Optimal Trajectory Planning and Control for Automated Truck and Trailer Systems

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr.Ing.)

von der KIT-Fakultät für Maschinenbau
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

M.Sc.

Mohit Kumar Garg

aus Sonipat, Indien

Tag der mündlichen Prüfung:	13.02.2025
Erster Gutachter:	Prof. Dr.-Ing. Christoph Stiller
Zweiter Gutachter:	Prof. Dr.-Ing. Matthias Althoff

Abstract

Truck-trailer systems, the backbone of long-haul freight transport, are currently grappling with significant challenges. These include safety risks, driver shortages, and environmental concerns. This thesis is a response to these pressing issues, offering optimal trajectory planning, optimization, and control strategies for fully automated truck-trailer systems. The focus is on trajectory planning and stabilization, particularly for one-trailer configurations, with an extension to n-trailer systems.

This research aims to enhance truck-trailer systems' stability, performance, and safety through innovative approaches. The system formulation leverages partial differential flatness to handle the inherent instability of truck-trailer systems. This new system modelling ensures stable motion, expanding the scope of model-based optimal trajectory planning and control.

Given the challenges in the real-time application of model-based optimal control for nonlinear systems, this work introduces the Nominally Guided MPC (NGMPC) framework. Based on this framework, three optimal trajectory control algorithms are developed: Model Predictive Control (MPC), Virtual Model Predictive Control (VMPC), and Flatness-Based Model Predictive Control (FMPC). VMPC utilizes a virtual system model with a virtual input, and FMPC utilizes a system model derived using partial flatness. These algorithms were tested and evaluated on actual prototypes.

Finding optimal and collision-free trajectories for truck-trailer systems is another significant challenge. Existing methods often rely on sampling-based path planning and model-based optimal trajectory stabilization

and handle instability by defining system dynamics around a nominal trajectory or using symmetry for reverse driving. This work utilizes the same numerically stable model used for Flatness-Based Model Predictive Control (FMPC) and extends the application to trajectory planning and optimization. This work formulates the trajectory planning problem as nonlinear programming subject to collision and system constraints.

Finally, this work presents a trajectory optimization algorithm that generates feasible and drive-able G^4 continuous trajectories near the planned reference. The optimization, coupled with sample-based planners, is computationally suitable for real-time applications. Additionally, the optimization generates sparse trajectories, simplifying the planning and storing of offline trajectories for logistics hubs or terminals.

This research significantly advances the state-of-the-art autonomous truck-trailer systems, providing robust trajectory planning and control approaches.

Kurzfassung

Lkw-Anhänger-Systeme, das Rückgrat des Güterfernverkehrs, stehen derzeit vor erheblichen Herausforderungen. Dazu gehören Sicherheitsrisiken, Fahrermangel und Umweltprobleme. Diese Dissertation reagiert auf diese drängenden Probleme und bietet optimale Strategien zur Trajektorienplanung, -optimierung und -steuerung für vollautomatisierte Lkw-Anhänger-Systeme. Der Schwerpunkt liegt auf der Trajektorienverfolgung und Stabilisierung, insbesondere für Konfigurationen mit einem Anhänger, mit einer Erweiterung auf n-Anhänger-Systeme.

Diese Forschung zielt darauf ab, die Stabilität, Leistung und Sicherheit von Lkw-Anhänger-Systemen durch innovative Ansätze zu verbessern. Die Systemformulierung nutzt partielle differentielle Flachheit, um die inhärente Instabilität von Lkw-Anhänger-Systemen zu bewältigen. Diese neue Systemmodellierung gewährleistet eine stabile Bewegung und erweitert den Anwendungsbereich der modellbasierten optimalen Trajektorienplanung und -steuerung.

Angesichts der Herausforderungen bei der Echtzeitanwendung von modellbasierten optimalen Steuerungen für nichtlineare Systeme wird in dieser Arbeit das Nominally Guided MPC (NGMPC)-Rahmenwerk eingeführt. Basierend auf diesem Rahmenwerk werden drei optimale Trajektoriensteuerungsalgorithmen entwickelt: Model Predictive Control (MPC), Virtual Model Predictive Control (VMPC) und Flatness-Based Model Predictive Control (FMPC). VMPC nutzt ein virtuelles Systemmodell mit einer virtuellen Eingabe, und FMPC verwendet ein Systemmodell, das mithilfe von partieller Flachheit abgeleitet wurde. Diese Algorithmen wurden an realen Prototypen getestet und evaluiert.

Die Suche nach optimalen und kollisionsfreien Trajektorien für Lkw-Anhänger-Systeme stellt eine weitere große Herausforderung dar. Bestehende Methoden verlassen sich oft auf stichprobenbasierte Pfadplanung und modellbasierte optimale Trajektorienstabilisierung und bewältigen Instabilität, indem sie die Systemdynamik um eine nominale Trajektorie definieren oder Symmetrie für das Rückwärtsfahren nutzen. Diese Arbeit verwendet dasselbe numerisch stabile Modell, das auch für FMPC verwendet wird, und erweitert die Anwendung auf Trajektorienplanung und -optimierung. Das Problem der Trajektorienplanung wird in dieser Arbeit als nichtlineare Programmierung formuliert, die Kollisionen und Systembeschränkungen unterliegt.

Schließlich wird in dieser Arbeit ein Trajektorienoptimierungsalgorithmus vorgestellt, der machbare und fahrbare G^4 kontinuierliche Trajektorien in der Nähe der geplanten Referenz erzeugt. Die Optimierung, gekoppelt mit stichprobenbasierten Planern, ist rechnerisch für Echtzeitanwendungen geeignet. Darüber hinaus erzeugt die Optimierung spärliche Trajektorien, was die Planung und Speicherung von Offline-Trajektorien für Logistikhubs oder Terminals vereinfacht.

Diese Forschung stellt einen bedeutenden Fortschritt für autonome Lkw-Anhänger-Systeme dar und bietet robuste Lösungen für die Trajektorienplanung und -steuerung.

Acknowledgements

My journey into the field of automotive engineering began during my undergraduate studies in India. Participating in competitions such as SAE BAJA India and Supra India fueled my passion for this discipline. Additionally, my involvement with the robotics club piqued my interest in robotics, ultimately leading me to pursue a master's degree in Automotive Engineering in Germany.

I am grateful to my uncle, who is also one of my professors. His guidance during my studies was instrumental in shaping my academic and professional trajectory. He motivated me to pursue a master's degree in Germany.

During my Masters, an internship at MAN Truck & Bus significantly influenced my decision to specialize further. This opportunity allowed me to develop control algorithms for the truck and trailer system for automated driving. Under the mentorship of Arne Christoph Hildebrandt and Sven Kraus, I worked to validate these algorithms on actual prototypes. This hands-on experience greatly enhanced my knowledge and deepened my interest in the field. Arne and Sven believed in my potential even when I doubted my ability to complete a PhD.

I am profoundly grateful to Professor Christoph Stiller for his valuable suggestions, feedback, and continuous motivation throughout my PhD journey. His insights were pivotal in guiding my research and improving my work.

During my doctoral studies, I worked closely with Peter Strauss, whose exceptional leadership and support in conceptual brainstorming and regular reviews were invaluable. My colleagues at MAN Truck & Bus, particularly Özkan and Simon, provided crucial assistance in setting up prototypes and conducting experiments. Frowin Koop and his team ensured that the trucks remained operational by supporting continuous testing. Varun and Daniel, for reviewing my work, also need to be mentioned. I would also like to acknowledge Varun and Daniel for their valuable feedback during the review of my work. I sincerely thank Sven Kraus, my industrial supervisor, for his unwavering support and guidance.

I am also grateful to the MAN automation team for their collective efforts and support throughout my research. Associations with FZI and the Institute of Measurement and Control Systems (MRT) at Karlsruhe Institute of Technology (KIT) significantly enriched my research. Sahin, in particular, deserves special mention for his role in organizational activities.

I especially thank Andreas Haas, whose significant contributions during his thesis helped me take a big step forward in my research. I also thank all my supervised students for their hard work and dedication.

I sincerely appreciate the constant support from my wife, Himani Garg, and my family and friends, who stood by me throughout this journey. Their encouragement and understanding were vital to my success.

Contents

Appendix

Notation

This chapter introduces the notation and symbols which are used in this thesis.

General Notation

Scalars	italic Roman and Greek lowercase letters	x, α
Scalar Parameters	Roman and Greek lowercase letters	x
Sets	Greek uppercase letters	Θ
Vectors	bold Roman lowercase letters	$\mathbf{x}$
Matrices	Roman uppercase letters	A
State spaces	bold calligraphic Roman uppercase letters	$\boldsymbol{\mathcal{X}}$
Trajectory	calligraphic Roman lowercase letters with bold italic vector	$\boldsymbol{x}$
Equivalent system matrices for trajectories	calligraphic Roman uppercase letters	$\mathcal{A}$
Scalar valued function	italic Roman uppercase letter	$J()$
Vector function	bold Roman lowercase letter	$f()$
Matrix-valued function	bold Roman uppercase letter	$\mathbf{F}()$

Numbers and indexing

Mathematical Notations

$\mathbb{N}$	natural numbers
$\mathbb{N}_0$	natural numbers including zero (non-negative integers)
$\mathbb{R}$	real numbers
k, t, i, f	discrete points in time
$\mathrm{n}, \mathrm{j}, \mathrm{t}, \mathrm{v}, \mathrm{h}$	indexing for objects, measurements, and points

Geometry (coordinates, vehicle)

x, y, z	world coordinates
v	velocity
a	acceleration
δ	steering angle
θ	hitch angle
G^1 continuity	first-order geometric continuity
G^2 continuity	second-order geometric continuity
G^3 continuity	third-order geometric continuity
G^4 continuity	fourth-order geometric continuity

One-General Trailer Notations

x	trailer's x-coordinate
y	trailer's y-coordinate

ψ	heading angle of trailer
φ	heading angle of truck
δ	steering angle
θ	hitch angle
γ	rate of change of hitch angle
$\mathrm{l_h}$	off-axle hitching distance
$\mathrm{l_v}$	truck's wheelbase
$\mathrm{l_t}$	trailer's length

Acronyms

DGNSS	Differential Global Navigation Satellite System
FMPC	Flatness-Based Model Predictive Control
LC	Lyapunov Control
LQ-MPC	Linear Quadratic Model Predictive Control
LQR	Linear Quadratic Regulator
MPC	Model Predictive Control
NGMPC	Nominally Guided MPC
n-GT	n-General Trailer
NMPC	Non-Linear Model Predictive Control
n-ST	n-Standard Trailer
one-GT	One-General Trailer
one-ST	One-Standard Trailer
PPC-LC	Cascade Control consisting of a High-Level Pure Pursuit Control and a Low-Level Lyapunov Control

QP	Quadratic Programming
RHC	Receding Horizon Control
SC	Sharpness Continuous
SLC	Simiplified Lyapunov Control
TS	Truck-Semitrailer
two-GT	Two-General Trailer
VMPC	Virtual Model Predictive Control

1. Introduction

Logistics and transportation are critical components of modern commerce and industry, facilitating the movement of goods from producers to consumers. The global transportation network comprises various modes, including rail, road, and waterways, each playing a distinct role in the supply chain.

Railroads are the backbone of freight transportation, particularly for long-haul routes. They offer efficient, high-capacity movement of bulk goods, such as coal, minerals, and agricultural products, over land. Rail terminals with extensive infrastructure serve as critical nodes in the logistics network where goods are loaded, unloaded, and transferred to other transport modes.

Waterways encompassing maritime and inland navigation are integral for international trade and the movement of large cargo volumes. Ports, acting as maritime hubs, facilitate the transfer of goods between ships and other transportation modes, such as rail and road. Waterway transportation is exceptionally cost-effective for heavy and non-perishable goods over long distances.

Road transport is a crucial and widely used mode of transport connecting various modes to ensure goods reach their final destinations. It is essential for last-mile delivery and connecting different logistics hubs. Truck-trailer systems are vital for long-haul transportation in the EU and the US, often serving as the most practical solution [Ink20]. They support the distribution of goods to various destinations, including urban centers and remote areas.

A logistics hub, or terminal, is a central point where goods are received, sorted, and dispatched to their next destination. These hubs can be part of a port, a railway terminal, or a dedicated shipping center. They play a pivotal role in ensuring the efficient flow of goods within the supply chain.

Within logistics hubs, trucks navigate complex environments to load and unload goods. This task requires precise maneuvering, adherence to safety protocols, and efficient coordination to minimize delays. Drivers must be skilled in handling tight spaces and high-traffic conditions typical of busy terminals.

Once loaded, trucks transition from the hub to the highway, which involves navigating urban or suburban roads, adhering to traffic regulations, and ensuring the load's stability. The ability to efficiently merge onto highways is crucial for maintaining delivery schedules.

On the highway, trucks travel long distances to reach destination logistics hub. This phase demands sustained attention to driving conditions, fuel management, and compliance with long-haul driving regulations. The focus is maintaining steady speeds, ensuring driver rest breaks, and optimizing route efficiency.

As trucks approach the destination hub, they must exit the highway and navigate local roads again. This phase involves re-entering complex environments similar to the departure hub, requiring careful driving to ensure safe and timely delivery of goods.

Efficient management of driving tasks within and between logistics hubs is crucial for optimizing the supply chain. By understanding the unique challenges and requirements of each driving phase, from terminals to highways, the logistics industry can enhance the reliability and efficiency of freight movement. Such as, introducing driver-less haul trucks in Australian surface mining. Bellamy et al. assessed the impact of these autonomous trucks in surface mining operations, focusing on efficiency, safety, and overall implications of implementing such technology in the mining sector [Bel11].

Environmental considerations also play a significant role in shaping the discourse around truck-trailer systems. The transportation sector is a major contributor to greenhouse gas emissions, with heavy-duty vehicles, including trucks and trailers, accounting for approximately twenty five percent of carbon emissions from road transport in the EU [Gan18]. Improving the efficiency of truck-trailer systems is crucial for reducing their environmental impact and aligning with sustainability goals. Additionally, efforts to enhance the efficiency and sustainability of truck-trailer systems include aerodynamic improvements to reduce energy consumption [Zah23, Sha23] and investigating the impact of automation on long-haul trucking operator-hours [Moh22].

Research has also focused on optimizing urban logistics facility locations to reduce truck-related greenhouse gas emissions [Gan18] and designing regional logistics networks to mitigate congestion caused by truck flow [Gan20]. The importance of truck drivers in sustainable transportation and logistics within the supply chain has also been emphasized [Moc23].

Safety is one of the primary concerns driving research in autonomous driving. These systems are susceptible to accidents such as jackknifing, where the trailer swings out of alignment with the truck, potentially leading to severe crashes [Ink20]. The Federal Motor Carrier Safety Administration (FMCSA) reported over 500,000 truck accidents in the US in 2019, with a notable portion involving articulated vehicles, emphasizing the need for enhanced safety measures [Moh22].

Moreover, the logistics industry is grappling with a shortage of skilled drivers, exacerbated by the demanding nature of long-haul driving and hazardous working conditions. This shortage is particularly acute in the EU, where the International Road Transport Union (IRU) noted that twenty percent of truck driver positions remained unfilled in 2019 [Sha23]. To address these challenges, automation and advanced control systems are being explored as solutions to reduce reliance on human drivers, enhance safety, and improve operational efficiency.

The evolution of the transportation and logistics industry has led to increasing demands for efficiency, safety, and automation. Advanced control strategies for articulated vehicles, such as truck-trailer systems, have garnered significant attention due to their importance in transporting goods over long distances and through intricate urban environments. However, controlling these systems poses unique challenges due to their non-linear dynamics and potential for instability [Gan20].

In conclusion, truck-trailer systems are indispensable for the efficient movement of goods in the logistics sector. Despite their significance, truck-trailer systems encounter challenges requiring advanced trajectory control and planning strategies to ensure safety, efficiency, and reliability. Addressing the challenges faced by these systems through advanced control strategies is crucial to ensuring safety, efficiency, and reliability in transportation operations. By leveraging automation, enhancing safety measures, and improving environmental sustainability, the industry can navigate the complexities of modern logistics and meet the evolving demands of global supply chains.

1.1. Motivation

The evolution of the transportation and logistics industry is increasingly leaning towards automation to enhance efficiency, safety, and reliability. Due to various advancements, the development of autonomous truck-trailer systems has gained significant attention. These systems are crucial for transporting goods long distances, particularly in hub-to-hub logistics operations.

Figure 1.1.: MAN Truck & Trailer Prototype for Hamburg Truck Pilot Project

The Hamburg TruckPilot project, a prime example of pioneering work in this domain, serves as an inspiration. MAN Truck & Bus and Hamburger Hafen und Logistik AG completed this project, which focused on developing and practically testing autonomous trucks in container handling at the HHLA Container Terminal Altenwerder (CTA). During the project, trucks autonomously navigated terminal areas, demonstrating the feasibility and efficiency of self-driving technology in logistics processes. These autonomous trucks drove to their destinations within the terminal and maneuvered themselves with high precision into parking positions, highlighting the potential of autonomous systems in complex environments.

The primary objective of this work was to deliver a robust trajectory control algorithm for truck-trailer systems, a crucial step for the applications in the Hamburg TruckPilot project. The research focuses on optimal trajectory planning and control within terminal environments, encompassing forward and reverse driving.

Current control methods often need help with the inherent complexities of truck-trailer dynamics, leading to suboptimal performance and safety

concerns. Traditional methods may not effectively address issues such as jackknifing, which can result in severe accidents or the precise maneuvering required in confined terminal spaces. Furthermore, these systems must handle truck-trailer configurations' nonlinear dynamics and potential instability, particularly during reverse maneuvers.

This thesis aims to bridge the gap by developing advanced control and planning strategies to handle truck-trailer systems' dynamics and nonlinear nature.

1.2. Problem Statement

Despite the progress in trajectory planning and control strategies for truck-trailer systems, several key issues still need to be solved. The primary problems addressed in this thesis are:

1. **Stabilization of Hitch Angle:** The hitch angle between the truck and trailer must remain within safe limits to prevent jackknifing and improve maneuverability. Current methods often fail to provide robust performance under varying conditions.

2. **Numerical Stability:** The presence of numerical instability in the mathematical truck-trailer systems can significantly affect the effectiveness of control algorithms. Even a steady-state integration of the model can eventually lead to instability. Therefore, developing a method that can effectively mitigate this instability is crucial for ensuring the reliability of simulations and the successful application of control strategies in real-world scenarios.

3. **Trajectory Planning:** The task of finding feasible trajectories for truck-trailer systems in logistics hubs and terminals is a complex and challenging one. This complexity is primarily due to the large dimensions of the vehicles and the need to avoid collisions. While initial route finding is less of an issue in these structured environments, the planning of trajectories that account for the physical constraints of the truck and

trailer is a significant challenge. Moreover, the need for optimal trajectory planning is further compounded by the instability during backward driving and the requirement to park in narrow spaces.

4. **Trajectory Tracking and Optimization:** Precise trajectory tracking while optimizing the path for efficiency and safety remains a significant challenge. Traditional control methods may not adequately address truck-trailer systems' nonlinear dynamics and constraints.

