & \tilde{C}\tilde{B}_{k+N_c-1} \end{bmatrix}$$

4.3.3 Cost Function

In AV, the objective of optimal control is to closely follow a desired trajectory. The optimization is focused on reducing the deviation between the desired trajectory and the actual state of the AV, by taking into account system dynamics and constraints. The control inputs are modified in real-time to follow the reference trajectory as closely as possible. To continuously modify the control inputs and enhance the AV's tracking performance, optimization is carried out iteratively at each time step.

The LPV-MPC algorithm optimizes the control inputs by solving an optimization problem based on the cost function over a finite time horizon. The cost function in LPV-MPC for AV tracking is to achieve optimal control inputs by accurately following the desired trajectory. It encompasses various terms that penalize tracking errors, control efforts. The goal of the cost function is to find optimal control input variations that drive the system states towards the desired reference trajectory while satisfying the system dynamics and constraints.

The control objective function of MPC can be defined as follows:

$$\begin{gathered}
J(k) = \sum_{i=1}^{N_p} \left(R_{ref} - \chi\right)^T Q \left(R_{ref} - \chi\right) + \sum_{i=0}^{N_c - 1} \Delta u(k+i)^T R \Delta u(k+i) \\
s.t. \qquad \chi = \left[x(k+1) \quad x(k+2) \quad \dots \quad x\left(k+N_p\right)\right]^T \\
\Delta u(k+i) = u(k+i) - u(k+i-1), \\
\Delta u_{min}(k) \le \Delta u(k) \le \Delta u_{max}(k), k = 0,1,\cdots,N_c - 1, \\
u_{min}(k) \le u(k) \le u_{max}(k), k = 0,1,\cdots,N_c - 1, \\
x_{min}(k) \le x(k) \le x_{max}, k = 0,1,\cdots,N_c - 1
\end{gathered} \tag{4.15}$$

Where, $Q \in \mathbb{R}^{6\times6}$ and $R \in \mathbb{R}^{2\times2}$ are positive weighting matrices and R_{ref} is the desired reference trajectory.

The first term in equation (4.15) is the tracking error term, which quantifies the deviation between the AV states and the desired reference trajectory. It penalizes the tracking errors to ensure the vehicle closely follows the desired trajectory. The second term is the control effort term and it penalizes the magnitude of rate of change of control inputs, required to achieve the desired trajectory. By adjusting the weights in the state weighting matrix Q and the control weighting matrix R, the importance between tracking accuracy and control effort can be balanced (Hu, Qin, et al., 2019).

Through careful formulation and tuning of the cost function, the MPC controller can effectively track the desired trajectory. The cost function is designed to take into account various control objectives and constraints, to ensure the controller operates within acceptable limits to achieve accurate trajectory tracking. By optimizing the control inputs based on the cost function, the MPC controller can navigate the AV along the desired path.

4.3.4 Constraints

Handling constraints effectively is important for the safe and reliable operation of AVs. By properly including constraints into the control algorithms, AVs can navigate through complex environments while avoiding violations and unsafe situations. For AVs to operate safely and reliably, there must be constraints on their design and operation. These constraints specify the restrictions and limitations, which the AV must operate (Greenblatt, 2016). The input steering wheel angle and the applied longitudinal acceleration which is produced by throttle or brake is controlled within reasonable range. AVs have many constraints, such as speed restrictions, acceleration limits, steering angle restrictions and physical vehicle functioning limits. In order to guarantee safe and normal operation of AVs, it is necessary to restrict control inputs and some of the outputs in the constraint form in Table 4.1 which then will be included in the cost function. In the specific AV model considered in this thesis work, several constraints are taken into account to ensure safe and reliable operation as in (Chen et al., 2020). These constraints include:

Table 4.1: Constraints of the MPC controller for AV

Parameters	**Min value**	**Max value**
Steering wheel angle (δ)	$-\pi/6$	$\pi/6$
variation ($\Delta\delta$)	$-\pi/300$	$\pi/300$
Longitudinal acceleration (a)	$-4 + \frac{F_{yf}\sin(\delta)}{m} + \mu g - V_y\dot{\psi}$	$1 + \frac{F_{yf}\sin(\delta)}{m} + \mu g - V_y\dot{\psi}$
variation (Δa)	$-0.1 m/s^2$	$0.1 m/s^2$
Longitudinal V_x	$1 m/s$	$30 m/s$
Lateral V_y	$-0.1V_x$	$0.1V_x$

Constraints are applied in the steering wheel and acceleration, control inputs and some other state variables to make sure the AV stays within reasonable limits to track the desired trajectory and maintain safety regulations.

The predicted input signals from equation (4.2) can be expressed as:

$$
\begin{aligned}
u(k) &= u(k-1) + \Delta u(k) \\
u(k+1) &= u(k-1) + \Delta u(k) + \Delta u(k+1) \\
&\vdots \\
u(k+N_c-1) &= u(k-1) + \Delta U^T
\end{aligned}
\tag{4.16}
$$

An inequality constraint equation can be created when placing restrictions on the control signal expressed in equation (4.16) of trajectory tracking for AVs.

$$
\begin{aligned}
&u_{min} \le u(k+N_c-1) \le u_{max} \\
&u_{min} \le u(k-1) + \Delta U^T \le u_{max} \\
&u_{min} - u(k-1) \le \Delta U^T \le u_{max} - u(k-1) \\
&\Delta U^T \le u_{max} - u(k-1) \\
&-\Delta U^T \le -u_{min} + u(k-1)
\end{aligned}
\tag{4.17}
$$

From equation (4.16) and (4.17) the general upper and lower expression of input constraints is expressed as:

$$
\begin{bmatrix} I \\ -I \end{bmatrix} \Delta U \le \begin{bmatrix} u_{max} - u(k-1) \\ -u_{min} + u(k-1) \end{bmatrix} \tag{4.18}
$$

And for the state constraints the following formulations are made.

$$
y_k^* = \begin{bmatrix} V_{x_k} \\ V_{y_k} \\ \delta_{k-1} \\ a_{k-1} \end{bmatrix} = \tilde{C}^* x(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_{x_k} \\ V_{y_k} \\ \psi_k \\ \dot{\psi}_k \\ X_k \\ Y_k \\ \delta_{k-1} \\ a_{k-1} \end{bmatrix} \tag{4.19}
$$

Where y_k^* is the state constrained matrix which contains $[V_{x_k} \quad V_{y_k} \quad \delta_{k-1} \quad a_{k-1}]^T$.

$$
\begin{aligned}
&y_{k_{min}}^* \le y_k^* \le y_{k_{max}}^* \\
&\begin{cases} -y_k^* \le -y_{k_{min}}^* \\ \;\; y_k^* \le y_{k_{max}}^* \end{cases}
\begin{cases} -\tilde{C}^* \xi \Delta U - \tilde{C}^* \zeta x(k) \le -y_{k_{min}}^* \\ \;\; \tilde{C}^* \xi \Delta U + \tilde{C}^* \zeta x(k) \le y_{k_{max}}^* \end{cases}
\end{aligned}
\tag{4.20}
$$

From equation (4.20), the upper and lower expression of state constraints is expressed as:

$$
\begin{bmatrix} \tilde{C}^* \xi \Delta U \\ -\tilde{C}^* \xi \Delta U \end{bmatrix} \Delta U \le \begin{bmatrix} y_{k_{max}}^* - \tilde{C}^* \zeta x(k) \\ -y_{k_{min}}^* + \tilde{C}^* \zeta x(k) \end{bmatrix} \tag{4.21}
$$

By using equations (4.18) and (4.21) the general input and state inequality constraint equation is provided as:

$$
A * \Delta U \le b \tag{4.22}
$$

Where, $A = \begin{bmatrix} I \\ -I \\ \tilde{C}^*\xi\Delta U \\ -\tilde{C}^*\xi\Delta U \end{bmatrix}$, $b = \begin{bmatrix} u_{max} - u(k-1) \\ -u_{min} + u(k-1) \\ y^*_{k_{max}} - \tilde{C}^*\zeta x(k) \\ -y^*_{k_{min}} + \tilde{C}^*\zeta x(k) \end{bmatrix}$

Due to the presence of inequality constraints, finding the optimal value of ΔU becomes challenging. To address this, the built-in MATLAB function "$quadprog$" is utilized.

$$\Delta U = quadprog(H, f, A, b) \tag{4.23}$$

Where, $H = \Gamma^T Q \Gamma + R$, and $f = -\Gamma^T Q\big(R_{ref} - \Phi x(k)\big)$

The ΔU obtained is specific to the current time of the controlled AV. As the system progresses to the next time point, a new prediction of the system output is generated. To address this, the objective function is solved again to calculate a new control sequence for the AV at the next time point (Nouwens et al., 2021). This process allows for obtaining an optimal control increment that corresponds to the desired control actions needed to achieve the desired system performance. By iteratively updating the control sequence based on the changing system dynamics and objectives, the controller can continuously adapt and generate control inputs suitable for the evolving conditions and time steps (Pang et al., 2022).

The cost function in LPV-MPC is designed to capture the control objectives, which are trajectory tracking and constraint satisfaction requirements. The optimization algorithm used in this thesis specifically MATLAB's built-in function "$quadprog$", is quadratic programming, that directly optimizes the cost function subject to constraints.

4.4 Weighting Matrices Tuning Using PSO for Controller Optimization

PSO is a population-based stochastic optimization algorithm that is inspired by the social behavior of fish schools and bird flocks. Using their individual best solution and the most recent global best solution, a population of particles in PSO iteratively updates their positions in search of the ideal answer. PSO's simplicity and quick convergence characteristics make it a good choice for real-time trajectory optimization in MPC. Since it can manage both continuous variables and restrictions, a variety of optimization issues can be solved with it. PSO makes it possible to effectively explore and exploit the search space, producing efficient and reliable optimization outcomes by taking use of the social behavior of particles and their capacity to communicate and share information (Boumaza et al., 2018).

The weighting matrices employed in the controller design in this thesis are adjusted using the PSO algorithm. By adjusting these matrices iteratively to minimize a particular objective function, PSO is used to determine the best values for them.

Table 4.2: PSO parameters

Parameters for Algorithm	**Value**
Maximum generation	30
Population size	50
No of parameters	8
Lower Boundary	[0 0 0 0 0 0 0 0]
Upper Boundary	[50 10 10 10 100 100 1 10]

The maximum and minimum values of the weighting matrices indicate the priority given to the corresponding states that the weighting matrices supposed to multiply in the objective function. By adjusting these values, the trade-off between tracking performance and energy efficiency can be controlled.

CHAPTER FIVE

5 RESULT AND DISCUSSION

5.1 Chapter Overview

The results and analysis of the proposed LPV-MPC approach are the main topics of this chapter, which also compares it to Linear MPC-controlled AV. MATLAB scripts are created depending on the particular controller in order to mimic the system response and evaluate the controller's efficacy. These MATLAB scripts were thoughtfully created to investigate the control strategy in great detail. In particular, the parameters used for the MATLAB simulations are reported in Table 3.1 and are based on the Chevrolet SUV car as in (Alcalá et al., 2019).

5.2 Simulation Results

The simulation results of the AV's trajectory tracking provide the properties and behavior during operation. The effectiveness and efficiency of the LPV-MPC and Linear MPC controllers in tracking accuracy of the trajectory tracking can be analyzed through these simulations. The simulations also make it possible to evaluate how the AV responds to various road conditions.

The simulation results provide a complete analysis of the AV's performance in a number of aspects, including trajectory tracking, longitudinal and lateral control, and error analysis. These results make it possible to assess how well the AV maintains the necessary acceleration, steers precisely along specified trajectories, and adjusts to changing environmental conditions. The results of the simulation under different conditions explain capabilities and limitations of the LPV-MPC and Linear MPC controlled AVs.

5.2.1 Trajectory Tracking of a Linear Reference (Ramp Reference)

This section focuses on assessing the trajectory tracking capabilities of an AV controlled with LPV-MPC and compares it with Linear MPC controlled AV. The AV follows a linear, straight-line reference trajectory. Tracking accuracy is used as the primary criterion to assess how well LPV-MPC and Linear MPC perform in comparison. The goal is to determine how closely the AV can follow to the required trajectory. The PSO method is applied to obtain optimal trajectory tracking. PSO aids in choosing the optimum weighting matrices for the control method. The goal is to improve the AV's capability to reliably track a linear reference trajectory by utilizing the LPV-MPC technique and optimizing the weighting matrices through the PSO algorithm. The effectiveness can be evaluated by the comparison with Linear MPC.

The straight-line ramp trajectory is defined as follows:

$$R_{ref} = \begin{cases} X = 15 * t \\ Y = 15 * t \end{cases} \tag{5.1}$$

The simulation time is 50sec, and the prediction horizon is 10.

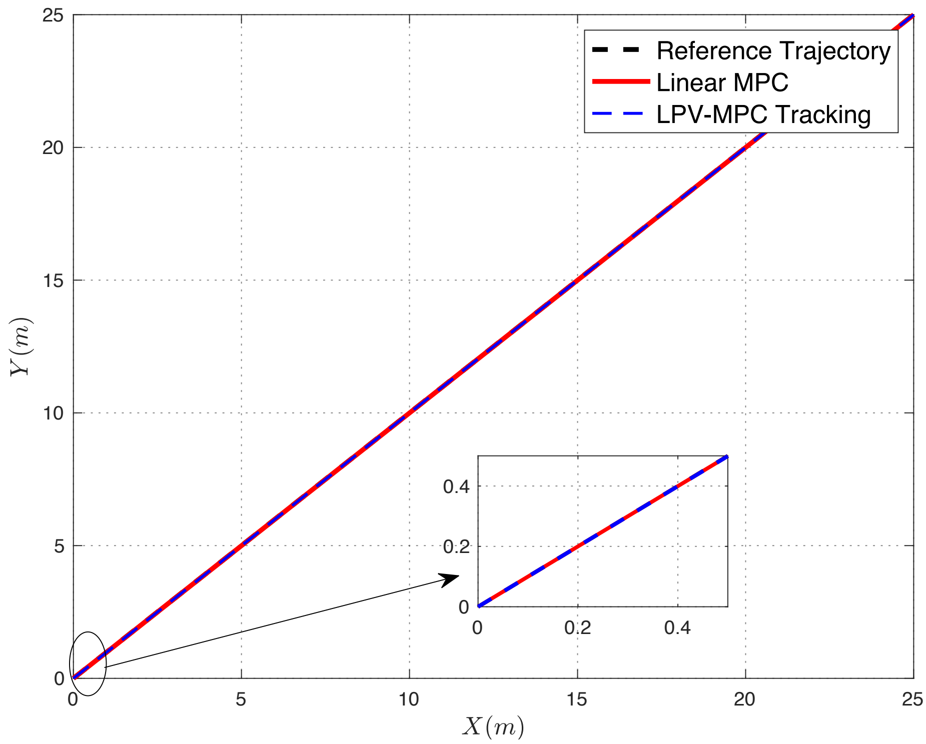

Figure 5.1: Comparison of linear ramp Trajectory Tracking: LPV-MPC vs Linear MPC

The obtained optimized weighting matrices have the following values, which have been tuned through the PSO algorithm:

$Q = diag([32.74 \quad 6.5e-2 \quad .863 \quad 4.17e-3 \quad 48.736 \quad 26.912])$ and

$R = diag([.297 \quad .846])$

Figure 5.1 illustrates the linear straight line road trajectory tracking performance of the AV under the control of both LPV-MPC and Linear MPC, utilizing the aforementioned weighting matrices. Both control strategies demonstrate remarkable capabilities in enabling the AV to closely follow the desired linear ramp reference route. As a result, the AV exhibits a very minimal deviations from the intended trajectory, showcasing its ability to maintain high trajectory accuracy. This impressive level of accuracy highlights the effectiveness of the control algorithms employed, which enable the AV to achieve precise adherence to the specified reference road.

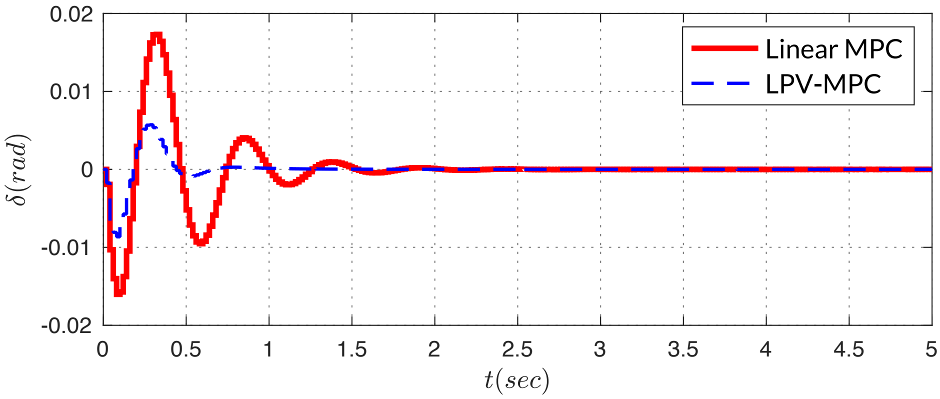

Figure 5.2: Steering angle (δ) control inputs for ramp reference (straight road)

In Figure 5.2, the control input δ for the designated linear straight-line reference road is presented, showcasing a comparison between the performance of LPV-MPC and Linear MPC. The plot reveals that the control input δ generated by LPV-MPC exhibits reduced overshoot and settling time in comparison to Linear MPC. This observation emphasizes the efficiency and effectiveness of LPV-MPC in achieving precise control of the AV's steering. The reduced overshoot signifies that the AV controlled by LPV-MPC reaches the desired steering angle with minimal deviation, minimizing any oscillatory behavior. Additionally, the reduced settling time indicates that the AV reaches its desired steering angle quickly and settles into the desired trajectory faster, improving the overall responsiveness of the control system.

Figure 5.3 illustrates the control input a for the designated linear ramp reference road, comparing the performance of LPV-MPC and Linear MPC. The graph shows how the acceleration is dynamically adjusted over time to accurately track the desired trajectory. A closer at the plot, reveals LPV-MPC exhibits slightly higher overshoot compared to Linear MPC. Which implies that the AV controlled by LPV-MPC experiences a temporary deviation from the desired acceleration, resulting in a brief period of acceleration overshoot. However, LPV-MPC demonstrates a faster settling time, indicating its ability to reach the desired trajectory more quickly and efficiently compared to Linear MPC.

The use of a time-varying model, which more correctly represents the dynamics of the system under various operating situations, is responsible for LPV-MPC's enhanced trajectory tracking skills. The LPV-MPC produces control inputs that better adapt to the unique road circumstances and optimize the trajectory tracking performance by taking into account the varying dynamics and uncertainties of the AV system.

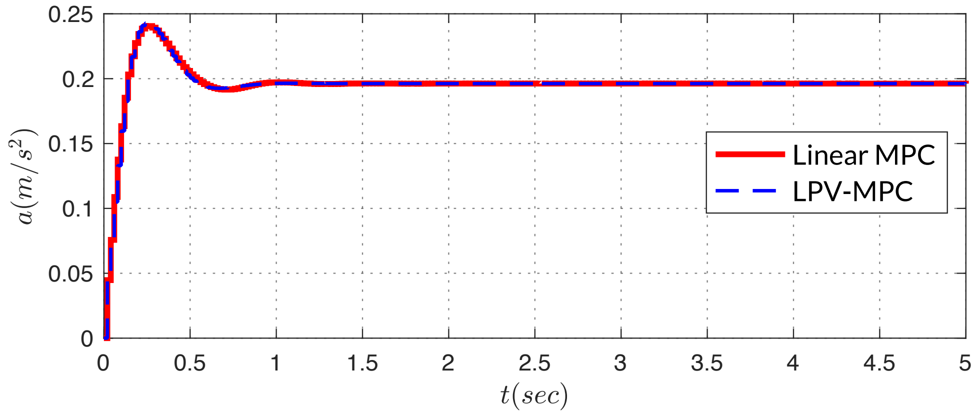

Figure 5.3: Longitudinal acceleration (a) control inputs for linear road (straight road)

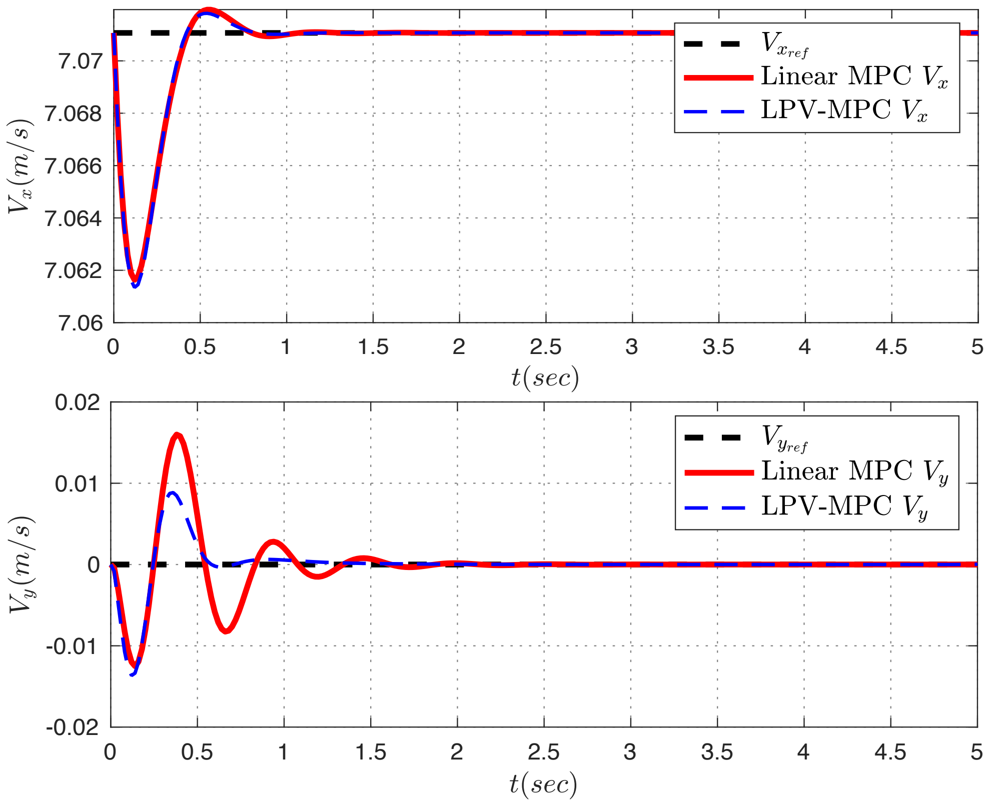

Figure 5.4: a) Longitudinal velocity (V_x) comparison on linear ramp road b) Lateral velocity (V_y) comparison linear ramp road

The behavior of the AV controlled by LPV-MPC and Linear MPC is contrasted with the reference ramp trajectory in Figure 5.4 (a) and (b). Along with the reference velocity on the linear route, the plots display the AV's longitudinal velocity (V_x) and lateral velocity (V_y) with time. When LPV-MPC and Linear MPC are compared in terms of performance, LPV-MPC comes out slightly ahead. Due to its capacity to take into consideration the time-varying nature of the AV's dynamics and its adaptation to different operating situations, it exhibits decreased overshoot and settling time in both V_x and V_y, exhibiting the ability to produce more accurate and responsive control. More proof that the LPV-MPC control algorithm tracks the AV's longitudinal and lateral velocities is provided by this. Additionally, it makes it possible for the LPV-MPC controller to provide smoother and more accurate control, which leads to less overshoot and a quicker settling time than Linear MPC.