5. **Real-World Validation:** Theoretical and simulation-based validations are essential. However, demonstrating the practical feasibility and effectiveness of the proposed control strategies through real-world experiments is crucial for their adoption in the industry.

The research presented in this thesis aims to tackle these problems by developing and validating trajectory planning and control strategies that improve truck-trailer systems' stability, performance, and safety.

1.3. Contributions

This thesis presents several significant contributions to truck-trailer systems' control strategies and trajectory planning. The main focus areas include one-trailer systems and extending developed methods to n-trailer systems. The key contributions are as follows:

Derivation of Partial Flat Truck-Trailer System Dynamics: This research provides a detailed analysis and application of partial differential flatness to handle the numerical instability of the general one-trailer system. This work enhanced control and trajectory planning capabilities by utilizing the model derived using partial differential flatness.

Hitch Angle Stabilization Control of One-General Trailer (one-GT): This thesis introduces and implements the Lyapunov-based control law to stabilize the hitch angle in one-trailer systems. Simulation studies demonstrated the effectiveness of developed control laws under

various input conditions, including step, ramp, and sinusoidal inputs, highlighting its robust performance.

Nominally Guided MPC (NGMPC) Framework: The research introduces NGMPC as a practical trajectory tracking approach for real-time optimal control formulation of nonlinear systems. The framework can efficiently formulate a quadratic programming (QP) problem for controlling nonlinear systems, enhancing real-time capabilities while ensuring accuracy.

Optimal Trajectory Stabilization Control for one-GT: Building on the NGMPC, the study developed and analyzed three control methods for truck-trailer systems: Model Predictive Control (MPC), Virtual Model Predictive Control (VMPC), and Flatness-Based Model Predictive Control (FMPC). VMPC incorporates an integrated control law within the prediction model definition to address instability, while FMPC utilizes a flat model as the prediction model.

Optimal Trajectory Planning Algorithm for one-GT: This thesis addresses the problem of finding feasible and collision-free trajectories for truck-trailer systems in structured environments like logistics hubs and terminals. By considering the physical constraints of the vehicles, the study develops trajectory planning algorithms that account for the large dimensions of the trucks and trailers, ensuring safe and efficient navigation. Efficient collision constraints between the truck-trailer and other parked vehicles and obstacles and a method to map logistics hubs and parking lots are proposed to ensure safe and efficient planning.

Trajectory Optimization Ensuring Feasibility and Drivability: This contribution focuses on optimizing paths for both standard and general one-trailer models to ensure they are drivable and efficient. The research involved a comparative analysis of different optimization techniques to identify the most effective approach for the given system, thereby enhancing the practical applicability of the control strategies.

Extension of Partial Flatness to n-Trailer Systems: The thesis extends the developed control methods from one-trailer systems to

n-trailer configurations. A specific case study on a two-trailer system encompassed mathematical modeling, stability analysis, and experimental validation. This extension demonstrates the versatility and scalability of the proposed control strategies.

Experimental Validation of Trajectory Stabilization: Finally, the developed NGMPC based control strategies were validated on a real truck and trailer system, demonstrating their practical feasibility, effectiveness, and real-time capabilities. This real-world validation underscores the potential of the proposed methods for industrial adoption and their contribution to advancing autonomous vehicle technology.

As part of the Hamburg TruckPilot project, the developed motion control strategies are tested and validated on real prototype.

1.4. Outline

The outline of this work is as follows:

Chapter 1 introduces the problem statement and the contributions of this research and provides an outline of the thesis.

Chapter 2 presents the fundamentals and related work. It includes discussions on differential flat systems, stability criteria, optimal control, and related work in feedback control and model predictive control (MPC) for articulated vehicles.

Chapter 3 details the system modeling of the n-trailer system, including the general one-trailer model, and using the differential and partial differential flatness, it conceptualizes the flat model of One-Standard Trailer (one-ST) and One-General Trailer (one-GT) systems.

Chapter 4 focuses on the stabilization of the hitch angle of the one-GT. It discusses Lyapunov Control (LC) and Linear Quadratic Regulator (LQR) for hitch angle stabilization and presents simulation results and discussions, including step, ramp, and sinusoidal input responses.

Chapter 5 introduces Nominally Guided MPC (NGMPC) for truck-trailer trajectory tracking. It covers the linearization of system dynamics, formulation of the objective function, and derivation of quadratic optimization problems for Nominally Guided MPC (NGMPC). This chapter also discusses Virtual Model Predictive Control (VMPC) and Flatness-Based Model Predictive Control (FMPC) for the one-GT system and provides experimental results and validation.

Chapter 6 explores trajectory planning for the truck-trailer system in logistic terminals and parking lots. It presents the results of trajectory planning for various parking scenarios. Then, it explains trajectory optimization to enhance drivability and feasibility validation for the truck-trailer system, including the results.

Chapter 7 generalizes the concept of partial flatness to n-General Trailer (n-GT) systems. Based on this, it extends the Lyapunov Control (LC) and trajectory planning to Two-General Trailer (two-GT). Then, it presents the results and discussion of the LC and trajectory planning for a two-GT system.

Chapter 8 concludes the thesis with a summary of contributions and discusses future outlooks for further research in this area.

2. Fundamentals and Related Work

Optimal trajectory planning and control of articulated vehicles have been subjects of extensive research. The truck trailers have complex dynamics and practical significance in various applications. Researchers explored several approaches to enhance these systems' stability and trajectory control. This chapter provides a comprehensive overview of the fundamental concepts necessary to understand articulated vehicle trajectory optimization and control. The chapter concludes with a review of previous research and studies on the control of articulated vehicles, highlighting key findings and approaches.

2.1. Fundamentals

This section presents the fundamental concepts and principles necessary to understand trajectory optimization and control of articulated vehicles. Basic theoretical concepts of Differential Flat Systems and Partial Differential Flat Systems are introduced. The concepts of a differential flat system, a class of dynamical systems characterized by flatness, and the requirements a system must satisfy to be classified as flat are described. An example of a cart-pole system is discussed to help convey the idea of differential flatness. The concept of a flat output and its usage in describing desired outputs and their derivatives are covered. Partial differential flat systems in which flatness properties are exhibited only on a subset of the state space are introduced.

Next, the section discusses the Stability Criteria and Analysis, explaining the concept of stability in control theory and briefing the stability criteria: Lyapunov Stability, Asymptotic Stability, and Exponential Stability. Lyapunov Control is formulated using the Lyapunov Direct Method, a mathematical technique used to prove the equilibrium stability of nonlinear dynamical systems.

The section then summarizes optimal control methodologies, including the Linear Quadratic Regulator (LQR), Linear Quadratic Model Predictive Control (MPC), Nonlinear Model Predictive Control (NMPC).

Finally, various Optimization Methods are briefly introduced, including norm minimization, quadratic programming (QP), nonlinear programming (NLP), and Differential Dynamic Programming (DDP).

2.1.1. Differential Flat System

Control theory introduces a differential flat system, a specialized class of dynamical systems characterized by a certain property known as flatness [Rou93, Fli95, Mar01, Rou93].

Consider a system with state, $\mathbf{x} \in \mathbb{R}^{\mathrm{n}}$ and control input, $\mathbf{u} \in \mathbb{R}^{\mathrm{m}}$. The dynamics of the system are described by:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)), \tag{2.1}$$

where at $t = 0$, $\mathbf{x}_0 = \mathbf{x}_{\mathrm{init}}$ and $\mathrm{Rank}\left(\frac{\partial \mathbf{f}(\mathbf{x},\mathbf{u})}{\partial \mathbf{u}}\right) = \mathrm{m}$.

The system is flat if there exists an output, $\mathbf{y}(t) = [y_1(t), \ldots, y_m(t)]^{\mathrm{T}}$ which satisfies the following criteria:

- y_{i}, $\mathrm{i} = 0 \ldots m$ should be expressible as functions of state $\mathbf{x}$ and input $\mathbf{u}$ and a finite number of derivatives of the input.
- It should be possible to define states x_{i} and inputs u_{i} as functions of the output y_{i} and its finite number of derivatives.
- The components of $\mathbf{y}(t)$ should be differentially independent.

A characteristic feature of flat systems is a flat output, $\mathbf{z}$, constructed from the system states and inputs. This output and its derivatives can represent the desired outputs and their respective derivatives:

$$\mathbf{z} = h(\mathbf{x}, \mathbf{u}). \tag{2.2}$$

Flat systems can be derived from the method known as flat output parameterization. By systematically manipulating the systems dyanmics, relationships between system input $\mathbf{u}$ and flat output $\mathbf{z}$ can be established. This methodology aids in designing control laws capable of tracking desired system trajectories.

An illustrative example explains the differential flatness. This example, cited from [Fli95], is drawn from the field of control systems and pertains to a crane system depicted in fig. 2.1.

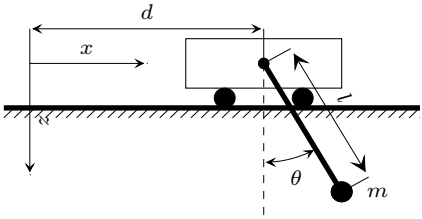


Figure 2.1.: Illustration of a cart-pole system with a pendulum.

To describe the load's behavior, a dynamic model is established through a set of differential and algebraic equations, which can be summarized as follows:

$$m\ddot{x} = -T\sin\theta, \quad \ddot{z} = -T\cos\theta + mg,$$
$$x = l\sin\theta + d, \quad z = l\cos\theta.$$

Here $\mathbf{x} = [x, z, \theta, T]^{\mathrm{T}}$ and $\mathbf{u} = [l, d]^{\mathrm{T}}$, where x and z denote the coordinates of the load, T represents the tension in the rope, θ is the angular displacement, d is the trolley's position and l is the length of the rope.

Consider $\mathbf{z} = [x, z]^{\mathrm{T}}$ as the system flat output, the elements of the state and input as derived in [Fli95] can be expressed as the function of this flat output and its finite derivatives as detailed below:

$$\sin\theta = \frac{x-d}{l}, \quad T = \frac{ml(g-\ddot{z})}{z},$$
$$d = x - \frac{\ddot{x}z}{\ddot{z}-g}, \quad l^2 = z^2 + \left(\frac{\ddot{x}z}{\ddot{z}-g}\right)^2.$$

In summary, differential flatness is widely used in motion control and planning because it simplifies control synthesis and trajectory generation, as flat outputs enable planning and control in the flat output space. The entire system state and input is a function of the flat output and a finite number of its derivatives. The advantage of this approach is most apparent when the system has fewer flat outputs than states and inputs, significantly reducing computational effort by eliminating the need to solve sensitivity differential equations in nonlinear programming (NLP). Some notable applications include:

- **Robotics**: Differential flatness in the control of robotic systems, such as manipulators, quadcopters, and mobile robots [Spo20, Nie98, Mur17], enables complex tasks like trajectory tracking, obstacle avoidance, and motion coordination.
- **Aerospace Systems**: Differential Flatness finds a significant application in aerospace systems, including aircraft and spacecraft [Bry18], aiding in control strategies for various flight phases while satisfying constraints and performance requirements.
- **Autonomous Vehicles**: Differential flatness plays a crucial role in autonomous vehicles, including self-driving cars and UAVs

[Pad16, Hao05]. It facilitates safe and smooth trajectory generation, obstacle avoidance, and cooperative behavior among multiple vehicles, underscoring its importance in this rapidly evolving field.

2.1.2. Partial Differential Flat Systems

Partial differential flat systems extend the concept of differential flatness to systems where only a subset of the state space exhibits flatness properties [Ram14]. This concept is particularly relevant in control systems where complete differential flatness is unattainable, yet a subset of the system's states is flat.

Partial differential flatness is similar to the relative flatness described in [Sil01], where subsystems exhibit differential flatness. Contrary to relative flatness, partial differential flatness allows for calculating all control inputs directly from the flat outputs and their higher-order derivatives. If the system's flat production and the initial condition of its non-flat component at time t_0 are known, the entire system state and control can be computed.

A system described by the state vector $\mathbf{x} \in \mathbb{R}^{\mathrm{n}}$ and control input $\mathbf{u} \in \mathbb{R}^{\mathrm{m}}$ is said to be partially differentially flat if there exists a subset of the state space such that

$$\mathbf{x} = \begin{bmatrix} \mathbf{s} \\ \mathbf{r} \end{bmatrix},$$

with $\mathbf{s} \in \mathbb{R}^{\mathrm{k}}$ and $\mathbf{r} \in \mathbb{R}^{\mathrm{n-k}}$. Here, $\mathbf{s}$ represents the flat states, and $\mathbf{r}$ represents the non-flat states.

The output $\mathbf{y} \in \mathbb{R}^{\mathrm{m}}$ of the form $\mathbf{y} = y(\mathbf{s}, \mathbf{u}, \dot{\mathbf{u}}, \dots, \mathbf{u}^p)$, such that the state-partition $\mathbf{s}$ and input $\mathbf{u}$ can be expressed as $\mathbf{s} = s(\mathbf{y}, \dot{\mathbf{y}}, \dots, \mathbf{y}^q)$, $\mathbf{u} = u(\mathbf{y}, \dot{\mathbf{y}}, \dots, \mathbf{y}^q)$ respectively, where p, q are finite integers, and the r-state dynamics are that of one or more chains of integrators.

If k = n, the partial flatness is equivalent to differential flatness and the discussed system represents a specific type of nonlinear system with a defect equivalent to the dimension of $\mathbf{r}$ as discussed in [Fli95].

To explore the concept of a partial differential flat system, consider the example of the cart-pole system, as depicted in fig. 2.2 and discussed in [Ram14].

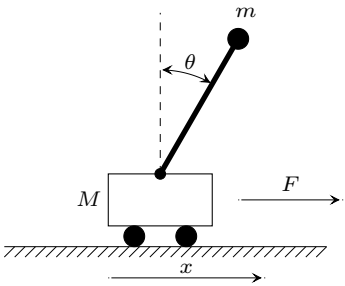


Figure 2.2.: Illustration of a cart-pole system with an inverse pendulum.

The dynamics of the cart-pole system are captured by the following equations of motion.

$$\begin{bmatrix} (M+m) & -ml\cos\theta \\ -ml\cos\theta & ml^2 \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} ml\dot{\theta}^2 \sin\theta \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -mlg\sin\theta \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix}, \quad (2.4)$$

where M is the mass of the cart, m the mass of the pendulum bob, θ the angle displacement of the pole from the vertical, F the force applied to the cart, x the horizontal displacement of the cart, l the length of the pole, and g the gravitational acceleration.

Considering $\mathbf{z} = \theta$ as the flat output, manipulation of the equations of motion yields:

$$F = (M+m)\left(l\ddot{\theta} - g\sin\theta\right)\cos\theta - ml\ddot{\theta}\cos\theta + ml\dot{\theta}^2\sin\theta,$$
$$\ddot{x} = \left(l\ddot{\theta} - g\sin\theta\right)\cos\theta.$$

From eq. (2.5), it is found that the control input F and the s-states $[\theta, \dot{\theta}]$ are functions of the flat output and its higher-order derivatives. Moreover, with the cart acceleration $\ddot{x}$ known from eq. (2.5) and given the initial conditions of the r-states, $[x(0), \dot{x}(0)]$, integration of eq. (2.5) allows for the determination of all system states.

This example demonstrates the application of partial differential flatness in computing the control input through the flat outputs and their derivatives.

Partial differential flat systems offer significant advantages and a wide range of applications in control systems and trajectory planning. They enable the separation of control design into flat and non-flat components, making control strategies more manageable, particularly for complex or high-dimensional systems. This decoupling leads to enhanced flexibility in control design and more efficient trajectory planning, as it obviates the need for integrating the system's dynamics for the flat part of the system. Such an approach is not only computationally efficient but also extends the applicability of flatness-based methods to a broader class of systems, including those that are not fully differentially flat [Ram14].

Moreover, partial differential flat systems are particularly relevant for under-actuated systems, where the number of control inputs is less than the number of states. By focusing on the flat components, the control of these systems is simplified, enhancing tractability.

Furthermore, integration can be avoided in numerical trajectory optimization by planning trajectories for flat and non-flat states. This method allows for obtaining control inputs and related functions directly from the state derivatives, facilitating the generation of optimized trajectories and imposing equality constraints on the system dynamics without the need for integration.

2.1.3. Stability Criteria and Analysis

Control theory defines stability as the property where any slight deviation or disturbance in the system's initial state results in bounded and manageable responses over time.

In mathematical terms, a system is stable if, for any given initial state $\mathbf{x}_0$ and control input $\mathbf{u}(t)$, the state $\mathbf{x}(t)$ remains bounded and converges to a specified equilibrium point or trajectory as time t progresses to infinity [Slo91, Lya92]. The following section explains three primary stability criteria: Lyapunov Stability, Asymptotic Stability, and Exponential Stability.

Lyapunov Stability

Lyapunov stability is a fundamental concept in control theory that concerns the behavior of a system's state in the vicinity of an equilibrium point. An equilibrium point $\mathbf{x}_e$ is said to be Lyapunov stable if, for every small positive number ϵ, there exists a corresponding positive number δ such that if the initial state $\mathbf{x}(0)$ is within a distance δ from $\mathbf{x}_e$, the state $\mathbf{x}(t)$ remains within a distance ϵ from $\mathbf{x}_e$ for all future times $t \geq 0$. In other words, small perturbations in the initial state do not cause the system's state to deviate significantly from the equilibrium point, ensuring that the system remains stable over time.

Asymptotic Stability

Asymptotic stability builds upon Lyapunov stability by ensuring that solutions that begin close enough to an equilibrium point remain close and converge toward the equilibrium as time progresses.
An equilibrium point is asymptotically stable if it's Lyapunov stable and if a region of initial states exists within a distance $\delta > 0$ from the equilibrium. When the system initiates within this region, the solutions will ultimately converge to equilibrium as time approaches infinity.

Exponential Stability

Exponential stability takes stability a step further by guaranteeing convergence at an exponential rate. An equilibrium point is exponentially stable if asymptotically stable and positive constants α, β, and δ exist. These constants ensure that if the system starts within a distance $\delta > 0$ from the equilibrium, the solutions converge to the equilibrium point with a decay rate faster than or equal to $\alpha\|x(0) - x_e\|e^{-\beta t}$ for all $t \geq 0$.

Lyapunov Control

The Direct Method of Lyapunov is a mathematical technique used to prove the equilibrium stability of nonlinear dynamical systems. A physical system whose total energy decreases will eventually reach an equilibrium point [Kha02, Lya92].

Consider a system represented as $\dot{\mathbf{x}} = f(\mathbf{x})$ with an equilibrium point $\mathbf{x}_\mathrm{e}$, the Direct Method states the stability of $\mathbf{x}_\mathrm{e}$ if there exists a continuously differentiable energy function $V(\mathbf{x}) : \mathbb{D} \to \mathbb{R}$ within a region $\mathbb{D} \subset \mathbb{R}^\mathrm{n}$ that satisfies the following conditions:

- $V(\mathbf{x}_\mathrm{e}) = 0$.
- $V(\mathbf{x}) > 0$ in $\mathbb{D}\backslash\{\mathbf{x}_\mathrm{e}\}$.
- $\dot{V}(\mathbf{x}) \leq 0$ in $\mathbb{D}$.