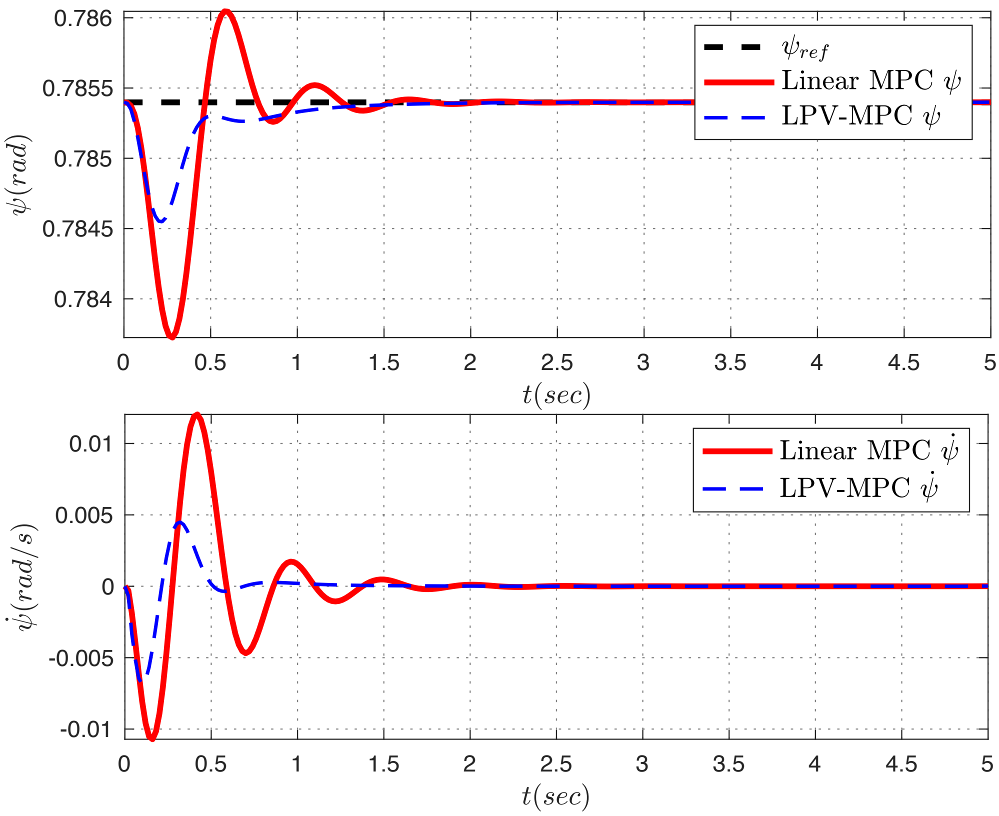

Figure 5.5: a) Yaw angle (ψ) response to trajectory on straight road b) Yaw angle rate ($\dot{\psi}$) response on linear road

Figure 5.5 (a) illustrates the AV's yaw angle (ψ) response when controlled by LPV-MPC and Linear MPC, as it aligns with the desired linear trajectory on the straight road. The plot clearly demonstrates the superior performance of LPV-MPC in accurately controlling the AV's yaw angle and ensuring its alignment with the desired trajectory.

As the AV follows the linear reference trajectory on the ramp road, the response of the AV's yaw rate ($\dot{\psi}$) is displayed for both the LPV-MPC and Linear MPC in Figure 5.5 (b). This graph shows the AV's yaw rate as it changes over time, demonstrating its capacity to correct for and maintain its yaw angle at various locations along the linear path. Once again, it is clear that LPV-MPC performance surpasses the Linear MPC in terms of efficiently regulating the AV's yaw rate and guaranteeing steady yaw angle adjustment.

The comparison between LPV-MPC and Linear MPC in terms of yaw angle and yaw rate responses further highlights the advantages of utilizing the LPV-MPC control algorithm. The LPV-MPC approach, with its ability to capture the time-varying dynamics of the AV, enables more accurate and precise control of the yaw angle and yaw rate, resulting in better alignment with the desired trajectory and improved stability throughout the maneuver.

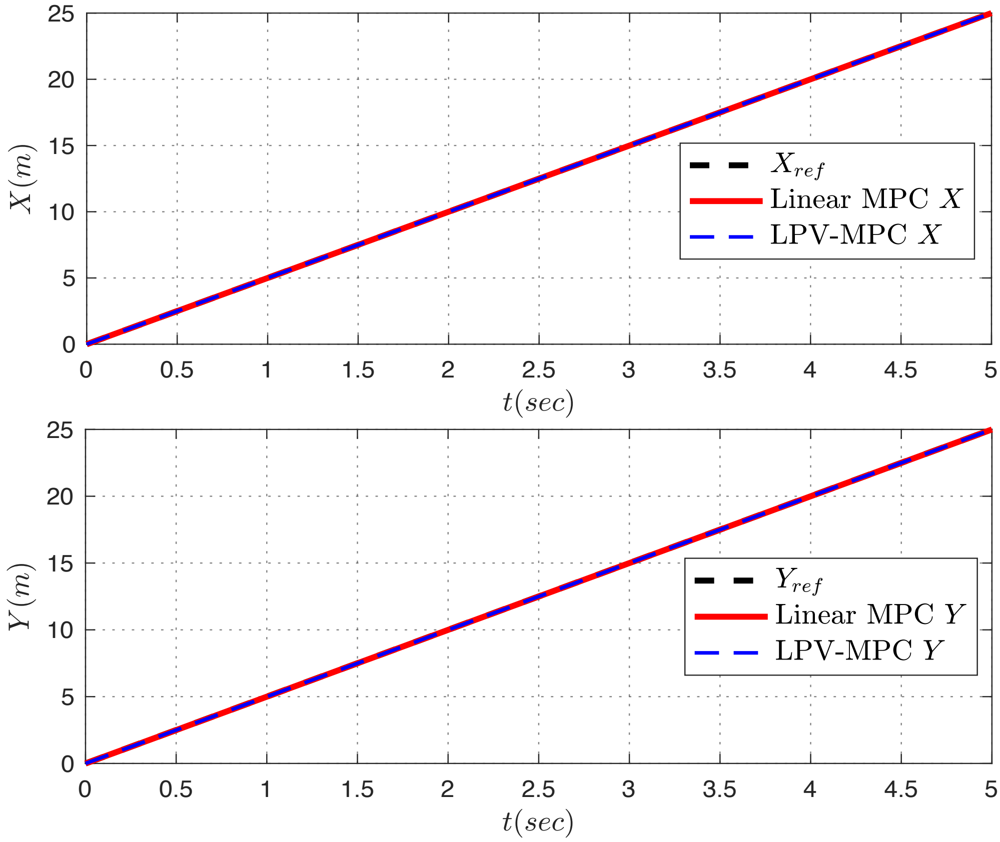

Figure 5.6: a) X position tracking of AV on ramp road b) Y position tracking of AV on ramp road

Figure 5.6 illustrates the tracking performance of the AV on a linear ramp road, controlled by the LPV-MPC and Linear MPC controllers. The plot displays the X and Y position of the AV over time while following a $750m$ trajectory. Both controllers demonstrate remarkable precision in tracking the desired trajectory. The graphical representation of the AV's X and Y coordinates showcases its ability to closely follow the specified linear reference trajectory. The figures offer a comprehensive evaluation of the AV's position accuracy along the linear route, emphasizing the effectiveness of the MPC controller in achieving precise tracking.

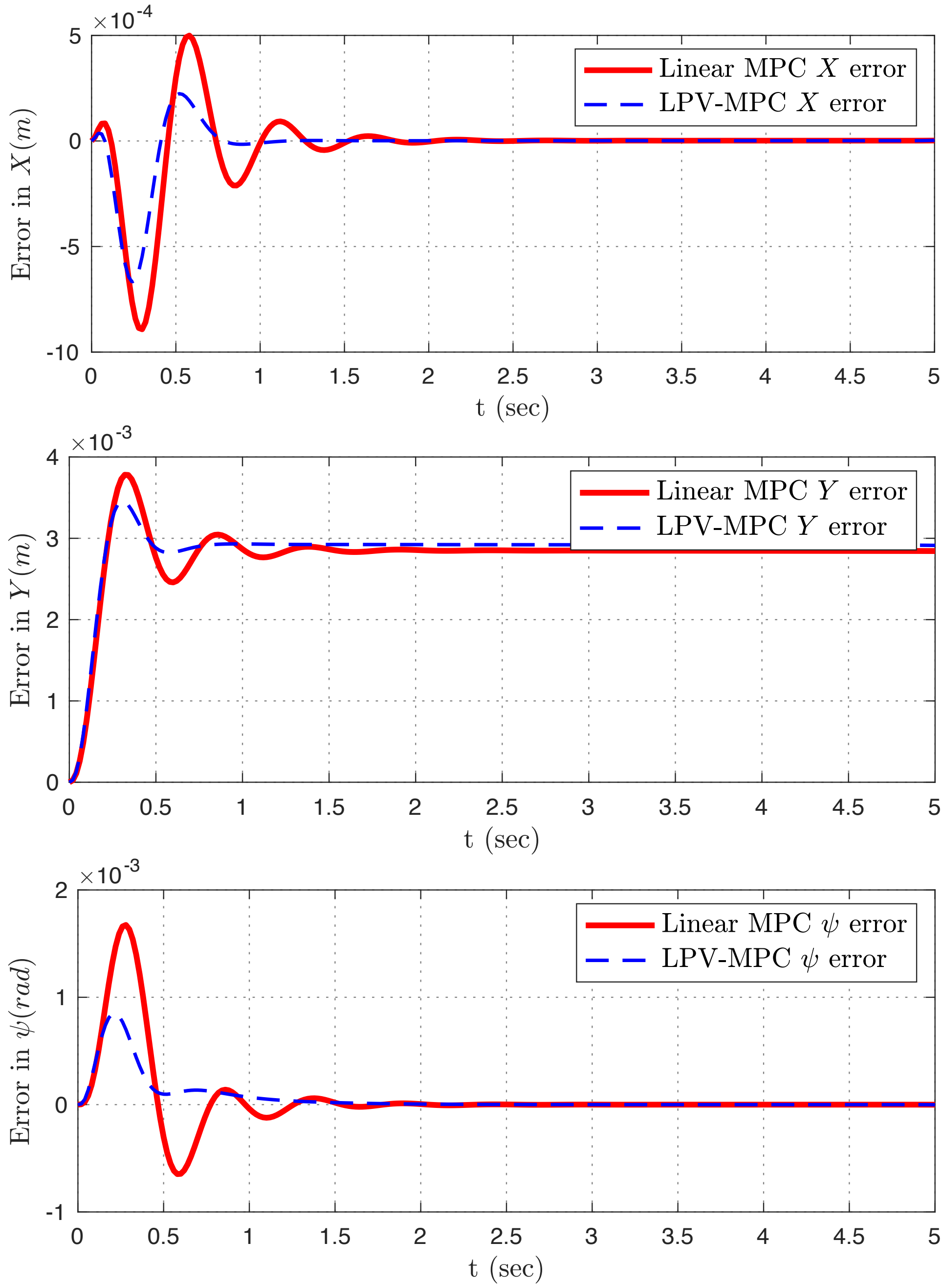

Figure 5. 7: a) X position error on linear road b) Y position error on linear road c) ψ error on linear road

Figure 5. *7* showcases the deviations between the actual X and Y positions, as well as the ψ of the AV under the control of LPV-MPC and Linear MPC, in comparison to the corresponding positions along the desired reference trajectory on the linear straight route. Both controllers demonstrate remarkable accuracy, with minimal errors and deviations that are close to zero.

The plots provide a clear visualization of the AV's ability to closely track the desired reference trajectory, as evidenced by the minimal discrepancies in the X and Y positions and the heading angle. These results highlight the effectiveness of MPC controller in achieving precise trajectory tracking and emphasize their capability to minimize positional errors.

5.2.2 Trajectory Tracking of Nonlinear Reference (Parabolic Reference)

This analysis compares the performance of an AV controlled by the LPV-MPC strategy to that of the Linear MPC approach in terms of a parabolic reference trajectory, which includes navigating a curved path. The primary purpose is to assess the precision of tracking achieved by both control strategies as the AV follows the planned parabolic trajectory. This assessment sheds light on the effectiveness of LPV-MPC and Linear MPC in controlling the AV's steering and acceleration in order to maintain accurate adherence to the curved course.

$$R_{ref} = \begin{cases} X = 15 * t \\ Y = \left(\frac{750}{900^2}\right) * X^2 + 250 \end{cases} \quad (5.2)$$

The PSO algorithm is used, to determine the optimized weighting matrices. The obtained optimized weighting matrices have the following values, which have been tuned.

$Q = diag([18.491 \quad 9.183e-2 \quad 1.013 \quad 1.6e-3 \quad 57.173 \quad 16.192])$ and

$R = diag([.343 \quad 0.802])$

The simulation time is 50sec, and the prediction horizon is 10.

In Figure 5.8, the performance of LPV-MPC and Linear MPC is compared for tracking a given parabolic reference trajectory. LPV-MPC exhibits a slight performance advantage over Linear MPC with slightly small deviation from the reference. The advantage observed in the parabolic trajectory tracking performance of LPV-MPC are attributed to its ability to handle the varying dynamics of the system more effectively. To better adapt to varying operating conditions, LPV-MPC uses a varying model that captures the nonlinearities of the AV’s dynamics. In contrast, Linear MPC focuses on linearization, which simplifies the systems dynamics by assuming a fixed linear model. Therefore, the LPV-MPC approach is better suited to handle the highly nonlinear dynamics of the AV and adapt to changing operating conditions, leading to improved trajectory tracking performance.

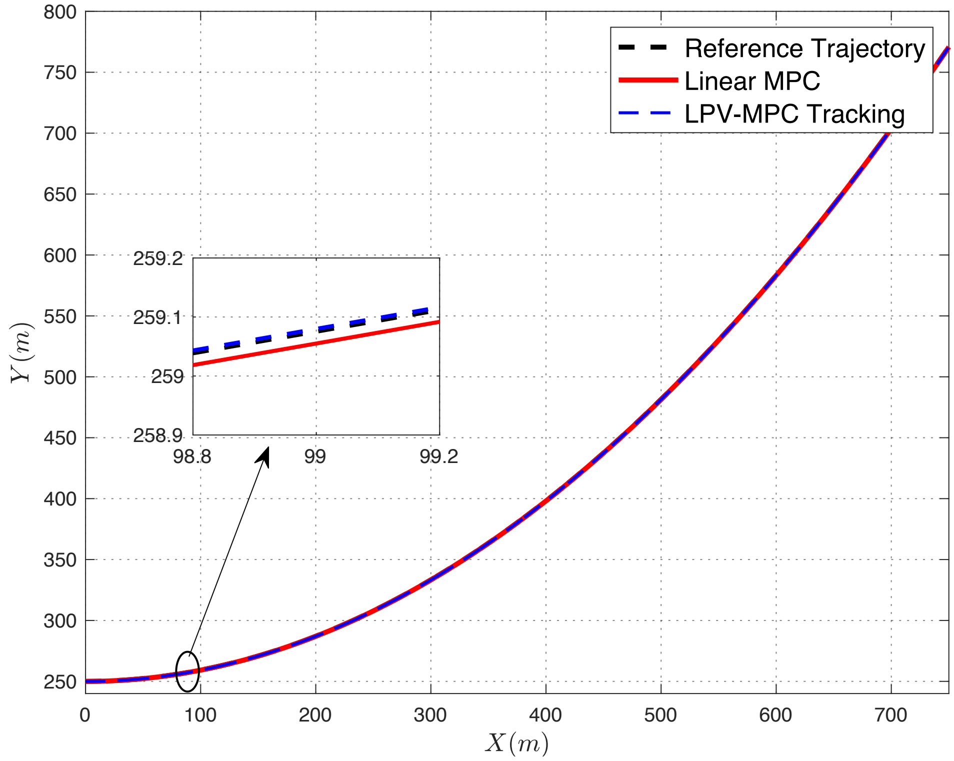

Figure 5.8: Comparison of Parabolic Trajectory Tracking: LPV-MPC vs Linear MPC

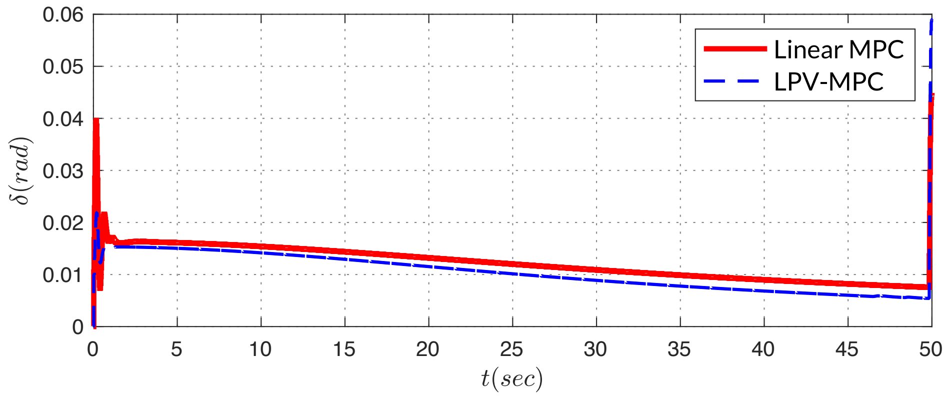

Figure 5.9: Steering angle (δ) control inputs for parabolic reference

The steering angle (δ) control input for the parabolic reference trajectory is shown in Figure 5.9, and the results show that both LPV-MPC and Linear MPC exhibit identical steering angles with little variation. The behavior of Linear MPC, however, shows a slight difference because it exhibits a tiny overshoot. This suggests that the Linear MPC controller likely react more aggressively, resulting in a brief departure from the target steering angle. However, LPV-MPC exhibits more accurate steering angle control, leading to smoother and more precise tracking of the parabolic reference trajectory.

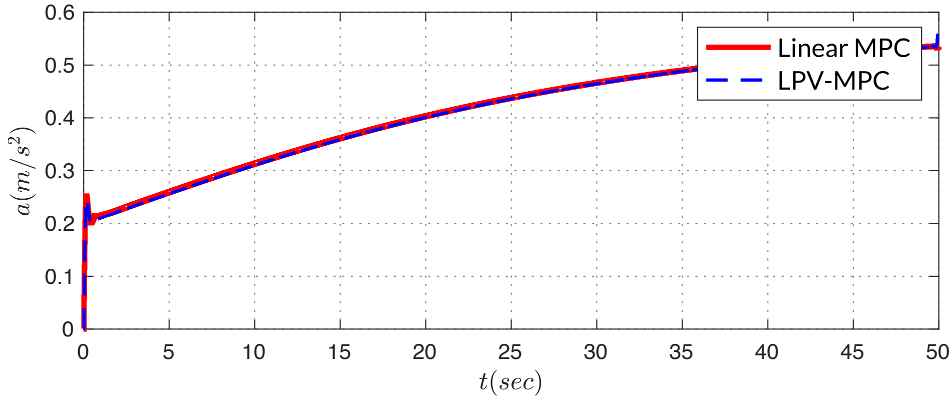

Figure 5.10: Longitudinal acceleration (a) control inputs for parabolic reference

Figure 5.10 illustrates the longitudinal acceleration (a) control input for the parabolic reference trajectory in both LPV-MPC and Linear MPC. The control inputs generated by both approaches exhibit a close resemblance, indicating that they yield similar longitudinal acceleration inputs. This similarity suggests that both LPV-MPC and Linear MPC effectively adjust the acceleration of the AV to track the desired parabolic trajectory. The comparable control inputs reflect the capability of both controllers to maintain consistent and appropriate acceleration levels, ensuring smooth and accurate trajectory tracking.

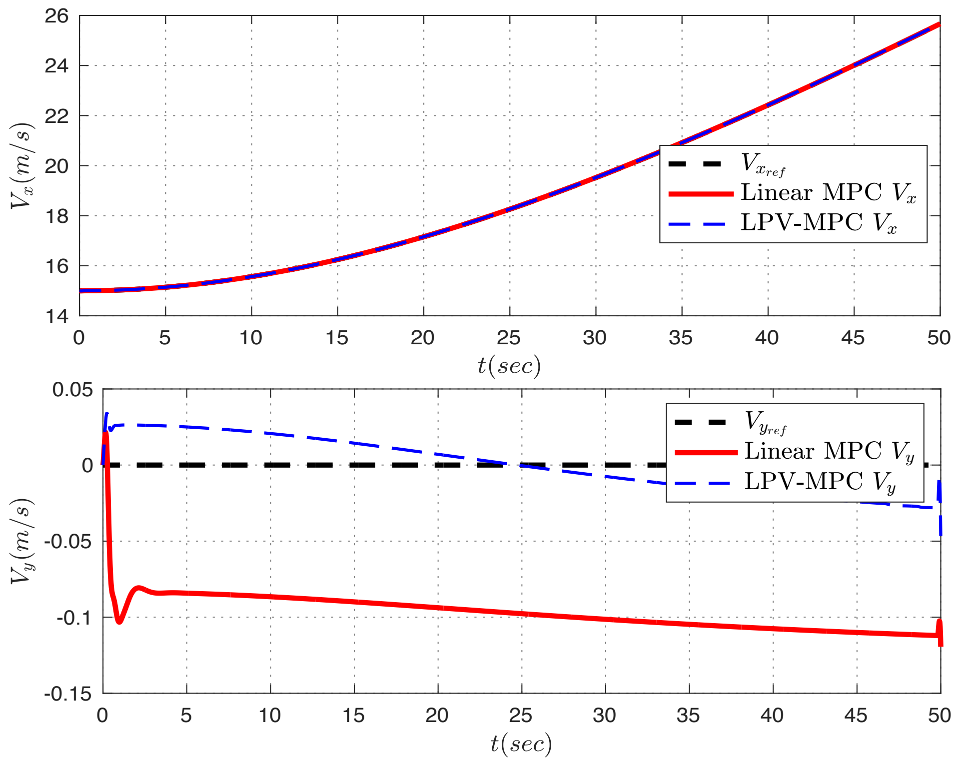

Figure 5.11: a) Longitudinal velocity (V_x) comparison on parabolic reference b) Lateral velocity (V_y) comparison on parabolic reference

Figure 5.11 illustrates the comparison between Linear MPC and LPV-MPC for trajectory tracking of a parabolic reference trajectory, focusing on the (a) longitudinal velocity (V_x) and

(b) lateral velocity (V_y) components. Upon analyzing the figure, it is evident that Linear MPC exhibits a deviation in the V_y component from the reference value of $V_y = 0$. This deviation is primarily caused by the linearization process employed in Linear MPC. In contrast, LPV-MPC, which can handle the highly nonlinear dynamics of the AV, shows better trajectory tracking performance with reduced deviation in the V_y component. LPV-MPC takes into account the varying models of the AV over time, allowing for more accurate and robust control actions. By considering the nonlinear behavior of the AV, LPV-MPC is able to better capture the dynamics and uncertainties of the system, resulting in improved trajectory tracking performance.

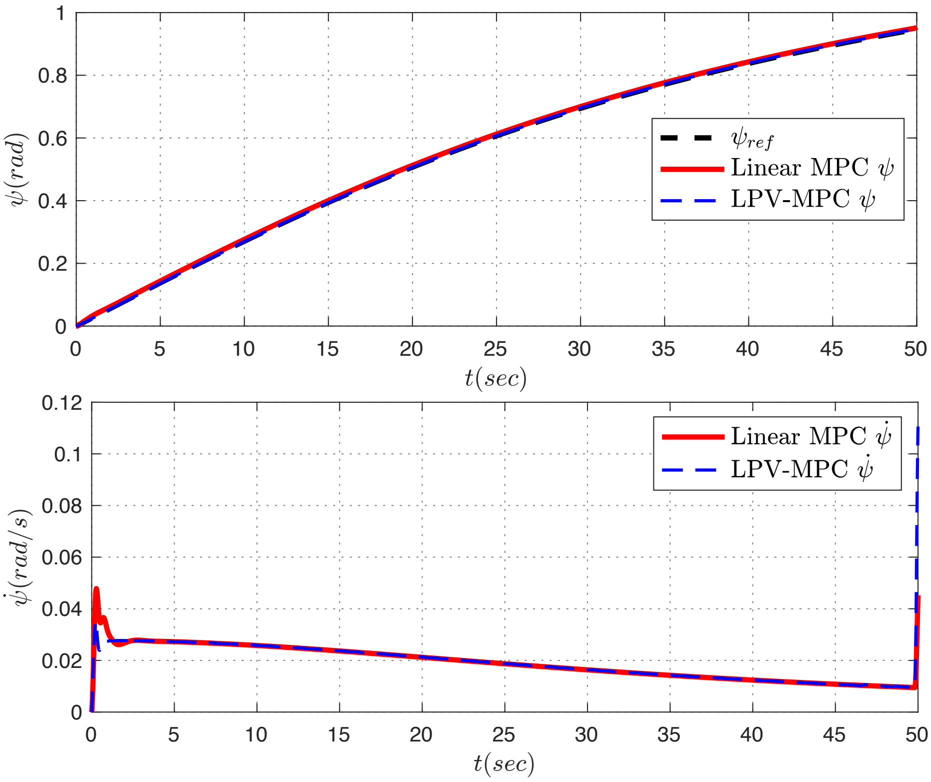

Figure 5.12: a) Yaw angle (ψ) response to trajectory on parabolic reference b) Yaw angle rate ($\dot{\psi}$) response on parabolic reference

In Figure 5.12, the responses of the yaw angle (ψ) and yaw angle rate ($\dot{\psi}$) to a parabolic reference trajectory are depicted for LPV-MPC, along with a comparison to Linear MPC. The response of the yaw angle (ψ) shown on (a) to the parabolic reference trajectory shows that both LPV-MPC and linear MPC are able to track the desired yaw angle reasonably well. However, LPV-MPC exhibits a slightly more accurate and smooth response, closely following the reference trajectory without significant deviations. And the response of the yaw angle rate ($\dot{\psi}$) in (b) to the parabolic reference trajectory.