Additionally, $\dot{V}(\mathbf{x}) < 0$ in $\mathbb{D}$ must be satisfied for all $\mathrm{x} \in \mathbb{D} \neq \mathrm{x}_\mathrm{e}$ for asymptotic stability.
Lyapunov Control (LC) is an application of the Direct Method for control design of systems of the form $\dot{\mathbf{x}} = f(\mathbf{x},\mathbf{u})$. Its objective is to determine a stabilizing feedback control law $\mathbf{u} = c(\mathbf{x})$. The design challenge lies in selecting the state feedback in a way that allows the existence of a Lyapunov function satisfying the stability conditions mentioned above. The choice of an appropriate Lyapunov function and control law depends

on the specific system's structure, complexity, and dynamics, making it highly individualized.

2.1.4. Optimal Control

Optimal control theory is a mathematical framework for determining an optimal control action over time for a given system. With its applications spanning aerospace engineering to economics, this theory has proven invaluable due to the inherent complexities of dynamical systems and the diverse objectives they encompass [Ber95].

Optimal control theory is fundamentally a concept of optimizing a designed performance measure. Mathematically, the objective is to optimize a function over a defined period. This mathematical function is commonly referred to as the cost or objective function [Ste94]. The aim is to determine the control action that either minimizes or maximizes this integral, depending on the specific optimization goal subjective to the system dynamics and constraints.

The general form of an optimal control problem is:

$$\begin{aligned} \underset{\mathbf{u}(t)}{\text{minimize}} \quad & \int_{t_0}^{t_\mathrm{f}} L(\mathbf{x}(t), \mathbf{u}(t), t)\, dt \\ \text{subject to} \quad & \dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t), t), \\ & \mathbf{x}(t_0) = \mathbf{x}_0, \end{aligned} \tag{2.7}$$

where $\mathbf{x}(t)$ is the state of the system, $\mathbf{u}(t)$ is the control input, $L(\mathbf{x}(t), \mathbf{u}(t), t)$ is the running cost, $L(\mathbf{x}(t_\mathrm{f}), \mathbf{u}(t_\mathrm{f}), t_\mathrm{f})$ is the terminal cost and $f(\mathbf{x}(t), \mathbf{u}(t), t)$ represents the system dynamics, and $\mathbf{x}_0$ is the initial state.

In optimal control, the terminal cost $L(\mathbf{x}(t_\mathrm{f}), \mathbf{u}(t_\mathrm{f}), t_\mathrm{f})$ plays a crucial role in guiding the system toward a desired final state while ensuring stability. For example, setting the terminal cost as a quadratic penalty on the deviation from the goal state can drive the system to a stable

equilibrium.

$$L(\mathbf{x}(t_\mathrm{f}), \mathbf{u}(t_\mathrm{f}), t_\mathrm{f}) = (\mathbf{x}(t_\mathrm{f}) - \mathbf{x}_\mathrm{goal})^\top Q_f (\mathbf{x}(t_\mathrm{f}) - \mathbf{x}_\mathrm{goal}),$$

where Q_f is a positive definite matrix that penalizes deviations from the desired final state $\mathbf{x}_\mathrm{goal}$. Additionally, in Receding Horizon Control (RHC) such as Model Predictive Control (MPC), a well-designed terminal cost, combined with a terminal constraint, can guarantee closed-loop stability by enforcing the Lyapunov stability conditions.

One foundational tool in optimal control theory is the Hamiltonian method, which offers a systematic approach to solving general optimization problems. The Hamiltonian function:

$$H(\mathbf{x}(t), \mathbf{u}(t), \lambda(t), t) = L(\mathbf{x}(t), \mathbf{u}(t), t) + \lambda(t)^\mathrm{T} f(\mathbf{x}(t), \mathbf{u}(t), t). \quad (2.8)$$

where $\lambda(t)$ represents the costate vector. By applying Pontryagin's Principle, which provides necessary conditions for optimality [Pon62], the Hamiltonian function is optimized to compute an optimal control $\mathbf{u}^*$. For a detailed explanation of this method, refer [Föl88].

In optimal control theory, the complexity inherent in system dynamics, cost functions, and constraints leads to various optimal control methodologies emerging from the general optimal control problem. A prime example is the Linear Quadratic Regulator (LQR), which optimizes a quadratic objective function across an infinite horizon assuming linear system dynamics. This section will delve into crucial control methodologies pivotal to this work, including the Linear Quadratic Regulator, Linear Quadratic Model Predictive Control (MPC), Nonlinear Model Predictive Control (NMPC), and Differential Dynamic Programming (DDP).

Linear Quadratic Regulator

The Linear Quadratic Regulator (LQR) [Kha02] is a control design technique for linear time-invariant systems described by the state-space representation $\dot{\mathbf{x}} = \mathrm{A}\mathbf{x} + \mathrm{B}\mathbf{u}$. It aims to compute a control law $\mathbf{u} = -\mathrm{K}\mathbf{x}$ that minimizes the following cost function:

$$J = \int_0^\infty (\mathbf{x}^{\mathrm{T}}\mathrm{Q}\mathbf{x} + \mathbf{u}^{\mathrm{T}}\mathrm{R}\mathbf{u})\, dt,$$

where $\mathbf{x}$ is the state vector, $\mathbf{u}$ is the control input, Q is a positive semi-definite matrix which defines the weights for the state, R is the control input weighting matrix which is a positive definite matrix, and K is the feedback gain matrix.
The goal of the Linear Quadratic Regulator (LQR) is to determine the control law $\mathbf{u}^*(t) = -\mathbf{K}\mathbf{x}(t)$ to regulate the state deviation from zero optimally. This feedback gain matrix $\mathbf{K}$ is computed by solving the Algebraic Riccati Equation (ARE) of the form:

$$\mathrm{PA} + \mathrm{A}^{\mathrm{T}}\mathrm{P} - \mathrm{PBR}^{-1}\mathrm{B}^{\mathrm{T}}\mathrm{P} + \mathrm{Q} = 0,$$

where P is a constant, time-invariant matrix. Solving this equation yields the matrix P, which is then used to compute the feedback gain matrix K in the control law. The control law can be expressed as follows:

$$\mathbf{u}^*(t) = -\mathrm{K}\mathbf{x}(t), \quad \text{where} \quad \mathrm{K} = \text{const.} \tag{2.9a}$$

$$= -\mathrm{R}^{-1}\mathrm{B}^{\mathrm{T}}\mathrm{P}\mathbf{x}, \quad \text{where} \quad \mathrm{P} = \text{const.} \tag{2.9b}$$

The choice of R and Q matrices in the ARE is crucial and can significantly affect the control performance. Detailed derivations, documentation, and methods for solving the ARE for time-invariant and time-variant LQR can be found in [Föl88].

Model Predictive Control (MPC)

Model Predictive Control (MPC) represents an optimal control strategy utilized across diverse sectors such as chemical processing, automotive, aerospace, and energy systems. MPC leverages a predictive system model to predict the behavior over a specified time horizon [Raw09, May00]. This predictive capability enables MPC to calculate optimal control actions by solving an optimization problem that minimizes a cost function subjective to system dynamics under system constraints, thereby facilitating proactive decision-making for optimized performance and safety.

MPC is a Receding Horizon Control (RHC), which iteratively solves an optimization problem at each control step to determine a sequence of future control actions, executing only the first control action. The receding horizon control scheme allows MPC to adjust to updated measurements and disturbances dynamically, resulting in better constraints handling and increased robustness against prediction inaccuracies and system uncertainties.

MPC offers a distinct advantage over traditional control methods, such as the Hamiltonian approach, by enabling optimization over a finite prediction horizon without necessitating the solution of differential equations. This optimization of input variables simplifies the incorporation of state and input constraints, allowing the constrained solution to be efficiently determined using numerical optimization techniques, including quadratic and nonlinear programming.

The goal of MPC is to identify a sequence of control inputs $u = [\mathbf{u}_{\mathrm{i}}, \dots, \mathbf{u}_{\mathrm{i+N-1}}]^{\mathrm{T}}$ that minimizes a specified cost function. The generic

optimization framework for MPC is outlined as follows:

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & J = l_f(\mathbf{x}_{\mathrm{i+N}}) + \sum_{j=\mathrm{i}}^{\mathrm{i+N-1}} l(\mathbf{x}_j, \mathbf{u}_j) \\ \text{subject to} \quad & \mathbf{x}_{j+1} = f(\mathbf{x}_j, \mathbf{u}_j), \\ & \mathbf{x}_j \in \mathcal{X}, \\ & \mathbf{u}_j \in \mathcal{U}, \end{aligned} \tag{2.10}$$

where $\mathbf{x}_j \in \mathbb{R}^{\mathrm{n}}$ represents the state vector and $\mathbf{u}_j \in \mathbb{R}^{\mathrm{m}}$ the control input vector at any arbitrary time step j. The function $f : \mathbb{R}^{\mathrm{n}} \times \mathbb{R}^{\mathrm{m}} \to \mathbb{R}^{\mathrm{n}}$ is a nonlinear mapping of the system dynamics. Here, $l_f(\cdot)$ denotes the terminal cost function, $l(\cdot, \cdot)$ the running cost function, $\mathcal{X} \in \mathbb{R}^{\mathrm{n}}$ the set of feasible states, and $\mathcal{U} \in \mathbb{R}^{\mathrm{m}}$ the set of feasible control inputs. This inherently nonlinear problem can be solved using nonlinear programming solvers.

Linear Quadratic Model Predictive Control (LQ-MPC), often synonymous with MPC due to its widespread adoption, simplifies the generic optimization problem by assuming linear system dynamics, given by the following Linear Time-Varying (LTV) system:

$$\begin{aligned} \mathbf{x}_{j+1} &= \mathrm{A}_j \mathbf{x}_j + \mathrm{B}_j \mathbf{u}_j, \\ \mathbf{y}_j &= \mathrm{C}_j \mathbf{x}_j + \mathrm{D}_j \mathbf{u}_j, \end{aligned} \tag{2.11}$$

In LQ-MPC, the optimization objective is a quadratic cost function usually aimed at minimizing energy deviations from desired state $\mathbf{x}_j^*$ and control input $\mathbf{u}_j^*$ trajectories over the prediction horizon N. An example of the cost function is given as:

$$\begin{aligned} J = \sum_{j=\mathrm{i}}^{\mathrm{i+N-1}} \Big(& (\mathbf{x}_j - \mathbf{x}_j^*)^T \mathbf{Q} (\mathbf{x}_j - \mathbf{x}_j^*) + (\mathbf{u}_j - \mathbf{u}_j^*)^T \mathbf{R} (\mathbf{u}_j - \mathbf{u}_j^*) \Big) \\ & + (\mathbf{x}_{\mathrm{i+N}} - \mathbf{x}_{\mathrm{i+N}}^*)^T \mathbf{P} (\mathbf{x}_{\mathrm{i+N}} - \mathbf{x}_{\mathrm{i+N}}^*). \end{aligned}$$

LQ-MPC formulates the optimization problem as a quadratic programming problem that mandates the constraints to be linear, thereby aligning with the linear nature of the system model.

Model Predictive Control (MPC) delineates two principal objectives: state regulation and reference tracking. Each strategy optimizes the control inputs to achieve specific system behavior, yet they are fundamentally distinct in their goals and the design of their cost functions.

State Regulation

State regulation focuses on guiding the system to a minimum energy state, often the origin. The aim is to minimize the deviation of system states from zero, effectively stabilizing the system at the origin. This approach is crucial for processes requiring stabilization or return to a steady state following disturbances.

The cost function for state regulation is typically formulated as:

$$J = \sum_{j=\mathrm{i}}^{\mathrm{i+N-1}} l(\mathbf{x}_j, \mathbf{u}_j) + l_f(\mathbf{x}_{\mathrm{i+N}}),$$

where J represents the total cost, $l(\mathbf{x}_j, \mathbf{u}_j)$ denotes the running cost at step j, incorporating the cost associated with the state $\mathbf{x}_j$ and control input $\mathbf{u}_j$, and $l_f(\mathbf{x}_{\mathrm{i+N}})$ is the terminal cost, related to the final state $\mathbf{x}_{\mathrm{i+N}}$. A common expression for the cost function used in LQ-MPC is:

$$l(\mathbf{x}_j, \mathbf{u}_j) = \mathbf{x}_j^{\mathrm{T}} \mathrm{Q} \mathbf{x}_j + \mathbf{u}_j^{\mathrm{T}} \mathrm{R} \mathbf{u}_j,$$

with Q and R being the weighting matrices that prioritize state minimization and control effort, respectively.

Reference Tracking

Reference tracking aims to minimize the deviation of the system output from a given desired reference trajectory. The cost function for reference

tracking is given by:

$$J = \sum_{j=\mathrm{i}}^{\mathrm{i+N-1}} (\mathbf{y}_j - \mathbf{r}_j)^{\mathrm{T}} \mathrm{Q} (\mathbf{y}_j - \mathbf{r}_j) + \mathbf{u}_j^{\mathrm{T}} \mathrm{R} \mathbf{u}_j + (\mathbf{y}_{\mathrm{i+N}} - \mathbf{r}_{\mathrm{i+N}})^{\mathrm{T}} \mathrm{Q}_{\mathrm{f}} (\mathbf{y}_{\mathrm{i+N}} - \mathbf{r}_{\mathrm{i+N}}),$$

where $\mathbf{y}_j$ and $\mathbf{r}_j$ represent the system output and the reference state at step j, respectively. Here, Q and R are weighting matrices that balance the importance of accurate reference tracking against the control input costs.

The optimization problem formulation for MPC exhibits significant variability, depending on the application, the selection of performance criteria, and system modeling approaches. This section provided a foundational overview of MPC, highlighting Non-Linear Model Predictive Control (NMPC) and LQ-MPC. The formulation of MPC employed in this research, including the detailed application context, chosen performance criteria, and system modeling, is conceptualized in chapter 5.

2.1.5. Optimization Methods

Optimization methods are essential for solving specific problem types in various disciplines. This section briefly introduces norm minimization, quadratic programming (QP), nonlinear programming (NLP), and Differential Dynamic Programming (DDP), focusing on their application through available solvers rather than detailed exploration.

Norm Minimization

Norm minimization seeks to minimize the norm of a vector, typically representing errors or deviations. It is formulated as:

$$\min \|\mathbf{x} - \mathbf{b}\|_p, \tag{2.12}$$

where $\|\|\|_p$ denotes the L_p norm. Common choices include the L_0 norm, which promotes sparsity; the L_1 norm, which minimizes a vector's absolute value; and the L_2 norm, which is used in the least squares minimization. Applications range from regression analysis to signal processing.

Quadratic Programming (QP)

Quadratic Programming involves minimizing a quadratic objective function subjective to linear constraints:

$$\min_{\mathbf{x}} \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathrm{H}\mathbf{x} + \mathbf{g}^{\mathrm{T}}\mathbf{x} \quad \text{s.t.} \quad \mathrm{A}\mathbf{x} \leq \mathbf{b}, \tag{2.13}$$

where H is a symmetric positive semi-definite matrix. Quadratic Programming (QP) problems can be efficiently solved using a variety of solvers, including Matlab's quadprog, CVXGEN, Qpoases CVXOPT, and OSQP for open-source options, and commercial solvers like Gurobi and MOSEK. Each offers unique strengths, from handling large-scale problems to providing high-speed and accurate solutions, catering to a wide range of optimization needs.

Nonlinear Programming (NLP)

Nonlinear Programming addresses problems with nonlinear objectives or constraints:

$$\min f(\mathbf{x}) \quad \text{s.t.} \quad g_i(\mathbf{x}) \leq 0,\ h_j(\mathbf{x}) = 0. \tag{2.14}$$

NLP's broad applications include optimal control, energy optimization, and model parameter estimation. Solution methods vary from Sequential Quadratic Programming and Interior Point Methods to Gradient Descent techniques.

Differential Dynamic Programming (DDP)

The principle of optimality, central to Differential Dynamic Programming (DDP) and many other dynamic optimization methods, was formalized by Richard Bellman in the 1950s [Bel57, Jac70]. This principle is defined by Bellman's equation, which serves as the foundation for dynamic programming. Bellman's equation provides a recursive decomposition strategy for solving optimization problems by breaking them into smaller, simpler subproblems.

The principle states that an optimal policy has the property that, regardless of the initial state and initial decision, the remaining decisions must constitute an optimal policy about the state resulting from the first decision. This recursive nature allows for the efficient computation of optimal strategies across various problems, from resource allocation to the control of dynamic systems. Bellman's work on the principle of optimality has profoundly influenced the optimization field, contributing to the development of algorithms like DDP capable of tackling complex problems in high-dimensional spaces and nonlinear dynamics.

2.2. Related Work

Stabilizing trailer trajectory, especially during reverse driving or finding an optimal trajectory of trailers, has been a longstanding research topic. Numerous researchers have contributed to this field, exploring various methods and approaches to enhance control and optimization of trailer systems. With advancements in modern computing, Model Predictive Control (MPC) has found significant applications in this area. This section reviews state-of-the-art studies and methodologies related to trailer control, focusing on feedback control techniques and the application of MPC. The discussion highlights the evolution of control strategies, the impact of computational advancements, and the current state-of-the-art in trailer trajectory optimization.

2.2.1. Feedback Control of Articulated Vehicles

Feedback Linearization

This section explores feedback linearization methods focused on two main objectives: asymptotic curvature stabilization and effective path tracking, as discussed in [Wer14]. Maneuvering a trailer system in reverse poses significant challenges due to its nonholonomic constraints, complicating the application of conventional control techniques. Feedback linearization addresses these challenges by transforming nonlinear system dynamics into a linear form, simplifying the control problem [Kha02]. For reversing a one-trailer system, this method calculates control inputs that negate the nonlinear behavior induced by the trailer's kinematics, resulting in a more manageable linear system model.

A critical aspect of reversing a trailer system is stabilizing its curvature to ensure it can follow a desired path without deviation. Asymptotic curvature stabilization achieves this by adjusting the steering inputs so that the trailer's curvature approaches the desired curvature over time. This ensures the trailer's trajectory converges to the planned path, enhancing the accuracy and reliability of reverse maneuvers.

Path tracking in reverse operations of a one-trailer system concerns the vehicle's ability to follow a predetermined path accurately. By leveraging the linearized system model from feedback linearization, control algorithms minimize deviations from the desired path, accounting for dynamic constraints and external disturbances. Werling et al. [Wer14] demonstrate the effectiveness of feedback linearization for asymptotic curvature stabilization and path tracking based on experimental results.

However, Astolfi et al. [Ast01] discuss the limitations of input-output linearization for tracking the trailer's lateral offset. They observed that the stability of the resulting zero-dynamics is guaranteed when the guide point is appropriately selected. Notably, as indicated in [DeS98], this stability does not hold when the hitching point between the tractor and

trailer is in front of the tractor's rear axle, resulting in a negative off-axle distance. Furthermore, the stability property is local, meaning convergence is assured only when the initial vehicle configuration closely approximates the desired one.

Linear Quadratic Regulator

This section examines the application of Linear Quadratic Regulator (LQR) controllers for hitch angle stabilization and feedback control in multi-trailer systems.

Evestedt et al. [Eve16] present a cascaded control approach for path tracking and stabilization of a reversing general 2-trailer configuration with off-axle hitching. This methodology integrates a low-level LQR controller for stabilizing internal angles and a high-level pure pursuit path tracking controller. The LQR controller, designed using a linearized model of the 2-trailer system, aims to stabilize trailer angles around an equilibrium point. In contrast, the pure pursuit controller adjusts the hitch angle based on a look-ahead distance to maintain the reference path. Simulation and experimental results demonstrate the effectiveness of this approach, exhibiting minimal tracking errors validated using a small-scale test platform.

In another work, Ljungqvist et al. [Lju16, Lju15] propose an LQR controller for stabilizing a 2-trailer system with off-axle hitching, enabling the system to follow planned paths during reverse motion. Utilizing a kinematic model, the controller ensures closed-loop stability around predefined paths. A linearized model in the Frenet frame describes tracking error dynamics. The LQR controller stabilizes the system around a nominal path, with feedback gain independent of velocity, ensuring consistent performance in both forward and reverse motions.

Simulations validate the controller's performance and its capability to handle disturbances effectively. The proposed controller successfully stabilizes the system around any precalculated path. However, experimental validation is missing.

2.2.2. Trajectory Planning and Control of Articulated Vehicle

Trajectory optimization techniques for nonholonomic systems address challenges in highly constrained environments [Lam03]. Researchers have explored various methodologies to address the complexities of trajectory planning [Hao05, Zha24b, Sal23, Nat23].

Trajectory planning for the truck trailer system is even more complex. Obstacle avoidance, a critical and integral aspect of trajectory planning, is addressed through sample-based planning methods such as lattice-based planners and RRT. Model-based trajectory optimization techniques often stabilize a vehicle on a planned reference through sample-based planners, while a high-level planer handles collision avoidance. The numerical instability of the system limits the application of model-based approaches for trajectory planning and collision avoidance. This section details various model-based approaches for truck-trailer systems.

Ljungqvist et al. [Lju20] propose a model predictive path-following controller for low-speed backward maneuvers of multi-steered articulated vehicles comprising a car-like tractor with a steerable front wheel and N trailers. The controller aims to follow nominal paths while satisfying physical constraints, thus preventing undesirable states such as jack-knifing. The path-following error model is defined similarly to their previous work [Lju19]. However, instead of formulating the problem as a Linear Quadratic Regulator (LQR), it is approached as a Model Predictive Control (MPC) problem. This formulation allows for applying constraints on the system inputs and states to enhance stability.

The proposed solution employs a nominal reference trajectory from a motion planner that adheres to state and control input constraints. Although the paper does not detail the computation of the nominal trajectory, it assumes the trajectory includes the entire state and control-input information, inferred to generate feasible paths by considering the vehicle's dynamics and constraints. The system model is computed by linearizing the lateral deviations and orientation errors around the reference path. Constraints on control inputs and hitch angles are incorporated, and the cost function penalizes path-following errors and control effort. Soft constraints for hitch-angle limits ensure robustness against modeling approximations.

Simulation results demonstrate the controller's effectiveness in tracking straight and figure-eight nominal paths, outperforming an LQ controller developed in [Lju19]. A lattice-based trajectory planning framework generates kinematically feasible and collision-free paths, coupled with a controller to stabilize lateral and angular errors during execution [Lju19].

Bergman et al. [Ber20] addressed the challenge of trajectory planning in unstructured, cluttered environments with an optimization-based receding horizon algorithm.

The optimal motion planning problem involves finding a feasible, collision-free trajectory while minimizing a performance measure. Due to the problem's nonconvex nature, approximate methods, such as sampling-based planners, are commonly used. Popular methods include random sampling-based planners like RRT star and deterministic sampling-based planners like lattice-based motion planners. While these methods can compute feasible solutions, they often require post-optimization steps to improve solution quality.

Bergman et al. similar to [Lju19] integrates motion planning and optimization, significantly enhancing solution quality and reliability. The proposed receding horizon trajectory planning approach reduces computation time, enabling faster execution. The approach addresses motion

planning for continuous-time nonlinear systems with constraints and obstacle avoidance, formulated as a continuous-time optimal control problem (OCP) with constraints on state and control signals. Considering environmental obstacles, it starts by using a motion planner to compute a nominal trajectory from the initial state to the desired terminal state.

The optimization-based receding horizon planner optimizes the nominal trajectory using an iterative approach. The planner solves a nonlinear optimal control problem over a sliding time window, ensuring a feasible trajectory beyond the planning horizon. This iterative process guarantees recursive feasibility, meaning a feasible solution exists at each iteration. The total objective function value is non-increasing, ensuring convergence towards the terminal state. The algorithm can handle piecewise continuous nominal control trajectories by adjusting the problem formulation.

The proposed approach is evaluated in challenging parking scenarios for a truck and trailer system. The nominal trajectories are computed using a lattice-based motion planning algorithm. The system is modeled as a 2-trailer with a car-like truck, with control signal and state constraints. The cost function in the optimization-based receding horizon planning (RHP) approach optimizes various parameters, resulting in significant improvements in objective function value and total time compared to the nominal solution. These improvements are particularly notable in parallel parking scenarios, with planning horizons of 60-80 seconds yielding the best results.

The two-step trajectory planning algorithm combines search-based motion and optimization-based receding horizon planning. This algorithm provides theoretical guarantees on recursive feasibility, non-increasing objective function value, and convergence to the terminal state. The approach is suitable for challenging trajectory planning problems for a truck and trailer system. However, the algorithm needs modification to ensure real-time performance using fast MPC for dynamic environments.

Wu and Hung (2017) also developed a model predictive controller (MPC) for a tractor-trailer system, focusing on minimizing tracking errors and ensuring stable performance in both forward and backward motions.

The proposed controller [Wu17] is designed to solve the path tracking problem by minimizing the trailer tracking error. The system uses a kinematic model with control delay, and the error state is defined based on the curvature of the reference path. The error state vector x is defined as:

$$x = [e_{\phi_r}, e_{\phi_t}, e_{y_t}, e_\delta],$$

where e_{ϕ_r} is the error in the trailer heading angle, e_{ϕ_t} is the error in the tractor heading angle, e_{y_t} is the error in the lateral position of the trailer, and e_δ is the error in the tractor steering angle.

The error system model is linearized around the steady-state values of the hitch angle (θ_{ss}) and the tractor steering angle (δ_{ss}) similar to the modelling in [Lju16].

The linearized model is utilized in the MPC to calculate the optimal control effort based on the predicted states and desired outputs. The proposed MPC significantly reduces tracking error at intersections of line and arc segments, eliminating lateral error and controlling trailer heading. Constraints on the hitch angle prevent the jackknife phenomenon, ensuring stability during maneuvers.

Simulation results demonstrate the MPC's satisfactory performance, showing its capability to guide the forward and backward system.

The approach by Wu and Hung is similar to the one proposed by Ljungqvist et al. (2020) [Lju20], which also employs a model predictive path-following controller. However, Ljungqvist et al. focus on a more complex system involving a multi-steered 2-trailer configuration, whereas Wu and Hung concentrate on a single trailer system. This distinction highlights the adaptability of MPC strategies across different articulated vehicle configurations.

Bourelius [Bou22] addresses the implementation of real-time MPC for an autonomous truck and trailer system, focusing on efficient quadratic program (QP) solvers and reducing the computational complexity.

The thesis aims to implement MPC in real-time applications, requiring efficient QP solvers to handle the optimization problem within the constraints of available computational resources. The experimental platform consists of a LEGO truck equipped with an onboard Raspberry Pi (RPi) and LEGO EV3 for implementing MPC. Various QP solvers, including DAQP, qpOASES, and the operator splitting method OSQP, are utilized to solve the MPC problem. The certification framework ensures the exact computational complexity of MPC using DAQP, allowing for a comparison of solver performance and guaranteeing maximum computational complexity.

The vehicle model consists of a car-like tractor, a dolly, and a semi-trailer, with parameters describing the length of each component. The kinematic model assumes low-speed maneuvers on a flat surface without wheel slippage. The system's stability varies with the direction of travel; it is unstable when reversing and stable when moving forward. The path-following controller aims to minimize the path-following error by stabilizing the vehicle around the planned path. This is achieved using a path-following error model, which describes deviations from the nominal path through error states representing lateral distance, orientation error, and joint angle errors.

The error states are linearized and discretized to form a linear system for the MPC problem. Special cases, such as the straight-path configuration, provide a simplified linear model with constant matrices, which are recalculated at each time step based on the reference configuration.

Two linearized models, including the straight-path configuration, were used in the MPC. Both models demonstrated similar path-following performance; however, the straight-path model allowed for faster computation times. Initial experiments showed that the exact iteration complexity could be determined by certifying every possible multi-parametric QP

(mpQP). The maximum number of iterations increased linearly with the MPC horizon when following a straight path.

This research highlights the practical application of MPC in real-time systems, contributing to the development of efficient and reliable control strategies for autonomous vehicles.

In conclusion, this section has explored MPC for path tracking and stabilization of articulated vehicles and has demonstrated that MPC outperforms traditional feedback control techniques. Wu and Hung (2017) developed an MPC for a tractor-trailer system, minimizing tracking errors by linearizing the error system model around the steady-state values of the hitch angle (θ_{ss}) and the tractor steering angle (δ_{ss}). This approach, formulated as a Quadratic Programming (QP) problem, was validated through simulations demonstrating satisfactory performance in avoiding jack-knife phenomena [Wu17]. However, the model's linearization around the steady-state values may limit its performance during transient maneuvers.

Similarly, Ljungqvist et al. proposed an MPC for multi-steered articulated vehicles, linearizing the path-following error model and solving the problem as a QP. Their simulations showed superior performance compared to a Linear Quadratic (LQ) controller [Lju19], particularly in handling a multi-steered 2-trailer system [Lju20]. However, the complexity of accurately modeling multi-steered systems may increase computational demands. Additionally, Ljungqvist et al. have not conducted real-world experiments on a full-scale test vehicle, and the limitations related to local stability and vehicle modeling inaccuracies still apply to work presented in [Lju19].

Bourelius (2022) focused on the real-time application of an MPC for a truck and trailer system. This approach involved linearizing path-following error states and employing QP solvers, including DAQP, qpOASES, and OSQP. The author validated the method on a LEGO truck with an onboard Raspberry Pi and LEGO EV3. It demonstrated effective real-time performance and identified DAQP as the best solver

due to its fast solution times and complexity certification [Bou22]. Despite its strengths, using a scaled-down LEGO truck limits the generalizability of the results to full-scale systems.

These studies collectively illustrate the robustness and versatility of MPC in handling path-following and stabilization tasks for articulated vehicles. They highlight using linearized models, efficient QP solvers, and validations through simulations and real-world applications. However, several challenges remain. Significant hurdles are ensuring robustness in dynamic environments, managing computational complexity, and generalizing findings to full-scale applications. Effectively tackling unstable dynamics is crucial, as the modeling heavily depends on the reference or nominal planned trajectory. The requirement for all reference states and inputs, the region of validity for the linearized model, and the impact of disturbances further complicate the implementation. Addressing these challenges is essential for advancing the practical application of MPC in real-world articulated vehicle systems.

3. System Modeling

Efficient trajectory planning and control for truck-trailer systems requires a thorough understanding of their dynamics. Section 3.1 begins with the modeling of the n-General Trailer (n-GT) system dynamics in chain form. In Section 3.2, the system dynamics for a One-General Trailer (one-GT) system are derived using the chain form of n-GT.

The concept of differential flatness is then explored in Section 3.3, highlighting its relevance and applicability to truck-trailer systems. This section establishes differential flatness for the One-Standard Trailer (one-ST) and the one-GT systems. A partial differential flatness is derived for the one-GT system, and a flat model for one-GT is derived. Finally, Section 3.5 presents a system analysis and comparison of the one-ST and one-GT systems.

3.1. n-General Trailer (n-GT): Chain Form

The n-General Trailer (n-GT) system, consisting of articulated trailers towed by a front wheel-steered vehicle is discussed in this section. The kinematic complexity and control strategies of such systems are critically dependent on the configuration of the hitching points, i.e., the locations at which trailers are articulated to their predecessor. Two primary configurations are recognized [Jea96, Lau93]:

1 **n-General Trailer (n-GT):** Trailers can be hitched at any arbitrary point along the length of a preceding trailer.

2 **n-Standard Trailer (n-ST):** Involves hitching each trailer directly at the axle of its predecessor, thereby simplifying the dynamics of the model.

This section explores the n-GT system, highlighting that the n-ST system is effectively a simplification of the n-GT model. The analysis assumes that trailers are rigidly connected to their predecessors, allowing only pivotal motion around the connecting joint without slippage. The key parameters defining the system include:

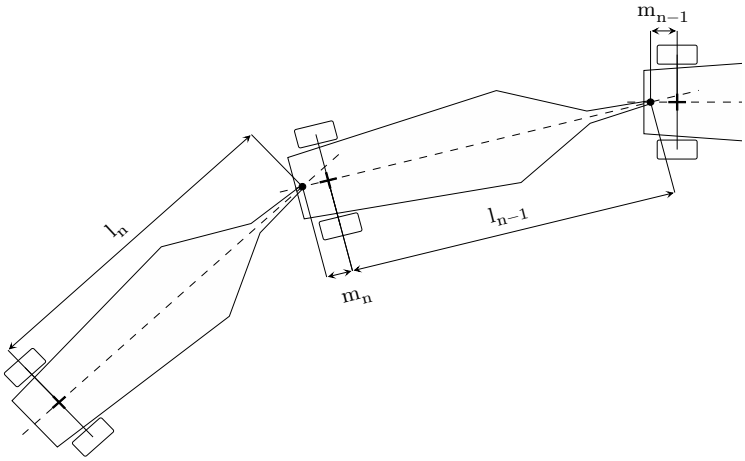


Figure 3.1.: Geometric illustration of a n-GT system. The parameters l_n and l_{n-1} represent the trailer's lengths, and m_n and m_{n-1} denote the distances between the axles of the trailers and their respective coupling points of the n^{th} trailer and its predecessor.

- ψ_0: Orientation of the truck.
- ψ_j: Orientation of the j^{th} trailer, for $j = 1, 2, \ldots, n$.
- (x_0, y_0): Position of the rear axle of the truck.

- (x_j, y_j): Position of the j^{th} trailer.
- l_j: Length of the j^{th} trailer.
- m_j: Distance between the hitch point of the j^{th} trailer, and the axle of its predecessor.
- δ: Steering angle of the truck.
- v: Velocity of the truck.

Figure 3.1 illustrates the n^{th} trailer and its predecessor and provides an illustrative overview of the parameters.

The kinematic equations for the n-GT system [Alt98, Alt01, Alt02] can be expressed as follows:

$$\begin{aligned}
\dot{x}_0 &= v \cos \psi_0, \\
\dot{y}_0 &= v \sin \psi_0, \\
\dot{\psi}_0 &= \frac{v}{l_0} \tan \delta, \\
\dot{x}_j &= v_j \cos \psi_j, \\
\dot{y}_j &= v_j \sin \psi_j, \\
\dot{\psi}_j &= \frac{v_{j-1}}{l_j} \sin(\psi_{j-1} - \psi_j) - \frac{m_j}{l_j} \cos(\psi_{j-1} - \psi_j) \dot{\psi}_{j-1}.
\end{aligned} \tag{3.1}$$

The position and orientation of each trailer can be defined based on the position and orientation of its predecessor, incorporating the trailer's length and the distance from its hitch point to the axle of its predecessor.

$$\begin{aligned}
x_j &= x_{j-1} - l_j \cos \psi_j - m_j \cos \psi_{j-1}, \\
y_j &= y_{j-1} - l_j \sin \psi_j - m_j \sin \psi_{j-1}, \\
\dot{\psi}_j &= \frac{v_{j-1}}{l_j} \sin(\psi_{j-1} - \psi_j) - \frac{m_j}{l_j} \cos(\psi_{j-1} - \psi_j) \dot{\psi}_{j-1}, \\
v_j &= v_{j-1} \cos(\psi_j - \psi_{j-1}) + m_j \sin(\psi_j - \psi_{j-1}) \dot{\psi}_{j-1}.
\end{aligned} \tag{3.2}$$

The system is structured as a chain, allowing the computation of successor states using a trailer mechanism, similar to the methods described in the works by [Sor93] and [Liz01].

3.2. One-General Trailer (one-GT): System Dynamics

Building on the foundational concepts introduced in section 3.1, this section is dedicated to the one-GT system dynamics, focusing on a configuration where a truck is followed by a single trailer. By applying a similar analytical approach, the equations of motion for a one-trailer system are derived. The motion of an one-GT system can be described by the following equations:

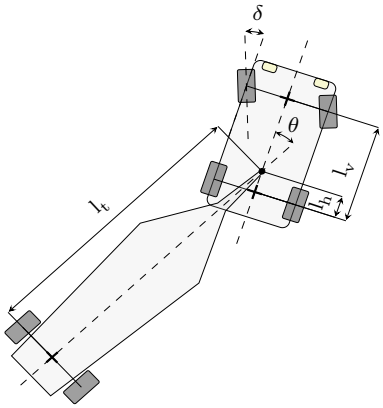


Figure 3.2.: Geometric Illustration of General one-Trailer: l_v, l_t, and l_h are truck's wheelbase, the trailer's length, and the distance between the rear axle of the truck, and the coupling point of the trailer, respectively.

$$\dot{x}_\mathrm{v} = v_\mathrm{v} \cos\varphi, \tag{3.3a}$$
$$\dot{y}_\mathrm{v} = v_\mathrm{v} \sin\varphi, \tag{3.3b}$$
$$\dot{\varphi} = \frac{v_\mathrm{v}}{\mathrm{l}_\mathrm{v}} \tan\delta, \tag{3.3c}$$
$$\dot{x} = v_\mathrm{v} \cos\psi, \tag{3.3d}$$
$$\dot{y} = v_\mathrm{v} \sin\psi, \tag{3.3e}$$
$$\dot{\psi} = \frac{v_\mathrm{v}}{\mathrm{l}_\mathrm{t}} \sin(\varphi - \psi) - \frac{v_\mathrm{v}\mathrm{l}_\mathrm{h}}{\mathrm{l}_\mathrm{t}\mathrm{l}_\mathrm{v}} \cos(\varphi - \psi) \tan\delta, \tag{3.3f}$$
$$v = v_\mathrm{v} \cos(\varphi - \psi) + \frac{v_\mathrm{v}\mathrm{l}_\mathrm{h}}{\mathrm{l}_\mathrm{v}} \sin(\varphi - \psi) \tan\delta, \tag{3.3g}$$

where $(x,\, y)$ denotes the position of the trailer's axle, and $(x_\mathrm{v},\, y_\mathrm{v})$ denotes the position of the truck's rear axle. φ, and ψ represent the orientation of the truck and trailer, respectively. Additionally, $\theta = \varphi - \psi$ stands for the hitch angle between the truck and the trailer, and δ represents the steering angle. Figure 3.2 demonstrates a one-GT model. This work primarily focuses on a one-trailer system; therefore, for simplicity, the parameters l_v, l_t, and l_h are introduced, representing the truck's wheelbase, the trailer's length, and the distance between the rear axle of the truck and the coupling point of the trailer, respectively.