In Figure 5.13, the comparison of tracking performance using LPV-MPC and Linear MPC of the AV's position on a parabolic reference trajectory is presented, focusing on the (a) X position and (b) Y position components. The X position tracking of the AV illustrates the ability of the controller to accurately follow the desired X position along the parabolic reference trajectory. Both LPV-MPC and linear MPC exhibit satisfactory performance, closely tracking the reference trajectory without significant deviations. And the Y position tracking of the AV demonstrates the controller's capability to accurately track the desired Y position perpendicular to the reference trajectory. The ability of LPV-MPC to account for the nonlinear dynamics of the AV enables it to achieve more accurate and precise trajectory tracking, leading to enhanced overall control performance.

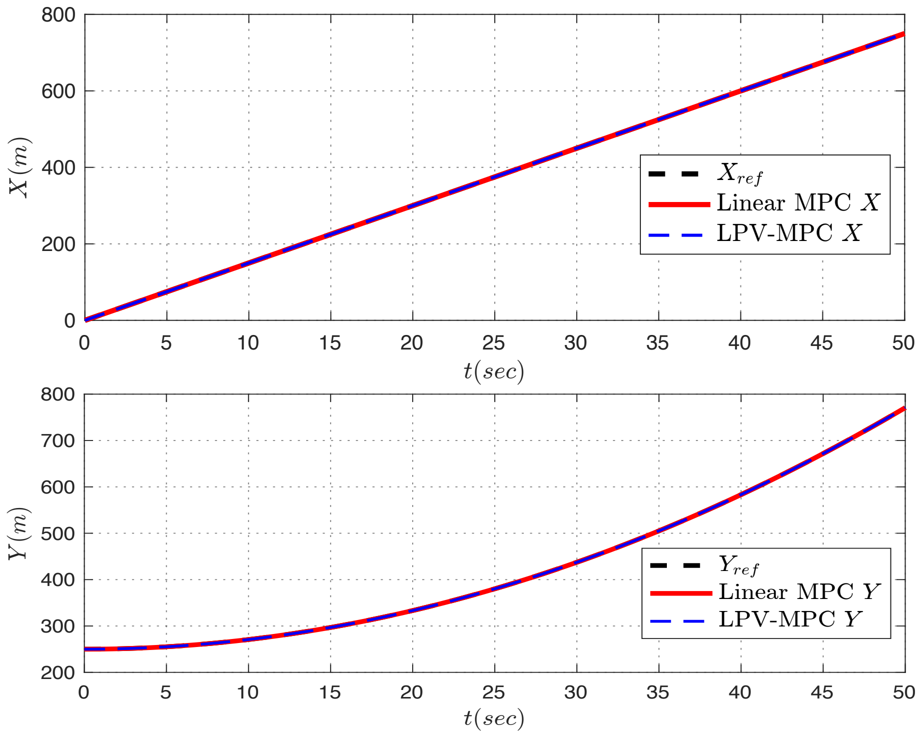

Figure 5.13: a) X position tracking of AV on parabolic reference b) Y position tracking of AV on parabolic reference

In Figure 5.14, the errors in the X and Y positions and yaw angle orientation of the AV are presented for both LPV-MPC and Linear MPC, of the parabolic reference trajectory. The X position error in (a), represents the deviation of the AV's X position from the desired X position along the parabolic reference trajectory. It can be observed that LPV-MPC exhibits smaller errors compared to linear MPC, indicating better accuracy in tracking the X position. LPV-MPC demonstrates a closer alignment with the desired X position, resulting in reduced deviations and superior performance in terms of X position tracking. The Y position error in (b), represents the deviation of the AV's Y position from the desired Y position perpendicular to the parabolic reference trajectory. Both LPV-MPC and linear MPC exhibit relatively small deviations from reference Y position, with comparable errors. In (c), the yaw angle error of LPV-MPC demonstrates slightly smaller errors compared to Linear MPC, which indicates that LPV-MPC achieves better accuracy in controlling the AV's yaw angle orientation along the parabolic reference trajectory.

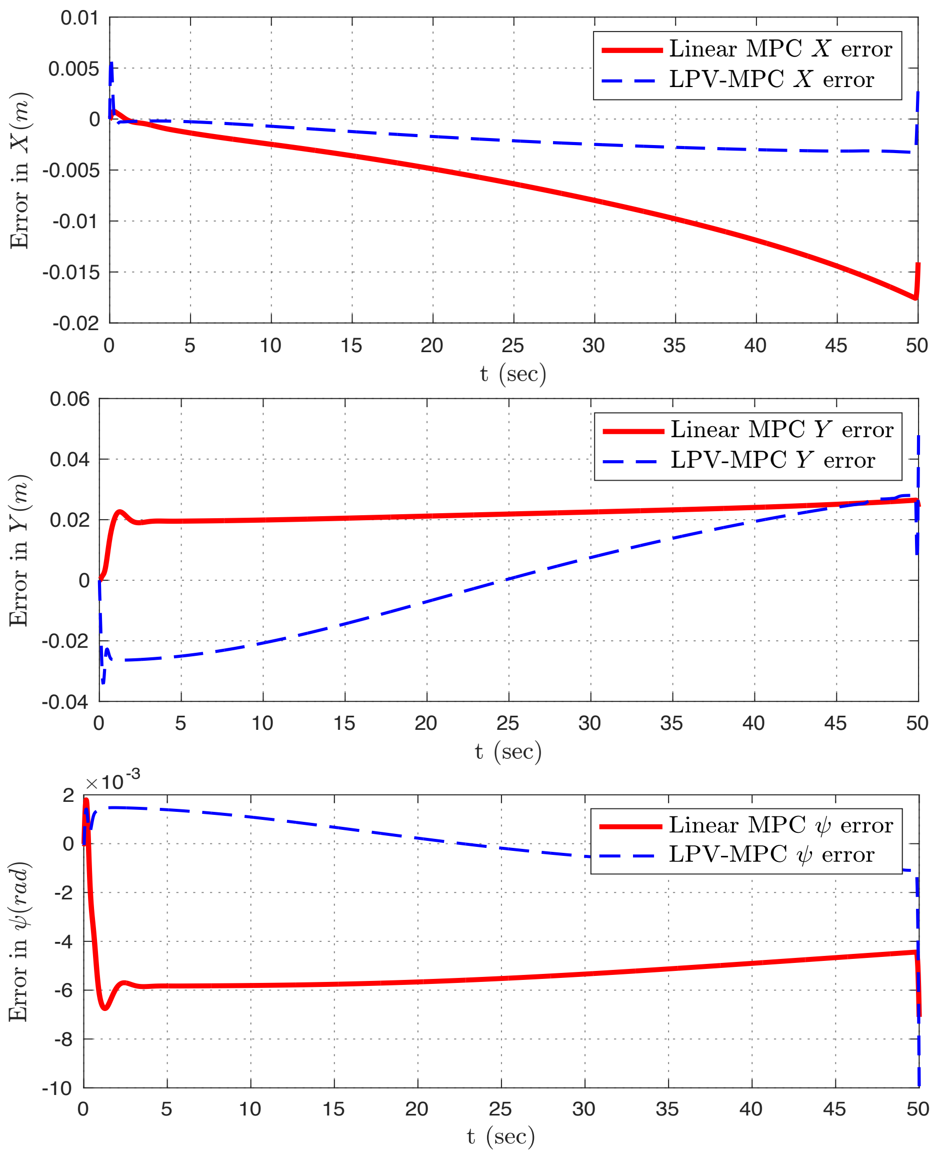

Figure 5.14: a) X position error on parabolic reference b) Y position error on parabolic reference c) ψ angle error on parabolic reference

5.2.3 Trajectory Tracking of Nonlinear Reference (U-Shape Reference)

In this section, the trajectory tracking performance of an AV controlled by LPV-MPC is assessed using a U-shaped reference trajectory. The performance of LPV-MPC is compared to that of Linear MPC to evaluate the effectiveness of the LPV-MPC approach in accurately following complex and curved paths, of the U-shaped reference road in (Alcala et al., 2018).

The primary objective of this analysis is to evaluate the AV's precise navigation capabilities in challenging real-world scenarios involving curved roads, intersections and complex maneuvers. Using LPV-MPC controlled AV along with comparison to Linear MPC in such environments. The aim is to ensure safe and accurate navigation by effectively tracking the desired path, which is given as:

$$
\begin{aligned}
R_{ref_1} &= \begin{cases} X_1 = 15 * t \\ Y_1 = 50 * \sin(2\pi t) + 250 \end{cases} \\
R_{ref_2} &= \begin{cases} X_2 = 300 * \cos(2\pi t) + 600 \\ Y_2 = 300 * \cos(2\pi t) + 500 \end{cases} \\
R_{ref_3} &= \begin{cases} X_3 = 600 - 15 * t \\ Y_3 = 50 * \cos(2\pi t) + 750 \end{cases}
\end{aligned}
\tag{5.3}
$$

The simulation time is 140sec, and the prediction horizon is 10.

To enhance control and improve trajectory following, the PSO algorithm is utilized to optimize the weighting matrices. This optimization process helps fine-tune the control parameters, enabling the AV to achieve more accurate and responsive trajectory tracking. The resulting optimized values of the weighting matrices, obtained through the PSO algorithm, are as follows:

$Q = diag([46.695 \quad 5.496 \quad .176 \quad 0.045 \quad 61.739 \quad 38.528])$ and

$R = diag([.877 \quad 1.918])$

Figure 5.15 shows the comparison between LPV-MPC and Linear MPC in terms of trajectory tracking on a U-shaped reference trajectory. The results demonstrate the superior performance and accuracy of the LPV-MPC controlled AVs in navigating complex and curved paths.

The efficiency of LPV-MPC is attributed to its ability to handle the high nonlinearities present in the AV system. By using a time-varying model, LPV-MPC adapts and responds to the complexities of the U-shaped trajectory, resulting in minimal deviation from the reference path. The LPV-MPC controller effectively captures the dynamic variations of the AV, allowing for precise and reliable control throughout the maneuver. Which emphasizes its potential in enabling AVs to navigate challenging and nonlinear driving scenarios with enhanced accuracy and performance. In contrast, the Linear MPC controller exhibits larger deviations in turning maneuvers from the reference trajectory, indicating its limitations in accurately handling the nonlinearities of the AV system. The linearization process in Linear

MPC did not adequately capture the varying dynamics and uncertainties, leading to less precise trajectory tracking.