3.2.1. System Dynamics: Trailer's Frame

Stabilizing the trailer during parking maneuvers in narrow spaces is crucial. Therefore, the equations of motion will be derived with respect to the trailer's motion to focus on the dynamics and stabilization of trailers, specifically maintaining a trailer on a reference trajectory. Consequently, the velocity of truck v_v in equations for trailer's motion eq. (3.3) are substituted with the trailer's velocity using eq. (3.3g). Drawing insights

from [Wer14] revised system of equations are presented below:

$$\dot{x}_\mathrm{v} = \frac{v \mathrm{l}_\mathrm{v} \cos\varphi}{\mathrm{l}_\mathrm{v} \cos(\varphi - \psi) + \mathrm{l}_\mathrm{h} \sin(\varphi - \psi) \tan\delta}, \tag{3.4a}$$

$$\dot{y}_\mathrm{v} = \frac{v \mathrm{l}_\mathrm{v} \sin\varphi}{\mathrm{l}_\mathrm{v} \cos(\varphi - \psi) + \mathrm{l}_\mathrm{h} \sin(\varphi - \psi) \tan\delta}, \tag{3.4b}$$

$$\dot{\varphi} = \frac{v \tan(\delta)}{\mathrm{l}_\mathrm{v} \cos(\varphi - \psi) + \mathrm{l}_\mathrm{h} \sin(\varphi - \psi) \tan(\delta)}, \tag{3.4c}$$

$$\dot{x} = v \cos\psi, \tag{3.4d}$$

$$\dot{y} = v \sin\psi, \tag{3.4e}$$

$$\dot{\psi} = \frac{v \left(\mathrm{l}_\mathrm{v} \sin(\varphi - \psi) - \mathrm{l}_\mathrm{h} \cos(\varphi - \psi) \tan(\delta)\right)}{\mathrm{l}_\mathrm{t} \left(\mathrm{l}_\mathrm{v} \cos(\varphi - \psi) + \mathrm{l}_\mathrm{h} \sin(\varphi - \psi) \tan(\delta)\right)}. \tag{3.4f}$$

The position of the truck can be directly mapped as an algebraic function of the trailer's position, the trailer's orientation, and the truck's orientation using eq. (3.2).

$$\begin{aligned} x_\mathrm{v} &= x + \mathrm{l}_\mathrm{t} \cos\psi + \mathrm{l}_\mathrm{h} \cos\varphi, \\ y_\mathrm{v} &= y + \mathrm{l}_\mathrm{t} \sin\psi + \mathrm{l}_\mathrm{h} \sin\varphi. \end{aligned} \tag{3.5}$$

Hence, solving all the differential equations in eq. (3.4) is not necessary, and the entire state space can be computed using state vector $\mathbf{x} = [x, y, \psi, \varphi]$ and the algebraic equation presented in eq. (3.5).

To address the challenges of asymptotic stabilization due to the nonholonomic nature at low speeds as highlighted in [Bro83], the system is

transformed from temporal to a spatial domain as suggested by Werling et al. [Wer14]. This transformation converts the system's equations relative to the distance traveled by the trailer.

$$x' = \cos\psi, \tag{3.6a}$$

$$y' = \sin\psi, \tag{3.6b}$$

$$\psi' = \frac{\mathrm{l_v}\sin(\varphi-\psi) - \mathrm{l_h}\cos(\varphi-\psi)\tan\delta}{\mathrm{l_t}\left(\mathrm{l_v}\cos(\varphi-\psi) + \mathrm{l_h}\sin(\varphi-\psi)\tan\delta\right)}, \tag{3.6c}$$

$$\varphi' = \frac{\tan\delta}{\mathrm{l_v}\cos(\varphi-\psi) + \mathrm{l_h}\sin(\varphi-\psi)\tan\delta}. \tag{3.6d}$$

The system outlined in eq. (3.6) is characterized by the continuous nonlinear system $\mathbf{x}' = f(\mathbf{x},\mathbf{u})$ with $\mathbf{x}_k = [x, y, \psi, \varphi]^T$, and the steering angle δ serves as the input $\mathbf{u}$ of the system.

3.3. Differential Flatness of the Truck-Trailer Systems

Differential flatness in a trailer system refers to a property where a set of flat outputs can fully describe the system's dynamics, typically the position of the rear axle of the last trailer. This property simplifies trajectory generation and control design for trailer systems. The kinematic model of a car with trailers is differentially flat, making trajectory planning a straightforward task [Bul96, Fli95]. By leveraging the differential flatness property of a truck-tractor-trailer system, trajectory planning algorithms can ensure compliance with the kinematic constraints while considering inter-vehicle collisions and path constraints [Hao05, Hao03]. Moreover, the differential flatness property enables the transformation of a system into a chained form, simplifying control design and stability analysis [Til95]. However, it is essential to note that not all trailer systems are inherently differentially flat, as highlighted by the need for specific assumptions and conditions for achieving differential flatness in

N-trailer systems [Mic17]. The flatness property allows for the linearization of the system's dynamic model using dynamic feedback linearization, a subset of differential flatness [Ryu08].

3.3.1. One-Standard Trailer (one-ST)

A One-Standard Trailer (one-ST) model refers to a configuration where a single trailer is directly hitched to the rear axle of a truck. In this model, the distance $\mathrm{l_h}$ to the hitch point is zero, simplifying the system's dynamics.

The n-ST system exhibits differential flatness, a property that allows its inputs and states to be expressed as functions of the flat outputs and their derivatives, as detailed in [Rou93]. For one-ST system, the flat output corresponds to the position of the trailer's axle, enabling a transformation of the state space $\mathbf{x} = [x_\mathrm{v}, y_\mathrm{v}, \varphi, x, y, \psi]^\mathrm{T}$, and the input $\mathbf{u} = [\delta, v_\mathrm{v}]^\mathrm{T}$ to a flat state space $\mathbf{x}_\mathrm{f} = [x_\mathrm{f}, y_\mathrm{f}]^\mathrm{T}$.

The heading ψ, and curvature κ of the trailer's path is calculated as:

$$\psi = \arctan \frac{y_\mathrm{f}'}{x_\mathrm{f}'}, \kappa = \frac{x_\mathrm{f}' y_\mathrm{f}'' - x_\mathrm{f}'' y_\mathrm{f}'}{\left(x_\mathrm{f}'^2 + y_\mathrm{f}'^2\right)^{\frac{3}{2}}}.$$

The entire system state can be computed from the flat output, using eq. (3.2) and eq. (3.3):

$$\begin{bmatrix} x_\mathrm{v} \\ y_\mathrm{v} \\ \varphi \\ \theta \end{bmatrix} = \begin{bmatrix} x_\mathrm{f} + \mathrm{l_t} \cos\psi \\ y_\mathrm{f} + \mathrm{l_t} \sin\psi \\ \psi + \arctan(\mathrm{l_t}\kappa) \\ \arctan(\mathrm{l_t}\kappa) \end{bmatrix}. \tag{3.7}$$

The steering angle δ, and the trailer's velocity are derived from the flat state, ensuring that the model's control inputs can be calculated directly from the path's curvature, and its rate of change as explained in [Rou93]

and derived in [Kva19] :

$$\delta = \arctan\left(\frac{\mathrm{l_v}}{\sqrt{1+\mathrm{l_t}^2\kappa^2}}\left(\kappa + \frac{\mathrm{l_t}}{1+\mathrm{l_t}^2\kappa^2}\alpha\right)\right), \tag{3.8}$$

where $\alpha = \frac{d\kappa}{ds}$ represents the rate of change of curvature with respect to arc length, and the trailer's velocity is given by $v = \sqrt{\dot{x}_\mathrm{f}^2 + \dot{y}_\mathrm{f}^2}$. The truck's velocity is similarly computed as [Rou93].

$$v_\mathrm{v} = \sqrt{1+\mathrm{l_t}^2\kappa^2}v. \tag{3.9}$$

Equation (3.8) establishes an important requirement for the continuity of the trailer's path. To ensure a continuous steering wheel angle, the path followed by the trailer must have at least G^3 continuity. G^3 continuity implies that the path's curvature, its first and second derivatives are all continuous.

Next, the system dynamics of an one-ST will be derived using differential flatness. The flat point corresponds to the center of the trailer's axle. The motion of the flat point is defined using the equation of an Euler spiral or clothoid, where the curvature changes linearly concerning the distance covered. Assume α is the rate change of the curvature defined as the sharpness w.r.t. distance. The equation of motion for the flat point is given by

$$x' = \cos\psi, \tag{3.10a}$$
$$y' = \sin\psi, \tag{3.10b}$$
$$\psi' = \kappa, \tag{3.10c}$$
$$\kappa' = \alpha_i. \tag{3.10d}$$

The state vector is augmented to include δ as an algebraic state, computed using eq. (3.8), resulting in a new state vector:

$$\mathbf{x}_i = [x_i, y_i, \psi_i, \kappa_i, \delta_i]^\mathrm{T}. \tag{3.11}$$

The hitch angle θ is computed from the curvature κ as:

$$\theta = \arctan(\mathrm{l_t}\kappa), \tag{3.12}$$

where $\mathrm{l_t}$ is the length of the trailer. The maximum curvature $\kappa_{\max}$ can be derived using this equation, thereby incorporating hitch angle constraints into the system. The trajectory planning and control for one-ST is possible using the system defined in eq. (3.10).

3.3.2. One-General Trailer (one-GT)

Unlike the n-ST configurations where differential flatness applies, a n-GT system is not inherently flat, as established in [Rou93]. However, an exciting exception exists for the one-GT system. In scenarios where the trailer is not directly attached to the center of the car's rear axle, the principle of flatness is applicable. The applicability of flatness to the one-GT system is thoroughly examined by Rouchon et al.. in [Rou93].

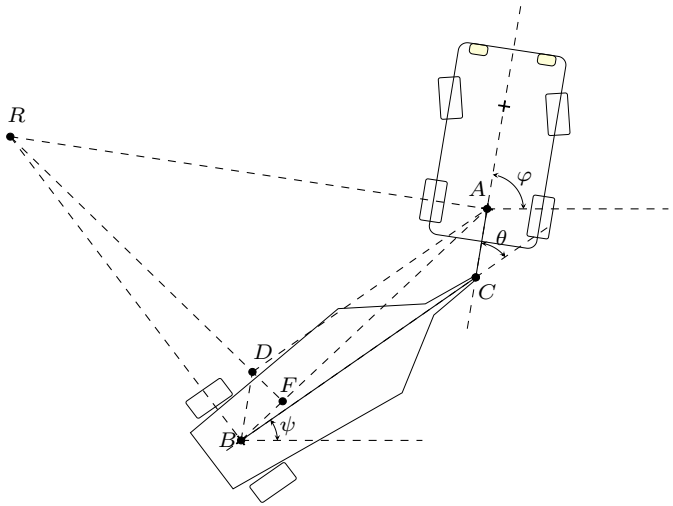


Figure 3.3.: Geometric Illustration of the Flat Point F and its Geometry for one-GT System

Based on a similar methodology as explained by [Rou93], a potential linearizing output is derived, denoted as $\mathbf{y}_\mathrm{f} = [x_\mathrm{f}, y_\mathrm{f}]^\mathrm{T}$, defined by the following relations:

$$x_\mathrm{f} = x_\mathrm{v} - \mathrm{l_t}\cos\psi - L(\theta)\frac{\mathrm{l_t}\sin\psi + \mathrm{l_h}\sin\varphi}{\sqrt{\mathrm{l_h}^2 + \mathrm{l_t}^2 + 2\mathrm{l_h}\mathrm{l_t}\cos\theta}},$$
$$y_\mathrm{f} = y_\mathrm{v} - \mathrm{l_t}\sin\psi + L(\theta)\frac{\mathrm{l_h}\cos\varphi + \mathrm{l_t}\cos\psi}{\sqrt{\mathrm{l_h}^2 + \mathrm{l_t}^2 + 2\mathrm{l_h}\mathrm{l_t}\cos\theta}},$$

where $L(\theta)$ is expressed as:

$$L(\theta) = \frac{-\mathrm{l_h}\mathrm{l_t}\sin\theta}{\sqrt{\mathrm{l_h}^2 + \mathrm{l_t}^2 + 2\mathrm{l_h}\mathrm{l_t}\cos\theta}}.$$

Here, $(x_\mathrm{f}, y_\mathrm{f})$ represent the Cartesian coordinates of point F, and the unit tangent vector τ at F aligns with segment AB as shown in fig. 3.3. The point R represents the instantaneous center of rotation for the line segment AB. $L(\theta)$ represents the length of the segment DF. The curvature κ is modeled as a function of θ as:

$$\kappa = K(\theta) = \frac{\sin\theta}{\cos\theta\sqrt{\mathrm{l_h}^2 + \mathrm{l_t}^2 + 2\mathrm{l_h}\mathrm{l_t}\cos\theta} - L(\theta)\sin\theta}.$$

Consequently, the set (x, y, φ, ψ) is a function of (F, τ, κ), and the steering angle δ is dependent on κ and its derivative with respect to the arc length s along F.
The dependencies of θ and δ solely on $(\kappa, \frac{d\kappa}{ds})$ emerge from the invariant nature of the problem under planar Euclidean transformations, illustrating how such physical and geometric symmetries can simplify calculations and aid in pinpointing the coordinates of the linearizing point F [Rou93]. The proof of flatness is also valid for a TS. By defining $\mathrm{l_h}$ as negative, the same equations remain applicable.

In contrast, the one-GT system exhibits a more complex relationship between curvature κ and hitch angle, making the derivation of analytical solutions for steering strategies significantly more challenging. This complexity limits the applicability of the flat model for one-GT systems in optimal control approaches such as MPC. The complexity is reduced by defining the system as a partially differential flat model, enhancing the applicability of optimization techniques and MPC.

3.4. Partial Differential Flatness: one-GT

The concept of partial differential flatness, as explored in section 2.1.2, also finds application in the one-GT system. The dynamics of truck-trailer systems, detailed in Section 3.2, are traditionally maneuvered by the steering angle.
This subsection proposes a partially differential flat model for truck-trailer configurations with off-axle hitching; this approach introduces the flat state $\mathbf{s} = [\theta]^{\mathrm{T}}$ while designating the remaining system states as the non-flat state $\mathbf{r} = [x, y]^{\mathrm{T}}$.

Utilizing Eq. eq. (3.6) to substitute expressions for ψ' and φ', the rate of change of the hitch angle, γ, is derived as follows:

$$\gamma = \varphi' - \psi' = \frac{\mathrm{l_t} \tan\delta - \mathrm{l_v} \sin\theta + \mathrm{l_h} \cos\theta \tan\delta}{\mathrm{l_t} \left(\mathrm{l_v} \cos\theta + \mathrm{l_h} \sin\theta \tan\delta\right)}. \tag{3.13}$$

By solving the equation above, the steering angle, δ, can be formulated as a function of the hitch angle rate, $\gamma = \varphi' - \psi'$, and the hitch angle, $\theta = \varphi - \psi$:

$$\delta = u(\mathbf{s},\mathbf{s}') = \arctan\left(\frac{\mathrm{l_v} \sin\theta + \mathrm{l_v}\mathrm{l_t}\gamma\cos\theta}{\mathrm{l_t} + \mathrm{l_h}\cos\theta - \mathrm{l_t}\mathrm{l_h}\gamma\sin\theta}\right). \tag{3.14}$$

The evolution of the non-flat state $\mathbf{r}$, given by $\mathbf{r}' = f(\mathbf{s},\mathbf{r})$, allows for the determination of all $\mathbf{r}$ states through integration, provided that initial conditions are known.

Next, the flat model for the one-GT can be derived as a differential-algebraic system of equations and given as follows:

$$\begin{aligned} x' &= \cos\psi, \\ y' &= \sin\psi, \\ \psi' &= \frac{\mathrm{l_v}\sin(\varphi-\psi) - \mathrm{l_h}\cos(\varphi-\psi)\tan\delta}{\mathrm{l_t}(\mathrm{l_v}\cos(\varphi-\psi) + \mathrm{l_h}\sin(\varphi-\psi)\tan\delta)}, \\ \varphi' &= \frac{\tan\delta}{\mathrm{l_v}\cos(\varphi-\psi) + \mathrm{l_h}\sin(\varphi-\psi)\tan\delta}, \\ \delta &= \arctan\left(\frac{\mathrm{l_v}\sin\theta + \mathrm{l_v}\mathrm{l_t}\gamma\cos\theta}{\mathrm{l_t} + \mathrm{l_h}\cos\theta - \mathrm{l_t}\mathrm{l_h}\gamma\sin\theta}\right). \end{aligned} \tag{3.15}$$

The state vector $\mathbf{x}$ and input $\mathbf{u}$ of the partial flat model are defined as follows:

$$\mathbf{x} = [x, y, \psi, \varphi, \delta] \in \mathbb{R}^5,$$

where x and y are the position coordinates, ψ is the heading angle, and φ is the truck's heading angle, δ is the steering angle and computed using an algebric equation.

The control input vector $\mathbf{u}$ is defined as:

$$\mathbf{u} = [\gamma] \in \mathbb{R},$$

where γ is the hitch angle rate. The flat model can be effectively used for trajectory planning and control. By leveraging partial differential flatness, it mitigates the numerical instability inherent to the one-GT system, ensuring more stable and reliable trajectory generation and execution. The application of the flat model for trajectory planning and control is presented in section 6.1 and section 5.2.3.

3.5. System Analysis and Comparison

In this section, a detailed analysis of the motion primitives for a truck-trailer system is conducted to traverse a lemniscate or figure-of-eight trajectory. Two different trailer configurations are explored: a one-ST configuration where the trailer is hitched directly at the rear axle, as detailed in section 3.3.1, and a one-GT configuration as explained in section 3.3.2. The model parameters are defined in table 3.1.

Table 3.1.: Parameters for the truck-semitrailer configuration.

Parameter	Value
Wheelbase (l_v)	3.6 m
Trailer length (l_t)	7.7 m
Hitch Length (l_h)	-0.6 m

l_h being negative indicates that the trailer is attached in front of the rear axle, defining the configuration as a Truck-Semitrailer (TS). The concept of differential flatness, previously discussed, allows the system's state to be computed from the state of the flat point. This point's motion, which dictates system behavior, is defined by the trajectory illustrated in section 3.5. Section 3.5 also demonstrates the orientation of the path, the path's curvature, and the rate of curvature change, respectively. These figures demonstrate that the chosen path has at least G^3 continuity. Figure 3.5 illustrates the initial state and the truck and trailer trajectory of the one-ST and one-GT system.

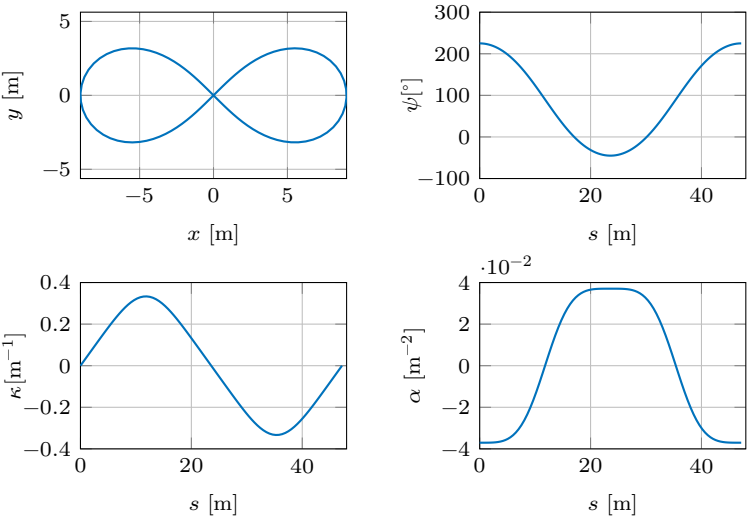


Figure 3.4.: Illustration of a continuous flat point trajectory meeting the minimum requirement of G^3 continuity. The figure includes all flat point states and the curvature's first derivative.

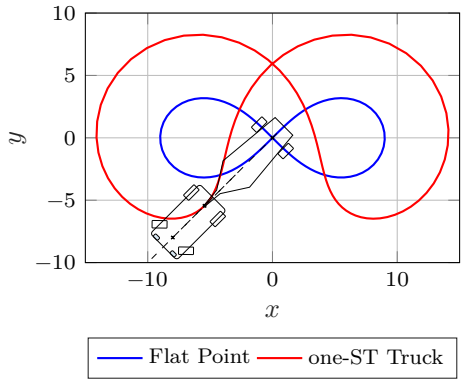


Figure 3.5.: Illustration highlighting the initial states of the system. The reverse driving maneuver is chosen due to the unstable dynamics encountered during reversing.

Figure 3.6 compares how each configuration, one-ST and one-GT, navigates the same trajectory. The reverse driving maneuver is chosen due to the unstable dynamics encountered during reversing. The analysis demonstrates the differences in path navigation between the one-ST and the one-GT configurations. For better visualization, the differences are illustrated in fig. 3.7.

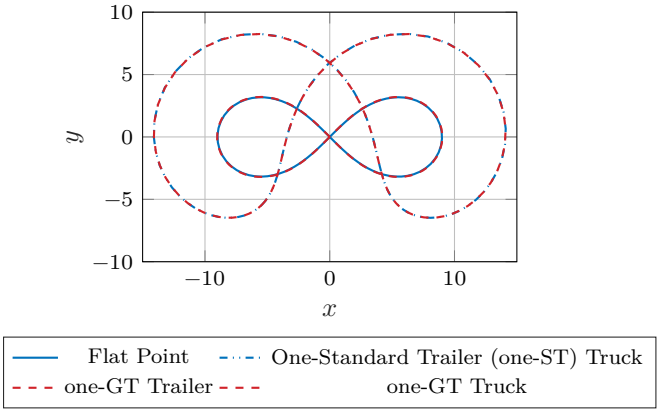
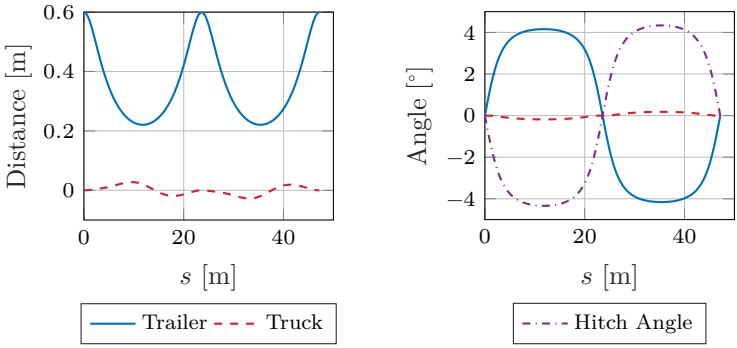


Figure 3.6.: Comparative path analysis of different truck and trailer configurations, illustrating how each configuration navigates the same flat point trajectory.

In Figure 3.7a, the Euclidean distance between the axle of the trailer of one-ST and one-GT and the distance between the truck's rear axle of the one-ST and one-GT configurations are illustrated. The distance of the trailer decreases as the path curvature increases. This behavior aligns with the flat model geometry illustrated in fig. 3.3.

If the one-GT is assumed to be a one-ST for simplicity, this difference in heading angles could pose significant challenges for collision-free path planning and effective control of the truck-trailer system. Additionally, Figure 3.7b demonstrates the heading difference between the trailer and

truck, as well as the hitch angle difference for one-ST and one-GT configurations. It can be noted that the heading of the trailer varies significantly from that of the flat point in the case of a one-GT configuration. The hitch and heading differences increase with curvature, which again can be deduced from the flat model geometry in fig. 3.3.



(a) Euclidean distance. **(b)** Heading difference.

Figure 3.7.: Comparative analysis of distance and angular differences for various trailer configurations. The distance figure demonstrates the variation between the rear axle of the trailer and truck for one-ST and one-GT configurations. The angular differences figure highlights the heading difference of the trailer and truck, as well as the hitch angle difference, which increase with curvature.

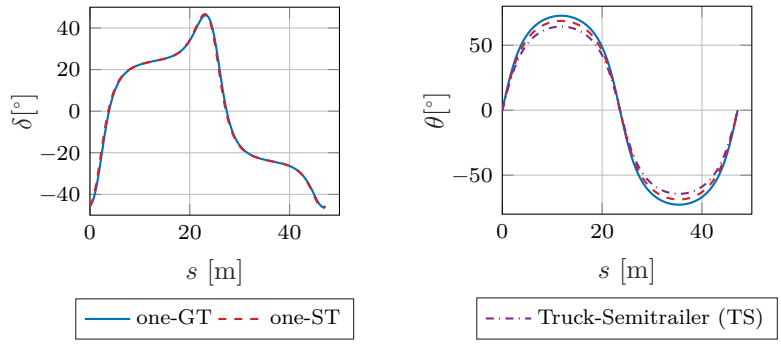
Approximating one-GT for significant coupling length with one-ST may simplify the initial design but can lead to inaccuracies during complex maneuvers, potentially compromising safety and operational efficiency. However, it is essential to note that the heading of the truck does not demonstrate much deviation, which might suggest that the primary control challenges and safety concerns are more pronounced in managing the trailer's dynamics than the truck's.

Figure 3.8a and fig. 3.8b provide further insights into the variations of steer and hitch angles for different trailer configurations. Notably,

fig. 3.7b reveals a significant difference in hitch angles between the one-ST and one-GT configurations. This difference in hitch angles is primarily due to the difference in the trailer's heading angle, as the truck's heading difference is not significant. The hitch angle difference will increase if the coupling point moves further from the truck's rear axle.

Figure 3.8b illustrates the effect of the hitching point on the required hitch angle for different trailer configurations to trace the same reference. The figure compares the hitch angle for a one-GT with a positive hitch length, a one-ST with zero hitch length, and a semitrailer with a negative hitch length.

It can be observed that the one-GT requires a higher hitch angle than the one-ST configuration. Conversely, the semitrailer with a negative hitch length requires a lower hitch angle than the one-ST.



(a) Steer angle. **(b)** Hitch angle.

Figure 3.8.: Comparative analysis of steer angle and hitch angle differences for various trailer configurations. The steer angle figure demonstrates a slight difference in slope between one-ST and one-GT configurations, while the hitch angle figure highlights the differences between one-GT, one-ST, and TS configurations.

These differences highlight the hitching point position's significant impact on the trailer systems' articulation and maneuverability. A negative hitch length in semitrailers enhances maneuverability by allowing tighter

turns with smaller hitch angles, thereby reducing the risk of jack-knifing. This makes controlling and navigating the trailer easier, especially in confined spaces. In the analysis of the truck-trailer system navigating a figure-of-eight trajectory, several key findings have been highlighted concerning the relationship between the heading of the trailer, hitch angles, and overall system behavior:

Difference in Heading and Hitch Angles: Despite notable differences in the heading of the trailer and hitch angles between configurations, the overall system behavior remains consistent for both configurations.

Continuity of Curvature and Rate of Change: The curvature of the path and the rate of change of curvature are both continuous. This continuity is reflected in the behavior of both the hitch and steering angles, which exhibit smooth transitions without abrupt changes. The findings can be generalized for an n-GT: to have a continuous steering angle, the curvature of the n^{th} trailer should be continuous up to the n^{th} degree [Rou93, Fli95].

In conclusion, whether a one-GT system can be approximated by a one-ST model depends on the application's precision requirements, the overall system architecture, and the dimensions of the system. While the approximation offers computational advantages, making it beneficial for general path-finding problems, there may be other options for achieving smooth and precise path-tracking control. This topic will be explored in detail further in the upcoming chapters.

3.6. Summary

This chapter explored the dynamics and modeling techniques of truck-trailer systems. It covered the modeling of n-GT systems, derived specific dynamics for one-GT systems, and introduced the concept of differential flatness, establishing its applicability to these systems. The chapter culminated with a comparative analysis of one-ST and one-GT systems.

Key Contributions:

- Developed a flat model for the one-ST system based on differential flatness.
- Derived a partial differential flatness for the one-GT system.
- Formulated a flat model for the one-GT system using partial differential flatness.

The flat models offer numerical stability, making them better for representing the system in model-based optimization and planning methods. The concept and the model derived in this section are utilized to formulate the trajectory planning, control, and optimization in chapter 5 and chapter 6. The differential flat model for one-ST is less complex than that for one-GT, and the same model can be extended to the n-ST system. It makes it suitable for computing initial solutions in high-dimensional, complex environments. The one-GT model can then be used to optimize the solution locally. Although this work focuses on the one-GT model and utilizes it directly, the one-ST model offers valuable advantages in specific scenarios.

4. Hitch Angle Stabilization for one-GT

Stability of the hitch angle is crucial for maintaining truck-trailer maneuverability, ensuring safety, and reducing the risk of jackknifing, especially during reverse maneuvers. This section provides further insights into the truck and trailer system dynamics concerning the hitch angle and stability.

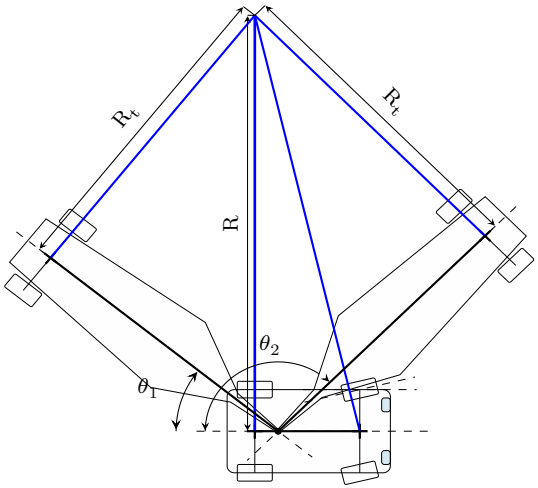


Figure 4.1.: Geometric illustration of steady-state cornering of a one-GT system, highlighting two equilibrium hitch angles, θ_1 and θ_2.

Consider a truck pulling a trailer; two equilibrium positions exist for the hitch angle, i.e., the rate of change of hitch angle is zero for a given radius

of rotation R of the truck. Figure 4.1 illustrates the two steady-state configurations of hitch angle θ_1 in the green zone and θ_2 in the red zone as shown in fig. 4.2. θ_2 is known as the jackknifing configuration[Jaz17]. Figure 4.2 depicts the case when the radius of rotation for the trailer becomes zero; here, the instantaneous center of rotation lies on the center of the axle.

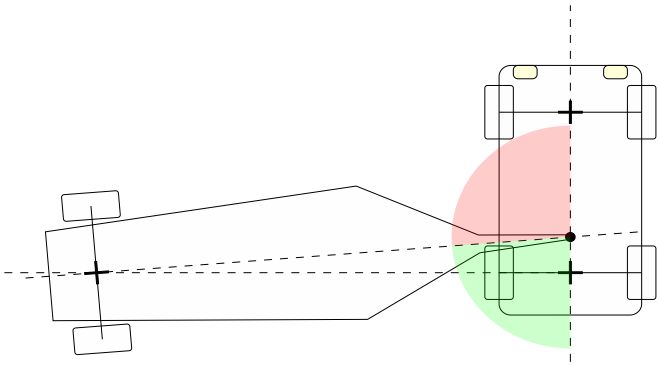


Figure 4.2.: Geometric illustration of a one-GT system where the radius of rotation for the trailer is zero, as the instantaneous center of rotation lies on the center of the trailer's axle. The hitch angle is divided into two zones: green and red. For simplification, the figure does not demonstrate right-hand turns, as they mirror the current figure.

In analyzing the stability of hitch angles during driving maneuvers, Table 4.1 categorizes the stability zones for both forward and backward driving scenarios. The table distinguishes between the Green and Red Zones of the hitch angles as depicted in Figure 4.2, indicating their stability conditions.

Table 4.1.: Hitch angle stability for forward and backward driving.

Hitch Angle	Forward Driving	Backward Driving
Green Zone	Stable	Unstable
Red Zone	Unstable	Stable

If the truck is moving forward and the truck-trailer combination is not jackknifed initially, it will remain unjackknifed. If the truck is moving backward, the truck-trailer combination has an unstable equilibrium point, i.e., the hitch angle lies in the green zone as shown in fig. 4.2, and any small perturbation in the hitch angle will result in jackknifing [Gon09]. Hence, for reversing a truck and trailer combination, trajectory stabilization needs to be focused on the trailer and hitch angle [Zan18]. Jazar provides an overview of trailer dynamics concerning hitch angle and system stability in [Jaz17].

Asymptotic stabilization of the hitch angle around the unstable equilibrium angle θ_1 is possible, as proven in [Jaz17, Lam03]. This chapter focuses on the control and stabilization of the hitch angle, presenting two control strategies for stabilizing the hitch angle around a reference angle, $\vartheta(t)$. Utilizing the principles of Lyapunov stability, a control scheme is formulated based on the Lyapunov function named LC. The LC control law is formulated using the concept of partial differential flatness, as detailed in Section 3.3.2. Additionally, a Linear Quadratic Regulator (LQR) control is presented for stabilizing the hitch angle around $\vartheta(t)$.

In conclusion, simulation results are presented, highlighting the advantages and drawbacks of all the developed controls to pinpoint the most effective hitch angle stabilization strategy for truck-trailer systems.

4.1. Lyapunov Control (LC)

The LC law has been extensively studied and applied in stabilizing the hitch angle in one-GT systems [Pra08, Pra07, Pra07]. By implementing an inner loop around the trailer hitch angle and treating the vehicle as a virtual articulated system, stable algorithms have been devised for trajectory stabilization [Pra08]. Moreover, feedback control strategies utilizing the LC law can effectively prevent trailer oscillations and instability during challenging maneuvers, such as extreme cornering [Zan18].

Furthermore, studies have highlighted the efficacy of Lyapunov techniques in stabilizing truck-trailer systems. For instance, research has addressed the asymptotic stabilization of tractor-trailer systems during various motions using Lyapunov methods [Ast04]. Additionally, the stability of control loops in autonomous truck and trailer systems has been validated through Lyapunov analysis, emphasizing the robustness of Lyapunov-based approaches in ensuring system stability [Rig20].

This section outlines the LC law for stabilizing the hitch angle in the one-GT system, as shown in fig. 3.2 and detailed in section 3.3.2.

Utilizing Lyapunov stability criteria as discussed in section 2.1.3, a Lyapunov function, $V(\theta) = \frac{1}{2}(\theta - \vartheta)^2$, is introduced to assess stability. Here, ϑ represents the reference angle, and θ signifies the hitch angle.
The system is stable within the region $\mathcal{D} = [\theta_{\min}, \theta_{\max}] \in \mathrm{R}$ if $V(\theta) > 0$ for $\theta \neq \vartheta$ and $V(\theta) = 0$ at the equilibrium point $\theta = \vartheta$.

To achieve asymptotic stability within $\mathcal{D}$, the derivative of $V(\theta)$, denoted as $V'(\theta)$, must be negative for all $\theta \neq \vartheta$, resulting in the control law for θ' as

$$\theta' = \vartheta' - K(\theta - \vartheta), \tag{4.1}$$

where K is a positive proportional control gain. The resulting $V'(\theta) = -K(\theta - \vartheta)^2$ is always negative for all positive K and $\theta \neq \vartheta$.

By correlating the hitch angle rates in eq. (4.1) and eq. (3.13), the steering angle input, δ, for the one-GT system is derived as follows:

$$\delta = \arctan\left(\frac{\mathrm{l_v}\sin\theta + \mathrm{sign}(\vec{v})\gamma \mathrm{l_t}\mathrm{l_v}\cos\theta}{\mathrm{l_t} + \mathrm{l_h}\cos\theta - \mathrm{sign}(\vec{v})\gamma \mathrm{l_t}\mathrm{l_h}\sin\theta}\right), \tag{4.2}$$

with $\gamma = \vartheta' - K(\theta - \vartheta)$, the control strategy dynamically adjusts for errors between the current and desired hitch angles. As the control error $(\theta - \vartheta)$ diminishes to zero, the control approach will track the hitch angle rate. Equation (4.2) aligns with the formulation in eq. (3.14), where the steering angle calculation leverages the partially differential flat system, as detailed in section 3.4. The derived law represents a feedback control on the hitch angle, i.e. the flat state of the partially differential flat system.

In a previous study [Kum22a], the control configuration is defined as SLC by adopting $\vartheta' = 0$, similar to the assumption made in [Pra07]. A key difference is that the control was developed for spatial stabilization of the hitch angle, whereas [Pra07] focused on stabilization in the temporal domain.

$$\gamma_{\mathrm{slc}} = -K(\theta - \vartheta).$$

This assumption simplifies the control strategy by considering the desired hitch angle as constant, proving advantageous for tracking trajectories where the hitch angle rate is either indeterminate or complex to compute.