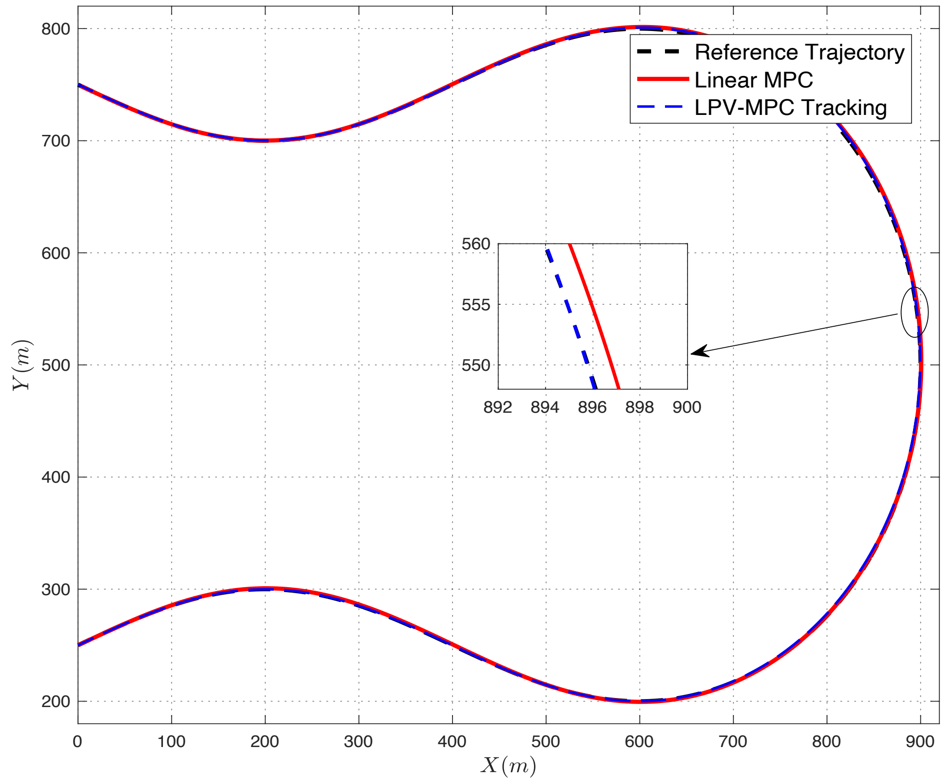

Figure 5.15: Comparison of U-shape Trajectory Tracking: LPV-MPC vs Linear MPC

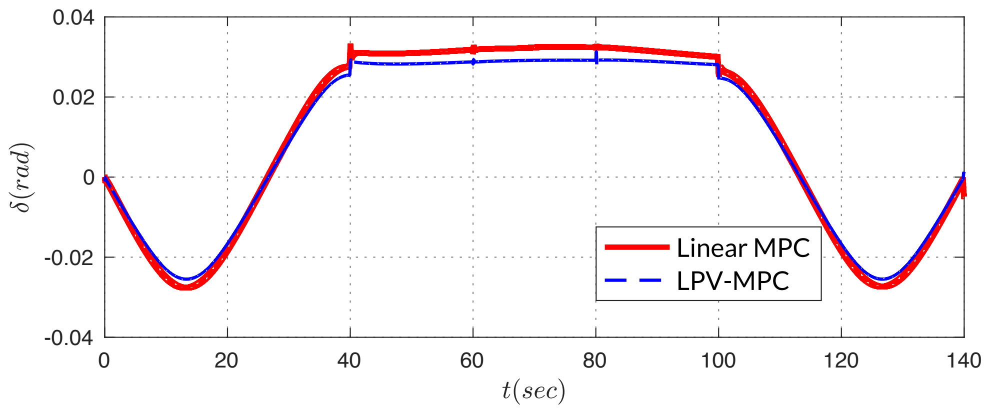

Figure 5.16: Steering angle (δ) control inputs for U-shaped reference

Figure 5.16 depicts the comparison between the control inputs of LPV-MPC and Linear MPC for the steering angle (δ) of the AV during the navigation of a U-shaped trajectory. From the plot, it can be observed that the control input, δ, for LPV-MPC is slightly lower than that of Linear MPC. This difference suggests that LPV-MPC adopts a more refined and precise control strategy in steering the AV along the U-shaped trajectory. The lower control input in LPV-MPC indicates a more efficient utilization of steering commands to achieve accurate trajectory tracking. In contrast, Linear MPC exhibits slightly higher δ control input, indicating a less refined control strategy when navigating the U-shaped trajectory.

In Figure 5.17, the control input for the acceleration (a) of the AV, controlled by both LPV-MPC and Linear MPC, is shown as it traverses a U-shaped reference trajectory. The plot illustrates how the acceleration is dynamically adjusted over time to ensure accurate tracking of the desired U-shaped path. From the figure, it can be observed that LPV-MPC exhibits slightly lower acceleration values compared to Linear MPC. This observation indicates that LPV-MPC provides a smoother and more controlled response in maintaining the AV's speed along the U-shaped trajectory. The lower acceleration values in LPV-MPC suggest a more gradual adjustment of the AV's speed, resulting in a smoother and more refined trajectory tracking performance.

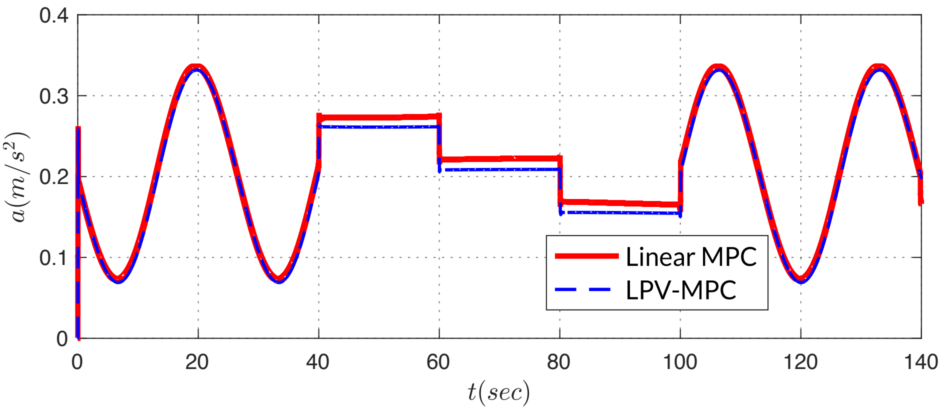

Figure 5.17: Longitudinal acceleration (a) control inputs for U-shaped reference

In Figure 5.18 (a), the comparison between the LPV-MPC and Linear MPC controlled AV is presented, focusing on the AV's longitudinal velocity (V_x) and its alignment with the reference velocity on a U-shaped reference. From the plot, it can be observed that both LPV-MPC and Linear MPC enable the AV to maintain its longitudinal velocity in close proximity to the reference velocity throughout the traversal of the U-shaped reference. Similarly, in Figure 5.18 (b), a comparison is made between LPV-MPC and Linear MPC in terms of the AV's lateral velocity (V_y) and its alignment with the reference velocity on the U-shaped reference. While there may be some deviation observed in both LPV-MPC and Linear MPC, the deviations remain within an acceptable range.

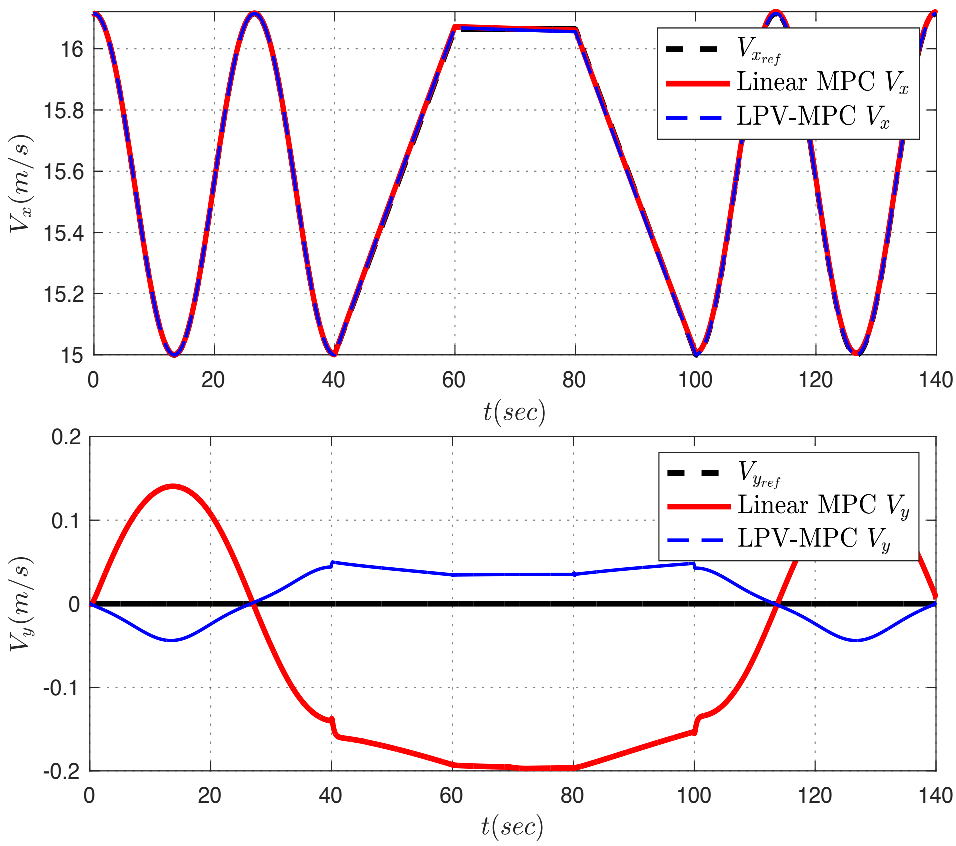

Figure 5.18: a) Longitudinal velocity (V_x) on U-shaped reference b) Lateral velocity (V_y) on U-shaped reference

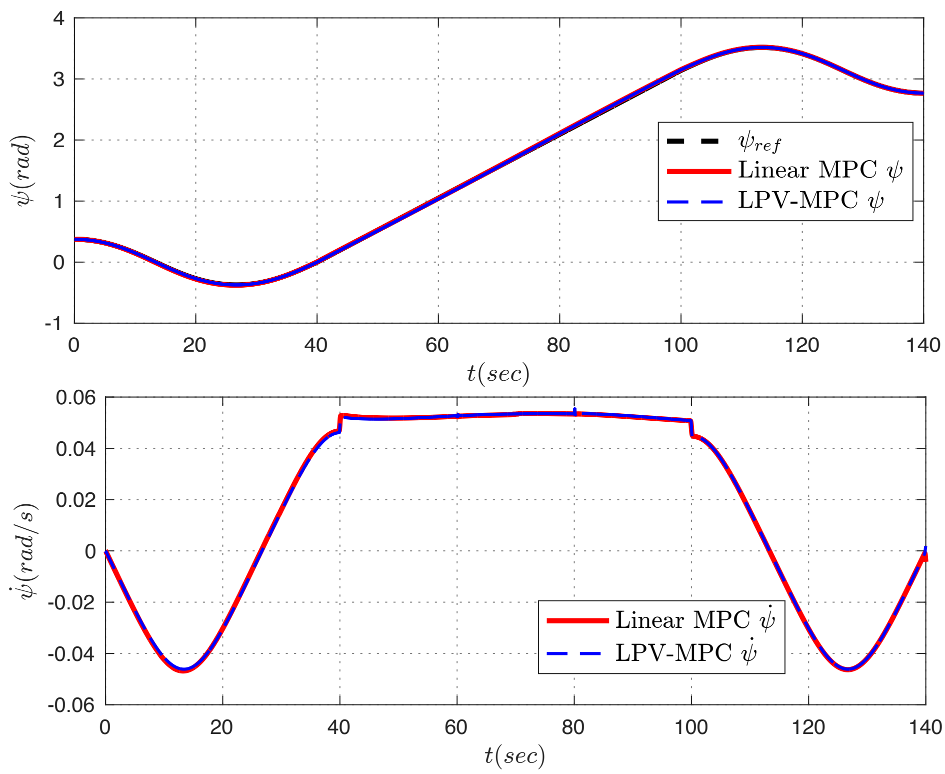

Figure 5.19: a) Yaw angle (ψ) response to trajectory on U-shaped reference b) Yaw angle rate ($\dot{\psi}$) response on U-shaped reference

In Figure 5.19 (a) and (b), the response of the AV's ψ and $\dot{\psi}$ to the U-shaped Road trajectory is shown. It illustrates how both the LPV-MPC and Linear MPC controlled AV adjust their orientation to align with the desired yaw angle trajectory. The graph highlights the AV's ability to accurately follow the intended heading throughout the maneuver.