4.2. Linear Quadratic Regulator (LQR)

This section introduces a Linear Quadratic Regulator (LQR) strategy for aligning the hitch angle θ with a desired reference ϑ, as explored in [Kum20]. Building on the system model in fig. 3.2, the hitch angle's

dynamics are described by the following equation:

$$\dot{\theta} = \frac{v \tan \delta}{\mathrm{l_v}} - \left(\frac{v \sin \theta}{\mathrm{l_t}} - \frac{\mathrm{l_h} v \tan \delta \cos \theta}{\mathrm{l_v l_t}} \right). \tag{4.3}$$

The steady-state relationship between steering angle δ_e and hitch angle θ_e is:

$$\delta_\mathrm{e} = \arctan \left[\frac{\mathrm{l_v} \sin \theta_\mathrm{e}}{\mathrm{l_h} \cos \theta_\mathrm{e} + \mathrm{l_t}} \right]. \tag{4.4}$$

Linearizing hitch angle dynamics around steady-state cornering yields:

$$\dot{\theta} = \mathrm{A}(\theta - \theta_\mathrm{e}) + \mathrm{B}(\delta - \delta_\mathrm{e}), \tag{4.5}$$

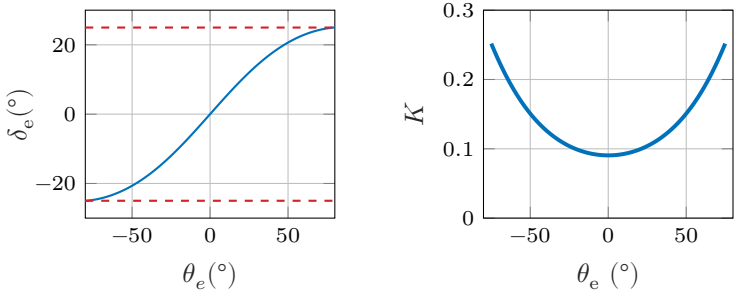
with

$$\mathrm{A} = -\frac{1}{\mathrm{l_t}} \left[\cos \theta_\mathrm{e} + \frac{\mathrm{l_h}}{\mathrm{l_v}} \sin \theta_\mathrm{e} \tan \delta_\mathrm{e} \right], \tag{4.6a}$$

$$\mathrm{B} = \frac{1}{\mathrm{l_v}} \left[1 + \frac{\mathrm{l_h}}{\mathrm{l_t}} \cos \theta_\mathrm{e} \right] (1 + \tan^2 \delta_\mathrm{e}). \tag{4.6b}$$

Given the reverse-driving instability of truck-semitrailer systems, limiting steering angles is crucial to prevent jack-knifing, with δ_max delineated as:

$$\delta_\mathrm{max} = \arctan \left[\frac{\mathrm{l_v}}{\sqrt{\mathrm{l_h}^2 + \mathrm{l_t}^2}} \right]. \tag{4.7}$$



(a) Figure illustrating the relationship between the Equilibrium Steer Angle δ_e, Maximum Steer Angle $\delta_{\max}$ against the Equilibrium Hitch Angle θ_e.

(b) Figure depicting the variation of the Linear Quadratic Regulator (LQR) control gain K concerning the equilibrium hitch angle θ_e for a range of hitch angles.

Figure 4.3.: Illustration of equilibrium steer angles and LQR control gain.

The LQR objective function, J, integrates deviations in θ and δ weighted by matrices Q and R:

$$J = \int_0^\infty (\Delta\theta^T Q \Delta\theta + \Delta\delta^T R \Delta\delta) dt, \tag{4.8}$$

leading to a feedback control input, $\mathbf{u}_{\text{lqr}} = -K(\theta - \theta_e)$, that minimizes J and stabilizes the hitch angle. K is the feedback gain matrix. K is plotted for selected combination of Q and R vs desired θ_e in fig. 4.3b.

The control input δ is the summation of the equilibrium steering wheel angle δ_e and the feedback control input $\mathbf{u}_{\text{lqr}}$:

$$\mathbf{u} = \delta = \delta_e + \mathbf{u}_{\text{lqr}} \tag{4.9}$$

4.3. Simulation Results and Discussions

In this section, the simulation aims to compare the developed control strategies: LQR, LC, and SLC. The focus is on assessing how each

strategy responds to different reference hitch angle profiles, specifically targeting hitch angle control. The simulation model is a simplified representation, without the inclusion of measurement noise, to isolate and evaluate the fundamental performance of each control method. The experiment uses step, ramp, and sinusoidal inputs as the reference hitch angle profiles. These profiles mimic real-world operational scenarios, comprehensively evaluating each control method's effectiveness under diverse, dynamic conditions.

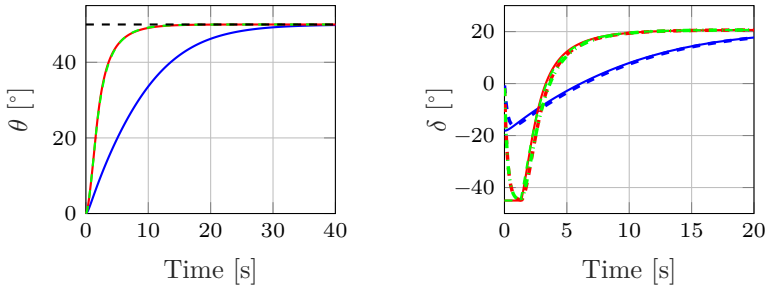
The simulation model extends the system defined in section 3.2 by incorporating steering angle dynamics modeled as a PT-1 system. The steering angle $\delta(t)$ is governed by the following equation:

$$\tau \frac{d\delta(t)}{dt} + \delta(t) = K\mathbf{u}(t) \tag{4.10}$$

The continuous system is integrated with a small simulation step of $1\,ms$ using the Euler discretization method. The truck's velocity is kept constant at $-1\,ms^{-1}$; negative velocity implies that the truck is driving in the reverse direction. The experimental design is structured to methodically apply each control strategy to the one-GT system, evaluating their performance across the three predefined input profiles. The step input tests the system's transient response and settling time, the ramp input examines its ability to track a continuously varying hitch angle, and the sinusoidal input assesses the control strategies' handling of dynamic, oscillatory inputs.

4.3.1. Step Input Response

Figure 4.4a illustrates the hitch angle response to a step change in the reference hitch angle. The figure highlights the transient behavior and settling time of the one-GT system under each control strategy.



(a) Hitch angle θ response to a step change in the reference hitch angle.

(b) Control input (- -) and steering angle δ following a step reference hitch angle.

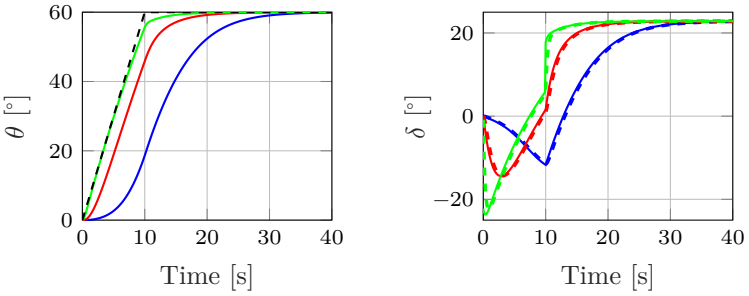
Figure 4.4.: Comparative analysis of hitch angle θ response and control input u for step reference hitch angle input under LQR (—), LC (—), and SLC (—) strategies in the one-GT system.

The LQR demonstrates a slower response than LC and SLC, with a noticeable delay in achieving the desired hitch angle. The LC achieves the fastest response, closely following the reference hitch angle with minimal overshoot. The Simiplified Lyapunov Control (SLC) is a simplified version of LC, where the rate of change of the reference hitch angle is considered zero. Therefore, in the case of a step response, LC and SLC perform identically.

4.3.2. Ramp Input Response

Figure 4.5a illustrates the hitch angle response to a ramp increase in the reference hitch angle. The figure highlights how well each control strategy tracks the changing reference.

The LQR has difficulty tracking the increasing reference hitch angle, resulting in a significant lag. The LC maintains a closer adherence to the reference hitch angle with minimal lag. SLC performs better than LQR in tracking the ramp input but demonstrates a moderate lag compared to LC.



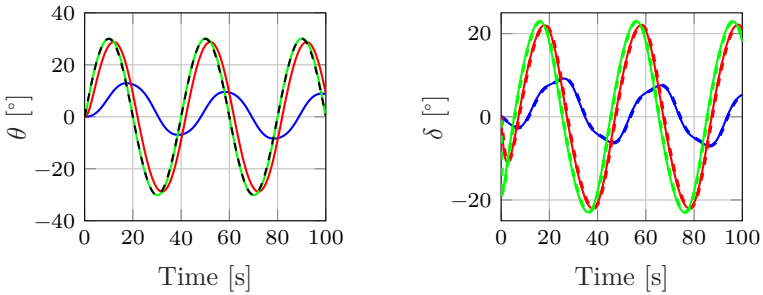
(a) Hitch angle response to a ramp-type increase in the reference hitch angle.

(b) Control input (- -) and steering angle delta behavior under a ramp reference hitch angle.

Figure 4.5.: Analysis of hitch angle response and control system behavior to ramp reference input LQR (—), LC (—), and SLC (—) strategies in the one-GT system.

4.3.3. Sinusoidal Input Response

Figure 4.6a compares the hitch angle response to a sinusoidal reference hitch angle input



(a) Hitch angle θ response for a sinusoidal reference hitch angle input.

(b) Control input (- -) and corresponding steering angle δ for a sinusoidal reference hitch angle input.

Figure 4.6.: Comparative analysis of hitch angle response and control input $\mathbf{u}$ for sinusoidal reference hitch angle input under LQR (—), LC (—), and SLC (—) strategies in the one-GT system.

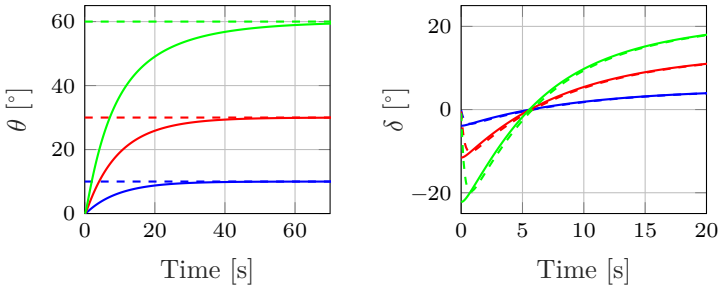
LQR fails to accurately follow the sinusoidal reference, exhibiting significant phase lag and amplitude error. LC accurately tracks the sinusoidal reference with minimal phase lag and amplitude error, demonstrating its proficiency in handling dynamic oscillatory inputs. SLC performs better than LQR but not as well as LC, showing moderate phase lag and amplitude error.

4.3.4. Evaluation of Lyapunov Control Performance

The system's performance under LC is evaluated using different input signals: step, ramp, and sinusoidal inputs with varying parameters. Each type of input tests the system's ability to maintain accurate hitch angle tracking and stable control efforts under varying conditions.

Figure 4.7a and fig. 4.7b depict the hitch angle response and control efforts for step inputs of varying heights. The results demonstrate the system adapts well to different step heights, demonstrating accurate hitch angle tracking and stable control efforts across all tested step inputs. The zoomed-in steering angle fig. 4.7b illustrates the dynamic phase from the initial condition where the steering angle is zero, providing a clear view of the control response.

For ramp inputs, fig. 4.8a and fig. 4.8b illustrate the system's response to ramps with varying slopes. LC effectively manages the increasing ramp rates, ensuring precise hitch angle tracking and stable control inputs. From fig. 4.8b, it is evident that there is a sudden jump in the requested steering wheel input when the hitch angle reaches a maximum ramp height. This sudden increase is due to the discontinuity in the rate of change of the hitch angle at this point, which also validates the findings and observations in section 3.5. This discontinuity is not visible in the step response request fig. 4.7b, as the rate of change of the hitch angle is continuous.



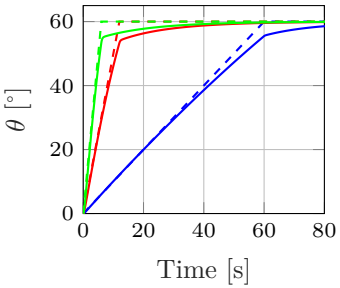
(a) Hitch angle responses to step inputs are depicted, where (- -) represents the reference hitch angle under LC.

(b) Control inputs and steering angle variations for step reference inputs under LC.

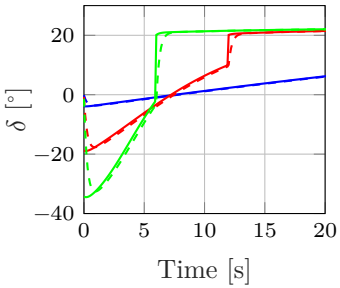
Figure 4.7.: Comparative analysis of the system's hitch angle and control input (- -) responses to step inputs under LC. The figure illustrates the performance at different step heights, from 10° (—), 30° (—) and 60° (—) , demonstrating the system's adaptability to sudden changes in the reference hitch angle.

Finally, the response to sinusoidal inputs with varying frequencies is shown in fig. 4.9a and fig. 4.9b. The control strategy demonstrates robustness in handling different frequencies, maintaining accurate hitch angle tracking, and consistent control inputs. This robustness is crucial for real-world applications where the system may encounter various dynamic conditions.

The results demonstrate that the control can effectively handle steady-state stabilization and dynamic trajectories where the hitch angle rate continuously varies. The simulation results confirm the assumptions that linearizing the system around the steady-state configuration for LQR limits its performance in following dynamic trajectories, as discussed in section 2.2 for the linearized system model in [Lju16, Lju20, Wu17].

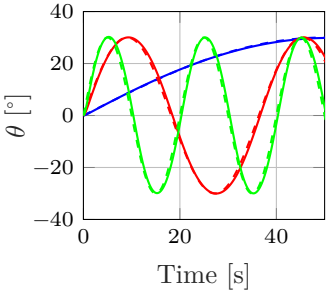


(a) Hitch angle responses to step inputs are depicted, where (- -) represent the reference hitch angle under LC.

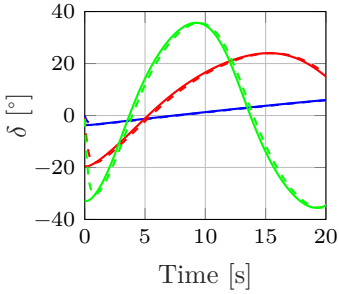


(b) Figure showcasing the system's control strategy for ramp inputs, comparing control input (- -) to the actual steering angles achieved.

Figure 4.8.: System response to ramp inputs: hitch angle and control input behavior under LC. Comparison of hitch angle responses to ramp inputs with rates from $1°s^{-1}$ (—), $5°s^{-1}$ (—) and $10°s^{-1}$ (—) , demonstrating system responsiveness.



(a) Hitch angle tracking performance for various sinusoidal inputs (- -) using LC.



(b) Figure showcasing the system's control strategy for sinusoidal inputs, comparing control input signals (- -) to the actual steering angles achieved.

Figure 4.9.: System response to sinusoidal inputs using LC: The frequency variation $1.8°s^{-1}$ (—), $9.9°s^{-1}$ (—) and $18°s^{-1}$ (—) challenges the control algorithm's ability accurately to maintain a dynamics hitch angle.

4.4. Summary

From the simulation results, it is evident that Lyapunov Control (LC) outperforms both the Linear Quadratic Regulator (LQR) and Simplified Lyapunov Control (SLC) in terms of response time, accuracy, and stability across all input profiles. LC consistently achieves the quickest response and shortest settling time for step inputs, the most effective tracking for ramp inputs, and the most accurate following for sinusoidal inputs with minimal phase lag and amplitude error. The findings make LC the most suitable control strategy for the one-GT system under diverse dynamic conditions.

SLC performs better than LQR but is less effective than LC, showing moderate stability and tracking ability. LQR, while functional, lags both LC and SLC in terms of response speed and tracking accuracy, indicating its limited applicability for scenarios requiring rapid and precise control adjustments. The related work conclusion also discusses the inability and limitation of the LQR to follow dynamic maneuvers.

Overall, the choice of control strategy should consider the specific requirements of the operational scenario, with LC being the preferred option for dynamic and precise hitch angle control in the one-GT system. This work focuses on developing an optimal trajectory tracking control for truck and trailer system. It uses hitch angle stabilization to compute the initial guess for the optimization problem and define a virtual system model. LC lays the foundation for the upcoming chapters of this work.

Additionally, a cascade control law Cascade Control consisting of a High-Level Pure Pursuit Control and a Low-Level Lyapunov Control (PPC-LC) explained in appendix A.1 from [Kum22a], where Pure Pursuit serves as the high-level control to compute the desired hitch angle and SLC is employed as the low-level control, is evaluated experimentally for trajectory tracking in section 5.4.

5. Nominally Guided MPC for one-GT

This chapter presents a Nominally Guided MPC (NGMPC) framework approach for nonlinear systems. This framework addresses the challenges inherent in nonlinear systems and optimization by formulating an efficient quadratic programming problem. One of the main objectives of this chapter is to formulate an optimal trajectory stabilization control for the one-GT system. The Nominally Guided MPC (NGMPC) framework is used for trajectory tracking control formulation of a one-GT system. Trajectory stabilization control ensures that the one-GT system accurately follows a predefined reference trajectory while maintaining stability and maneuverability. This control approach seeks to minimize deviations from the desired trajectory and to ensure smooth and efficient motion.

It is essential to highlight the distinction between optimal trajectory control and planning. Optimal trajectory control ensures the vehicle follows a given trajectory as precisely and efficiently as possible, considering dynamic constraints and control objectives. It involves real-time adjustments to the control inputs to minimize deviations from the desired parameters and optimize performance metrics such as smoothness and safety. Additionally, optimal trajectory control maintains the vehicle within a defined tolerance band around the reference trajectory, ensuring the system remains safe and stable despite minor deviations.

On the other hand, trajectory planning involves generating the desired trajectory for the vehicle to follow, typically considering the environment, obstacles, and desired final position. While trajectory planning

provides the reference trajectory, optimal trajectory control ensures that the vehicle adheres to this trajectory effectively.

The problem addressed in this chapter involves the stabilization of the system along the predefined trajectory. The state variables of the system, including the positions and orientations of both the truck and trailer, are assumed to remain within safe operational limits as long as they stay within the defined safe zone. This safe zone is characterized by a tolerance band around the reference trajectory, which accounts for minor deviations while ensuring the vehicle remains stable and secure. The control strategy keeps the vehicle's state variables within this tolerance band, maintaining safety and performance despite dynamic disturbances and uncertainties.

This chapter further explores three developed approaches for trajectory stabilization in one-GT systems:

- **Model Predictive Control (MPC)**: This approach utilizes the known dynamics of the one-GT system, based on the NGMPC framework.
- **Virtual Model Predictive Control (VMPC)**: This approach employs the same NGMPC framework but defines the model differently. VMPC uses a virtual model where the hitch angle is considered a virtual input, simplifying the control strategy.
- **Flatness-Based Model Predictive Control (FMPC)**: This approach leverages the concept of partial differential flatness, utilizing a partial flat model as the prediction model.

All approaches address the numerical instability of the one-GT system, particularly during reverse driving maneuvers. When the truck moves backward, it has an unstable equilibrium, and even a slight deviation can lead to jackknifing. An open-loop integration of the model with zero steering angle and zero hitch angle will inevitably result in jackknifing, even with minor numerical deviations. This inherent instability renders numerical open-loop forward integration impractical. The

NGMPC framework mitigates these issues by utilizing a nominal trajectory to guide the one-GT system to follow a reference trajectory. The nominal control strategies can effectively manage the inherent instability, ensuring stable and accurate trajectory tracking and serving as a linearization point for optimization problems. Additionally, Virtual Model Predictive Control (VMPC) and FMPC prediction models are inherently stable in both forward and reverse driving as compared to the classical one-GT system.