Figure 5.20 illustrates the X and Y positions tracking of the LPV-MPC controlled AV with comparison of Linear MPC controlled AV as it follows a U-shaped reference trajectory. The graph the evolution of the AV's X and Y positions over time while navigating the highly nonlinear and curved path.

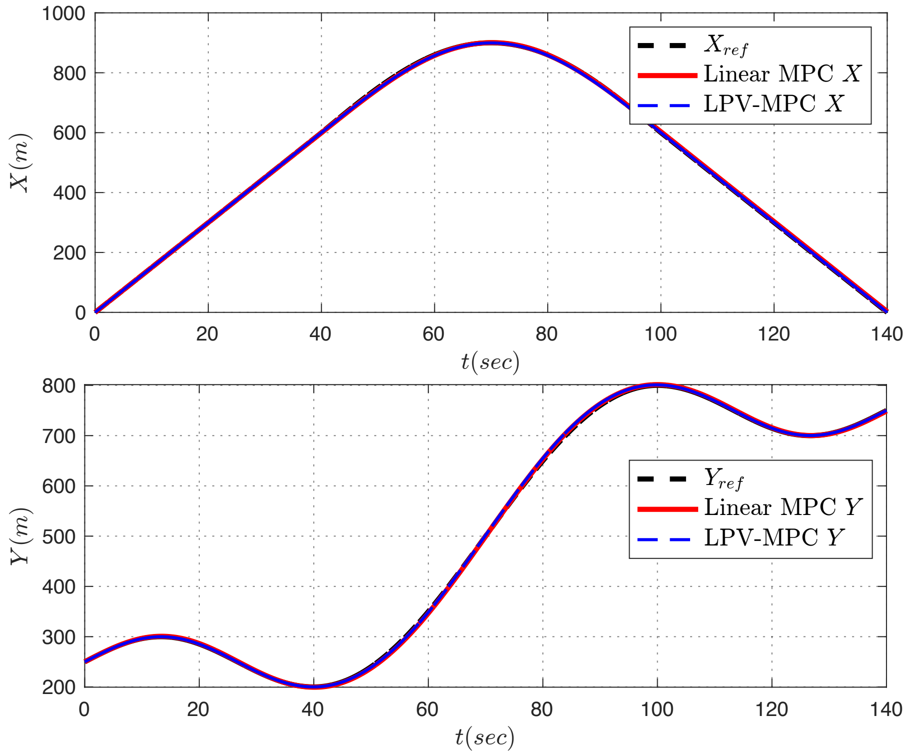

Figure 5.20: a) X position tracking of AV on U-shaped reference b) Y position tracking of AV on U-shaped reference

On the U-shaped reference trajectory, the X and Y for the LPV-MPC and Linear MPC operated AVs are shown in Figure 5.21. The differences between the actual AV positions and the desired reference positions along the trajectory are represented by these inaccuracies. As can be seen from the figure, the LPV-MPC controlled AV exhibits very minor deviations from the reference track, suggesting precise tracking performance. The Linear MPC controlled AV, on the other hand, exhibits a large deviation from the reference track of up to $4.4m$. This distinction underlines the drawbacks of linearizing the AV's dynamic model for control strategies that focus on forecasting and optimizing outcomes and inability to operate under varying conditions. When dealing with highly nonlinear reference trajectories that resemble real road conditions, the benefit of adopting LPV-MPC becomes apparent. The LPV-MPC approach operates well in a variety of circumstances, capturing the AV's nonlinear dynamics and facilitating more precise and dependable trajectory tracking. This outcome highlights how LPV-MPC outperforms Linear MPC in handling complicated and nonlinear road trajectories.

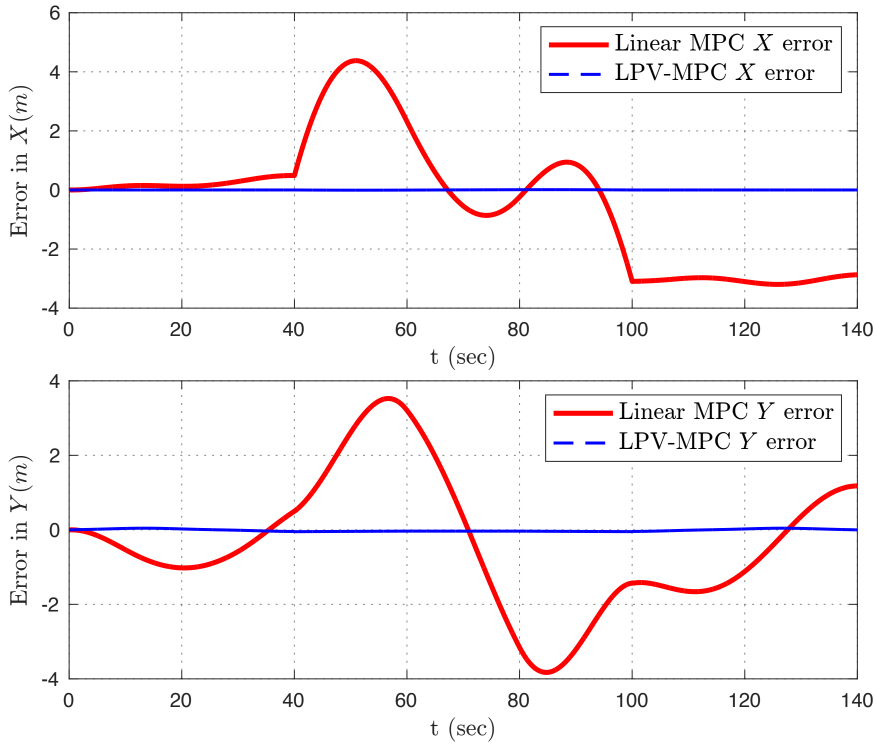

Figure 5.21: a) X position error on U-shaped reference road b) Y position error on U-shaped reference road

CHAPTER SIX

6 CONCLUSION AND RECOMMENDATION

6.1 Conclusion

Achieving accurate trajectory tracking in AVs is a challenging task due to their inherent instability and nonlinear dynamics. A carefully designed control system is essential to ensure precise trajectory tracking. This thesis proposes a MPC method, specifically the LPV-MPC, that is specifically tailored for trajectory tracking in AVs. The LPV framework, which incorporates a time-varying mathematical model capturing the dynamics of the vehicle while considering constraints, serves as the foundation for the MPC method used in this thesis. By generating optimal control inputs, the LPV-MPC control system guides the AV along the predetermined trajectory, enabling accurate and precise trajectory tracking.

The MPC control system is implemented and simulated using the MATLAB software. The weighting matrices of the controller are tuned and optimized using PSO, which results in best control performance. The simulation results of the proposed MPC controller for AV reference trajectory tracking have shown exceptional performance with a minimal tracking error. The controller effectively guides the AV along the desired trajectory, closely following the reference path with high precision. By continuously adjusting the control inputs based on the predicted system behavior, the MPC controller minimizes deviations from the reference trajectory, resulting in a minimal tracking error. Additionally, a Linear MPC approach is also utilized for comparison and evaluation purposes.

Generally, experimental results obtained from assessing the proposed LPV-MPC system provide strong confirmation of its successful performance. The LPV-MPC system surpasses the Linear MPC system, demonstrating exceptional accuracy in tracking the desired trajectory with an impressively low tracking error. Moreover, the LPV-MPC controller exhibits resilience by effectively navigating and tracking nonlinear reference trajectories. This ability is attributed to the system's capability to handle time-varying dynamics. These results serve as compelling evidence of the effectiveness of the proposed control system in achieving accurate trajectory tracking in AVs. The performance of the LPV-MPC system indicates its potential to significantly advance AV technology and enable safe and effective autonomous driving in various real-world scenarios.

6.2 Recommendation

The following recommendations for additional development and extension of the thesis work could be made based on the results and limitations noted in this study:

- ✓ While the proposed MPC system has shown promising results, implementation of NMPC for trajectory tracking of AVs offers the advantage of accurately controlling the complex nonlinear dynamics inherent in AV systems. By formulating the control problem as an optimization task over a finite time horizon, NMPC can generate optimal control inputs that minimize tracking errors less than the LPV-MPC and satisfy system constraints. Implementing NMPC in the trajectory tracking system can enhance its performance, robustness, and capability to handle challenging scenarios with varying dynamics.
- ✓ By incorporating the effects of inclined surfaces and nonlinear tire model, in the model, the control system would be better equipped to handle variations in road conditions, leading to more reliable trajectory tracking.
- ✓ By integrating obstacle detection and avoidance mechanisms into the control system, the AV would be able to perceive and react to obstacles in real-time, ensuring safe trajectory tracking while maintaining efficient navigation.
- ✓ The incorporation of path planning algorithms into the AV system allows it to autonomously determine its trajectory while integrating obstacle avoidance capabilities.
- ✓ While simulation-based evaluations provide valuable insights, validating the developed control system on physical hardware would further enhance the credibility and applicability of the research.

Future research can expand on the results of this thesis by taking into consideration these recommendations, advancing the control systems for AVs, addressing major problems, and advancing the implementation of safe and effective autonomous driving.

REFERENCES

Alcalá, E., Puig, V., & Quevedo, J. (2019). LPV-MPC control for autonomous vehicles. *IFAC-PapersOnLine*, *52*(28), 106–113.

Alcala, E., Puig, V., Quevedo, J., Escobet, T., & Comasolivas, R. (2018). Autonomous vehicle control using a kinematic Lyapunov-based technique with LQR-LMI tuning. *Control Engineering Practice*, *73*, 1–12.

Alcalá, E., Puig, V., Quevedo, J., & Rosolia, U. (2020). Autonomous racing using Linear Parameter Varying-Model Predictive Control (LPV-MPC). *Control Engineering Practice*, *95*, 104270. https://doi.org/10.1016/j.conengprac.2019.104270

Babu, M., Theerthala, R. R., Singh, A. K., Baladhurgesh, B. P., Gopalakrishnan, B., Krishna, K. M., & Medasani, S. (2019). Model Predictive Control for Autonomous Driving considering Actuator Dynamics. *2019 American Control Conference (ACC)*, 1983–1989. https://doi.org/10.23919/ACC.2019.8814940

Besselmann, T., & Morari, M. (2009). Autonomous Vehicle Steering Using Explicit LPV-MPC. *ECC' 09 : European Control Conference, August 23 - 26, 2009, Budapest, Hungary : Conference Proceedings*, 2628–2633. https://www.research-collection.ethz.ch:443/handle/20.500.11850/16721

Boumaza, H., Mustapha, H., & Belarbi, K. (2018, June). *An online solution for Model Predictive Control using Particle Swarm Optimization for autonomous vehicle driving*.

Boyali, A., Mita, S., & John, V. (2018). A Tutorial On Autonomous Vehicle Steering Controller Design, Simulation and Implementation. *CoRR*, *abs/1803.03758*.

Brunton, S. L., & Kutz, J. N. (2022). *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press.

Camacho, E. F., & Bordons, C. (2007). *Model Predictive control*. Springer London. https://doi.org/10.1007/978-0-85729-398-5

Chen, S., Chen, H., & Negrut, D. (2020). Implementation of MPC-Based Path Tracking for Autonomous Vehicles Considering Three Vehicle Dynamics Models with Different Fidelities. *Automotive Innovation*, *3*(4), 386–399. https://doi.org/10.1007/s42154-020-00118-w

Cisneros, P. S. G., Voss, S., & Werner, H. (2016). Efficient Nonlinear Model Predictive Control via quasi-LPV representation. *2016 IEEE 55th Conference on Decision and Control (CDC)*, 3216–3221. https://doi.org/10.1109/CDC.2016.7798752

Faulwasser, T., & Findeisen, R. (2011). A model predictive control approach to trajectory tracking problems via time-varying level sets of Lyapunov functions. *IEEE Conference on Decision and Control and European Control Conference*, 3381–3386. https://doi.org/10.1109/CDC.2011.6160492

Greenblatt, N. A. (2016). Self-driving cars and the law. *IEEE Spectrum*, *53*(2), 46–51. https://doi.org/10.1109/MSPEC.2016.7419800

Guo, N., Zhang, X., Zou, Y., Lenzo, B., & Zhang, T. (2020). A Computationally Efficient Path-Following Control Strategy of Autonomous Electric Vehicles With Yaw Motion Stabilization. *IEEE Transactions on Transportation Electrification*, *6*(2), 728–739. https://doi.org/10.1109/TTE.2020.2993862

Han, B., & Gao, H. (2022). Linear Parameter-Varying Model Predictive Control for Hydraulic Wind Turbine. *Actuators*, *11*(10), 292. https://doi.org/10.3390/act11100292

Hu, C., Qin, Y., Cao, H., Song, X., Jiang, K., Rath, J. J., & Wei, C. (2019). Lane keeping of autonomous vehicles based on differential steering with adaptive multivariable super-twisting control. *Mechanical Systems and Signal Processing*, *125*, 330–346. https://doi.org/10.1016/j.ymssp.2018.09.011

Hu, C., Wang, Z., Taghavifar, H., Na, J., Qin, Y., Guo, J., & Wei, C. (2019). MME-EKF-based path-tracking control of autonomous vehicles considering input saturation. *IEEE Transactions on Vehicular Technology*, *68*(6), 5246–5259.

Jeong, Y., & Yim, S. (2021). Model Predictive Control-Based Integrated Path Tracking and Velocity Control for Autonomous Vehicle with Four-Wheel Independent Steering and Driving. *Electronics*, *10*(22), 2812.

K. Nonami, M. Kartidjo, & Kwang-Joon Yoon. (2013). *Autonomous Control Systems and Vehicles* (K. Nonami, M. Kartidjo, K.-J. Yoon, & A. Budiyono, Eds.; Vol. 65). Springer Japan. https://doi.org/10.1007/978-4-431-54276-6

Kissai, M., Monsuez, B., Tapus, A., & Martinez, D. (2017). A new linear tire model with varying parameters. *2017 2nd IEEE International Conference on Intelligent Transportation Engineering (ICITE)*, 108–115. https://doi.org/10.1109/ICITE.2017.8056891

Lin, F., Chen, Y., Zhao, Y., & Wang, S. (2019). Path tracking of autonomous vehicle based on adaptive model predictive control. *International Journal of Advanced Robotic Systems*, *16*(5), 1729881419880089.

Liu, S. (2020). *Engineering autonomous vehicles and robots: the DragonFly modular-based approach*. John Wiley & Sons.

Liu, S., Li, L., Tang, J., Wu, S., & Gaudiot, J.-L. (2020). Creating autonomous vehicle systems. *Synthesis Lectures on Computer Science*, *8*(2), i–216.

Liuping, W. (2009). *Model Predictive Control System Design and Implementation Using MATLAB*®. Springer London. https://doi.org/10.1007/978-1-84882-331-0

Morato, M. M., Normey-Rico, J. E., & Sename, O. (2020). Model predictive control design for linear parameter varying systems: A survey. *Annual Reviews in Control*, *49*, 64–80. https://doi.org/10.1016/j.arcontrol.2020.04.016

Nouwens, S. A. N., Jager, B. de, Paulides, M., & Heemels, W. P. M. H. (2021). Constraint-adaptive MPC for large-scale systems: Satisfying state constraints without imposing them. *IFAC-PapersOnLine*, *54*(6), 232–237. https://doi.org/10.1016/j.ifacol.2021.08.550

Ozguner, U., Acarman, T., & Redmill, K. A. (2011). *Autonomous ground vehicles*. Artech House.

Pang, H., Liu, N., Hu, C., & Xu, Z. (2022). A practical trajectory tracking control of autonomous vehicles using linear time-varying MPC method. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, *236*(4), 709–723. https://doi.org/10.1177/09544070211022904

Rajamani, R. (2012). *Vehicle Dynamics and Control*. Springer US. https://doi.org/10.1007/978-1-4614-1433-9

Raković, S. V., & Levine, W. S. (2019). *Handbook of Model Predictive Control* (S. V. Raković & W. S. Levine, Eds.). Springer International Publishing. https://doi.org/10.1007/978-3-319-77489-3

Ranjan, S., & Senthamilarasu, S. (2020). *Applied Deep Learning and Computer Vision for Self-Driving Cars: Build autonomous vehicles using deep neural networks and behavior-cloning techniques*. Packt Publishing Ltd.

Reda, A., Bouzid, A., & Vásárhelyi, J. (2020). Model Predictive Control for Automated Vehicle Steering. *Acta Polytechnica Hungarica*, *17*(7), 163–182. https://doi.org/10.12700/APH.17.7.2020.7.9

Rossiter, J. A. (2018). *A first course in predictive control*. CRC press.

Samak, C. V., Samak, T. V., & Kandhasamy, S. (2020). *Control Strategies for Autonomous Vehicles*.

Schramm, D., Hiller, M., & Bardini, R. (2014). *Vehicle Dynamics*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-36045-2

Tang, L., Yan, F., Zou, B., Wang, K., & Lv, C. (2020). An Improved Kinematic Model Predictive Control for High-Speed Path Tracking of Autonomous Vehicles. *IEEE Access*, *8*, 51400–51413. https://doi.org/10.1109/ACCESS.2020.2980188

Vu, T. M., Moezzi, R., Cyrus, J., & Hlava, J. (2021). Model Predictive Control for Autonomous Driving Vehicles. *Electronics*, *10*(21), 2593.

Wang, H., Liu, B., Ping, X., & An, Q. (2019). Path Tracking Control for Autonomous Vehicles Based on an Improved MPC. *IEEE Access*, *7*, 161064–161073. https://doi.org/10.1109/ACCESS.2019.2944894

Wang, M., Chen, J., Yang, H., Wu, X., & Ye, L. (2022). Path Tracking Method Based on Model Predictive Control and Genetic Algorithm for Autonomous Vehicle. *Mathematical Problems in Engineering*, *2022*, 4661401. https://doi.org/10.1155/2022/4661401

Wang, Y., Gao, S., Wang, Y., Wang, P., Zhou, Y., & Xu, Y. (2021). Robust trajectory tracking control for autonomous vehicle subject to velocity-varying and uncertain lateral disturbance. *Archives of Transport*, *57*.

Xia, H., Chen, J., Lan, F., & Liu, Z. (2020). Motion Control of Autonomous Vehicles with Guaranteed Prescribed Performance. *International Journal of Control, Automation and Systems*, *18*(6), 1510–1517. https://doi.org/10.1007/s12555-019-0442-5

Zhu, G., Jie, H., & Hong, W. (2023). Nonlinear Model Predictive Path Tracking Control for Autonomous Vehicles Based on Orthogonal Collocation Method. *International Journal of Control, Automation and Systems*, *21*(1), 257–270. https://doi.org/10.1007/s12555-021-0812-7

APPENDIX

Appendix A: MATLAB Functions

Appendix A1: AV Model MATLAB function

```
function xdot = AV_Model(t, x, u)
  % Model parameters
  g = 9.81;              % Gravitational acceleration
  mu = 0.02;             % Coefficient of friction
  m = 1575;              % Vehicle mass
  Iz = 2875;             % Yaw moment of inertia
  Cf = 19000;            % Front tire cornering stiffness
  Cr = 33000;            % Rear tire cornering stiffness
  lf = 1.2;              % Distance between CG and front axle
  lr = 1.4;              % Distance between CG and rear axle

  % Compute matrix elements
  A11 = (-mu*g/x(1));
  A12 = Cf*sin(u(1))/(m*x(1));
  A14 = (Cf*lf*sin(u(1))/(m*x(1)) + x(2));
  A22 = -(Cr + Cf*cos(u(1)))/(m*x(1));
  A24 = ((Cr*lr - Cf*lf*cos(u(1)))/(m*x(1)) - x(1));
  A34 = 1;
  A42 = (Cr*lr - Cf*lf*cos(u(1)))/(Iz*x(1));
  A44 = -(Cf*lf^2*cos(u(1)) + Cr*lr^2)/(Iz*x(1));
  A51 = cos(x(3));
  A52 = -sin(x(3));
  A61 = sin(x(3));
  A62 = cos(x(3));

  B11 = -Cf*sin(u(1))/m;
  B12 = 1;
  B21 = Cf*cos(u(1))/m;
  B41 = Cf*lf*cos(u(1))/Iz;

  % Initialize the derivative of the state vector
  xdot = zeros(6, 1);

  % Compute the derivative of each state variable
  xdot(1) = A11*x(1) + A12*x(2) + A14*x(4) + B11*u(1) + B12*u(2);
  xdot(2) = A22*x(2) + A24*x(4) + B21*u(1);
  xdot(3) = A34*x(4);
  xdot(4) = A42*x(2) + A44*x(4) + B41*u(1);
  xdot(5) = A51*x(1) + A52*x(2);
  xdot(6) = A61*x(1) + A62*x(2);
end
```

Appendix A2: MPC Optimization MATLAB function

```
function u_opt = MPC_Optimization(A_aug, B_aug, x0, Q, R, N, x_ref)
  n_states = size(A_aug, 1);
  n_inputs = size(B_aug, 2);

  % Initialize the optimization variables
  x_opt = zeros(n_states, N+1);
  u_opt = zeros(n_inputs, N);

  % Set the initial condition
  x_opt(:, 1) = x0;

  % MPC Loop
  for k = 1:N
    % Define the optimization problem
    x_current = x_opt(:, k);

    % Define the constraints
    lb = -inf(n_inputs, 1);  % Lower bounds on inputs
    ub = inf(n_inputs, 1);   % Upper bounds on inputs

    % Solve the quadratic program
    options = optimoptions('quadprog', 'Display', 'off');
    u_k = quadprog(Q,zeros(n_inputs,1),[],[],[],[],lb,ub,[],options);

    % Store the optimal input
    u_opt(:, k) = u_k;

    % Apply the control input to the system
    x_next = AV_Model(A_aug, B_aug, x_current, u_k);

    % Store the next state
    x_opt(:, k+1) = x_next;
  end
end
```

Appendix A3: Cost Function MATLAB function

```
function cost = Cost_Function(Q, R, x, u, x_ref)
  % Calculate the tracking error
  error = x - x_ref;

  % Calculate the cost
  cost = sum(sum(error.^2 .* Q)) + sum(sum(u.^2 .* R));
end
```

Appendix B: MATLAB Codes

Appendix B1: LPV-MPC MATLAB loop

```
N = 10;   % Horizon length
Q = diag([32.74 6.5e-2 .863 4.173e-3 48.736 26.912]);%State cost matrix
R = diag([.297 .846]);   % Input cost matrix
for k = 1:length(t)
  % Update the reference trajectory
  x_ref_k = x_ref(:, k:k+N);

  % Run the MPC optimization
  u_k = MPC_Optimization(A_aug, B_aug, x(:, k), Q, R, N, x_ref_k);

  % Apply the first control input
  u(:, k) = u_k(:, 1);

  % Simulate the system
  x(:, k+1) = AV_Model(A_aug, B_aug, x(:, k), u(:, k));
end
```

Appendix B2: MATLAB Program for LPV-MPC Optimization

```
rng default
options = optimoptions('particleswarm');
options.PlotFcn = 'pswplotbestf';
options.SwarmSize = 30;
options.MaxIterations = 50;

nvar = 8;
LB = [0 0 0 0 0 0 0 0];
UB = [50 10 10 10 100 1 10];
[x, fval] = particleswarm(@evalObj, nvar, LB, UB, options)

function J = evalObj(para)

  Q = diag([para(1) para(2) para(3) para(4) para(5) para(6)]);
  R = diag([para(7) para(8)]);

  yreftot = AVReferenceTrajectory(time)';
  e1 = abs(xHistory(:,1)-yreftot(:,1));
  e2 = abs(xHistory(:,2)-yreftot(:,2));
  e3 = abs(xHistory(:,3)-yreftot(:,3));
  e4 = abs(xHistory(:,4)-yreftot(:,4));
  e5 = abs(xHistory(:,5)-yreftot(:,5));
  e6 = abs(xHistory(:,6)-yreftot(:,6));
  e=[e1;e2;e3;e4;e5;e6];
  J = 0.5*h*(e(1)+2*sum(e(2:end-1))+e(end));
end
```