5.1. Nominally Guided MPC (NGMPC) Framework

This section presents the NGMPC framework formulated for the control and optimization of discrete nonlinear dynamic systems. Consider a discrete nonlinear system is given by the generic function f.

$$\mathbf{x}_{i+1} = f(\mathbf{x}_i, \mathbf{u}_i). \tag{5.1}$$

The function f is differentiable and lineariziable for all state $\mathbf{x} \in \mathbb{R}^{n\times 1}$ and input $\mathbf{u} \in \mathbb{R}^{m\times 1}$, where n and m denote the number of state variables and the number of control inputs, respectively.

Given the nonlinear system described in eq. (5.1), MPC can be formulated as either a Quadratic Programming (QP) or a Nonlinear Programming (NLP) problem. While QP solutions tend to be computationally more efficient than their NLP counterparts, this efficiency makes QP-MPC highly suitable for real-time applications requiring fast update rates [Zei11, Aug09].

However, the QP approach within MPC is fundamentally designed for linear systems. When addressing nonlinear systems, a prevalent method is to use linearized versions of the system dynamics. This linearization often remains accurate only near a specific operating point [Sim13]. Deviations from this point can introduce errors [Mor88]. Furthermore,

linearizing nonlinear constraints might result in approximations causing infeasible solutions, underlining the difficulty of capturing complex nonlinear system behaviors using this method [Par16].

To overcome the inherent limitations of traditional QP-based MPC with nonlinear systems [Sim13, Ali10], the concept of NGMPC is introduced. Within the framework of NGMPC, several key components that form the basis of the NGMPC framework are detailed.

- **Reference State:** At each time step $i \in [0, \mathrm{N}]$, where $\mathrm{N} \in \mathbb{Z}^+$ signifies the horizon length, the reference state $\mathbf{r}_i$ signifies the system's target or desired state.
- **Nominal State, Input, and Trajectory:** For each time step i in the prediction horizon $i \in [0,\mathrm{N}]$, the nominal state $\bar{\mathbf{x}}_i$ and the nominal input $\bar{\mathbf{u}}_i$ are established. These define the nominal trajectory, representing the system's expected progression. Starting from the initial condition $\bar{\mathbf{x}}_0 = \mathbf{x}_0$, where $\mathbf{x}_0$ represents the observed state at $i = 0$, the subsequent nominal states and control inputs are computed iteratively. This process involves integrating the system model over the prediction horizon, while employing a nominal control law to determine the nominal system inputs.
- **Optimal State, Input, and Trajectory:** For each time step within the prediction horizon $i \in [0,\mathrm{N}]$, the optimal state $\mathbf{x}_i$ and optimal input $\mathbf{u}_i$ are computed. They comprise the optimal trajectory, reflecting the most cost-efficient trajectory for the system, given specific state and input constraints.

One of the distinguishing features of the NGMPC framework is the integration of nominal control law based on classical control approaches and modern MPC. This framework employs a nominal control law iteratively, in conjunction with the system's nonlinear dynamics, to generate a nominal trajectory over the prediction horizon. Subsequently, the optimization problem is formulated as a QP, subject to the linearized system dynamics, with the generated nominal trajectory serving as the operating point.

This fusion of nominal control and MPC addresses many concerns of utilizing QP for formulating MPC for nonlinear systems. In typical scenarios, the linearized system dynamics are valid only within a narrow region around the operating point. However, the developed approach ensures that the nominal trajectory computed with nominal control acts as a suboptimal trajectory, thereby keeping the nominal trajectory within the vicinity of the optimal, allowing for a more accurate linearized model. It, in turn, facilitates the computation of feasible QP solutions within a limited number of iterations, rendering real-time implementation feasible. If the optimization fails to converge, the system will still nominally track the reference.

A step-by-step exploration of the NGMPC framework will now be presented to provide a comprehensive understanding of its core algorithms.

5.1.1. Linearization of System Dynamics

A linear system model is essential to formulate the QP problem required for NGMPC. Consequently, first-order Taylor approximation linearizes eq. (5.1) around the nominal state $\bar{\mathbf{x}}_i$ and the nominal input $\bar{\mathbf{u}}_i \ \forall \ i \in [0,\mathrm{N})$, where $\mathrm{N} \in \mathbb{Z}^+$ represents the prediction horizon of a RHC. The linearized system is as follows:

$$\begin{aligned} \mathbf{x}_{i+1} &= \bar{\mathbf{x}}_{i+1} + \mathrm{A}_i(\mathbf{x}_i - \bar{\mathbf{x}}_i) + \mathrm{B}_i(\mathbf{u}_i - \bar{\mathbf{u}}_i), && \text{(5.2a)} \\ &= \bar{\mathbf{x}}_{i+1} - \mathrm{A}_i\bar{\mathbf{x}}_i - \mathrm{B}_i\bar{\mathbf{u}}_i + \mathrm{A}_i\mathbf{x}_i + \mathrm{B}_i\mathbf{u}_i, && \text{(5.2b)} \\ \Leftrightarrow \mathbf{x}_{\mathrm{i+1}} &= \mathrm{A}_i\mathbf{x}_i + \mathrm{B}_i\mathbf{u}_i + \mathrm{I}\mathbf{z}_i, && \text{(5.2c)} \end{aligned}$$

where $\mathrm{I}_{\mathrm{n}\times\mathrm{n}}$ is an identity matrix with n is the number of states and

$$\begin{aligned} \bar{\mathbf{x}}_{i+1} &= f(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i), \\ \mathbf{z}_i &= \bar{\mathbf{x}}_{i+1} - \mathrm{A}_i\bar{\mathbf{x}}_i - \mathrm{B}_i\bar{\mathbf{u}}_i, \\ \mathrm{A}_i &= \left.\frac{\partial f}{\partial \mathbf{x}_i}\right|_{\mathbf{x}_i=\bar{\mathbf{x}}_i} \quad \& \quad \mathrm{B}_i = \left.\frac{\partial f}{\partial \mathbf{u}_i}\right|_{\mathbf{u}_i=\bar{\mathbf{u}}_i}. \end{aligned}$$

$A_i \in \mathbb{R}^{n \times n}$ and $B_i \in \mathbb{R}^{n \times m}$ are the Jacobian matrices w.r.t. $\mathbf{x}_i$ and $\mathbf{u}_i$ at $\bar{\mathbf{x}}_i$ and $\bar{\mathbf{u}}_i$.

Equation (5.2) defines the linearized model for the dynamics of a nonlinear system around a nominal trajectory. This linearized model is a valuable approximation that provides a highly accurate representation of the system's behavior in the vicinity of the nominal state and input $[\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i]$.

5.1.2. Formulation of the Objective Function

The objective function in NGMPC addresses a primary challenge in control systems: guiding a dynamic system to follow a specified reference trajectory. This function in NGMPC has three core aims:

1 **Trajectory Tracking Precision:** This is achieved by penalizing deviations from the target reference states.

2 **Control Smoothness:** Emphasized by penalties on sudden or abrupt changes between consecutive control inputs and deviations from reference control inputs.

3 **Prediction Model Accuracy:** Assured by penalizing deviation from nominal control sequences, ensuring the linearized model remains near its operational point.

By penalizing deviations from the nominal trajectory, NGMPC ensures that the system behavior remains near this trajectory. In doing so, precision control is achieved by leveraging the nominal control law. Furthermore, this strategy compels the system to operate within regions where the linearized model remains valid, subsequently enhancing the feasibility of the optimal trajectory.

After defining the goals and concepts of the NGMPC objective function, it's essential to detail its mathematical formulation to see how these principles are applied.

The objective function J is written as:

$$\begin{aligned} J(\mathbf{x}_i,\mathbf{u}_i) = \frac{1}{2}\sum_{i=0}^{\mathrm{N}-1}\Big(& (\mathbf{x}_{i+1}-\mathbf{r}_{i+1})^{\mathrm{T}}\mathrm{Q}_{i+1}(\mathbf{x}_{i+1}-\mathbf{r}_{i+1}) \\ & + (\mathbf{u}_i-\mathbf{u}_{\mathrm{r},i})^{\mathrm{T}}\mathrm{R}_i(\mathbf{u}_i-\mathbf{u}_{\mathrm{r},i}) \\ & + (\mathbf{u}_i-\bar{\mathbf{u}}_i)^{\mathrm{T}}\mathrm{P}_i(\mathbf{u}_i-\bar{\mathbf{u}}_i) \\ & + (\mathbf{u}_{i-1}-\mathbf{u}_i)^{\mathrm{T}}\mathrm{S}_i(\mathbf{u}_{i-1}-\mathbf{u}_i)\Big), \end{aligned} \tag{5.3}$$

Where:

- $\mathbf{r}_i \in \mathbb{R}^{\mathrm{n}\times 1} \forall i \in [0,\mathrm{N}]$ are the target reference state.
- $\mathrm{Q} \in \mathbb{R}^{\mathrm{n}\times\mathrm{n}}$ is the cost matrix associated with deviations in state variables.
- $\mathrm{R} \in \mathbb{R}^{\mathrm{m}\times\mathrm{m}}$ penalizes deviations of control inputs from the reference.
- $\mathrm{P} \in \mathbb{R}^{\mathrm{m}\times\mathrm{m}}$ penalizes deviations of control inputs from their nominal values.
- $\mathrm{S} \in \mathbb{R}^{\mathrm{m}\times\mathrm{m}}$ penalizes differences between consecutive control actions to ensure smoothness.

5.1.3. Quadratic Optimization Problem

NGMPC aims to minimize the objective function (5.3), by adhering to the system dynamics (5.2) and determining the optimal control sequence $\boldsymbol{u}^*$. This section provides a comprehensive derivation of the quadratic optimization problem, focusing on minimizing the objective function subjective to the system's dynamics and considering the state and input constraints.

For all $i \in [0,\mathrm{N})$, the system takes the following vector form:

$$\boldsymbol{x} = \mathcal{A}\mathbf{x}_0 + \mathcal{B}\boldsymbol{u} + \mathcal{E}\boldsymbol{z}, \tag{5.4}$$

where $\mathbf{x}_0 \in \mathbb{R}^{n\times 1}$ is the current measured system state.

The sequences for control input, state, and equivalent disturbance are represented by the vectors $\boldsymbol{u}$, $\boldsymbol{x}$, and $\boldsymbol{z}$, respectively:

- $\boldsymbol{u} \in \mathbb{R}^{mN\times 1}$: The control input sequence, represented as

$$\boldsymbol{u} = [\mathbf{u}_0^\top, \mathbf{u}_1^\top, \dots, \mathbf{u}_{N-2}^\top, \mathbf{u}_{N-1}^\top]^\top.$$

- $\boldsymbol{x} \in \mathbb{R}^{nN\times 1}$: The state sequence, represented as

$$\boldsymbol{x} = [\mathbf{x}_1^\top, \dots, \mathbf{x}_{N-1}^\top, \mathbf{x}_N^\top]^\top.$$

- $\boldsymbol{z} \in \mathbb{R}^{nN\times 1}$: The equivalent disturbance sequence, represented as

$$\boldsymbol{z} = [\mathbf{z}_0^\top, \dots, \mathbf{z}_{N-2}^\top, \mathbf{z}_{N-1}^\top]^\top.$$

The matrices $\mathcal{A} \in \mathbb{R}^{nN\times n}$, $\mathcal{B} \in \mathbb{R}^{nN\times mN}$, and $\mathcal{E} \in \mathbb{R}^{nN\times nN}$ are state matrices, input matrices, and disturbance matrices for the equivalent linear system(5.4), respectively.

$$\mathcal{A} = \left[\mathrm{A}_0; \mathrm{A}_0\mathrm{A}_1; \dots, \mathrm{A}_0\mathrm{A}_1 \dots \mathrm{A}_{N-2}\mathrm{A}_{N-1}\right], \tag{5.5}$$

$$\mathcal{B} = \begin{bmatrix} \mathrm{B}_0 & & & & \\ \mathrm{A}_1\mathrm{B}_0 & \mathrm{B}_1 & & & \\ \mathrm{A}_1\mathrm{A}_2\mathrm{B}_0 & \mathrm{A}_2\mathrm{B}_1 & \ddots & & \\ \vdots & & & \ddots & \\ \mathrm{A}_1\mathrm{A}_2 \dots \mathrm{A}_{N-1}\mathrm{B}_0 & & \dots & & \mathrm{B}_{N-1} \end{bmatrix}, \tag{5.6}$$

$$\mathcal{E} = \begin{bmatrix} \mathrm{I} & & & \\ \mathrm{A}_1\mathrm{I} & \mathrm{I} & & \\ \mathrm{A}_1\mathrm{A}_2\mathrm{I} & & \ddots & \\ \vdots & & & \ddots \\ \mathrm{A}_1\mathrm{A}_2 \dots \mathrm{A}_{N-1}\mathrm{I} & & \dots & \mathrm{I} \end{bmatrix}. \tag{5.7}$$

The objective function given by eq. (5.3), similar to the system dynamics, can be expressed in its vector form as follows:

$$\begin{aligned} J(\boldsymbol{x}, \boldsymbol{u}) = &(\boldsymbol{x} - \boldsymbol{r})^\top \mathcal{Q} (\boldsymbol{x} - \boldsymbol{r}) \\ &+ (\boldsymbol{u} - \boldsymbol{u}_\mathrm{r})^\top \mathcal{R} (\boldsymbol{u} - \boldsymbol{u}_\mathrm{r}) \\ &+ (\boldsymbol{u} - \bar{\boldsymbol{u}})^\top \mathcal{P} (\boldsymbol{u} - \bar{\boldsymbol{u}}) \\ &+ (\mathbf{u}_{-1}\mathcal{D} + \mathcal{D}_1 \boldsymbol{u})^\top \mathcal{S} (\mathbf{u}_{-1}\mathcal{D} + \mathcal{D}_1 \boldsymbol{u}). \end{aligned} \tag{5.8}$$

The various vectors in eq. (5.3) and their corresponding definitions are given below:

- $\boldsymbol{r} \in \mathbb{R}^{\mathrm{nN}\times 1}$: The desired reference state for the system over the prediction horizon. It is represented as

$$\boldsymbol{r} = [\mathbf{r}_1^{\mathrm{T}}, \dots, \mathbf{r}_{\mathrm{N}-1}^{\mathrm{T}}, \mathbf{r}_{\mathrm{N}}^{\mathrm{T}}]^{\mathrm{T}}.$$

- $\boldsymbol{u}_\mathrm{r} \in \mathbb{R}^{\mathrm{mN}\times 1}$: The reference control input indicating the ideal control action. It is represented as

$$\boldsymbol{u}_\mathrm{r} = [\mathbf{u}_{\mathrm{r},0}{}^{\mathrm{T}}, \mathbf{u}_{\mathrm{r},1}{}^{\mathrm{T}}, \dots, \mathbf{u}_{\mathrm{r,N}-2}{}^{\mathrm{T}}, \mathbf{u}_{\mathrm{r,N}-1}{}^{\mathrm{T}}]^{\mathrm{T}}.$$

- $\mathbf{u}_{-1} \in \mathbb{R}^{\mathrm{m}\times 1}$: The current measured system input.
- $\bar{\boldsymbol{u}} \in \mathbb{R}^{\mathrm{mN}\times 1}$: The nominal input vector, computed using 1. It is represented as

$$\bar{\boldsymbol{u}} = [\bar{\mathbf{u}}_0^{\mathrm{T}}, \bar{\mathbf{u}}_1^{\mathrm{T}}, \dots, \bar{\mathbf{u}}_{\mathrm{N}-2}^{\mathrm{T}}, \bar{\mathbf{u}}_{\mathrm{N}-1}^{\mathrm{T}}]^{\mathrm{T}}.$$

The matrices $\mathcal{Q}$, $\mathcal{R}$, and $\mathcal{S}$ are the equivalent cost matrices for the vector form:

$$\begin{aligned}
\mathcal{Q} &= \mathrm{diag}(\mathrm{Q}_1,\mathrm{Q}_2,\dots,\mathrm{Q}_\mathrm{N}) && \in \mathbb{R}^{\mathrm{nN}\times\mathrm{nN}},\\
\mathcal{R} &= \mathrm{diag}(\mathrm{R}_0,\mathrm{R}_1,\dots,\mathrm{R}_{\mathrm{N}-1}) && \in \mathbb{R}^{\mathrm{mN}\times\mathrm{mN}},\\
\mathcal{P} &= \mathrm{diag}(\mathrm{P}_0,\mathrm{P}_1,\dots,\mathrm{P}_{\mathrm{N}-1}) && \in \mathbb{R}^{\mathrm{mN}\times\mathrm{mN}},\\
\mathcal{S} &= \mathrm{diag}(\mathrm{S}_0,\mathrm{S}_1,\dots,\mathrm{S}_{\mathrm{N}-1}) && \in \mathbb{R}^{\mathrm{mN}\times\mathrm{mN}}.
\end{aligned}$$

Additionally, the matrices $\mathcal{D}$ and $\mathcal{D}_1$ compute the control input deviation from the subsequent step, are defined as follows:

$$\mathcal{D} = [[1,\dots,1]_{\mathrm{m}\times 1}, 0, \dots, 0]^{\mathrm{T}} \qquad \in \mathbb{R}^{\mathrm{mN}\times 1},$$

$$\mathcal{D}_1 = \begin{bmatrix}
-\mathrm{I}_\mathrm{m} & 0 & 0 & \dots & \dots & 0\\
0 & \mathrm{I}_\mathrm{m} & \mathrm{I}_\mathrm{m} & 0 & \dots & 0\\
\vdots & \vdots & \ddots & \ddots & \vdots & \vdots\\
0 & 0 & \ddots & \ddots & 0 & \vdots\\
0 & \dots & \dots & \dots & \mathrm{I}_\mathrm{m} & -\mathrm{I}_\mathrm{m}
\end{bmatrix} \qquad \in \mathbb{R}^{\mathrm{mN}\times\mathrm{mN}}.$$

By substituting the state and input matrices eq. (5.4) into the cost function eq. (5.8), the expression becomes more concise in its quadratic form.

$$J(\boldsymbol{u}) = \boldsymbol{u}^\top \mathcal{H}\boldsymbol{u} + g\boldsymbol{u}, \tag{5.9}$$

where,

$$\begin{aligned}
\mathcal{H} &= \mathcal{B}^\top \mathcal{Q}\mathcal{B} + \mathcal{R} + \mathcal{P} + \mathcal{D}_1^\top \mathcal{S}\mathcal{D}_1\\
&\quad \in \mathbb{R}^{\mathrm{mN}\times\mathrm{mN}},\\
g &= 2(\mathbf{x}_0^{\mathrm{T}}\mathcal{A}^{\mathrm{T}}\mathcal{Q}\mathcal{B} + \boldsymbol{z}^{\mathrm{T}}\mathcal{E}^{\mathrm{T}}\mathcal{Q}\mathcal{B} - \boldsymbol{r}^{\mathrm{T}}\mathcal{Q}\mathcal{B} - \boldsymbol{u}_{\mathrm{r}}^{\mathrm{T}}\mathcal{R} + \boldsymbol{u}_{-1}^{\mathrm{T}}\mathcal{D}^{\mathrm{T}}\mathcal{S}\mathcal{D}_1)\\
&\quad \in \mathbb{R}^{\mathrm{mN}\times 1}.
\end{aligned} \tag{5.10}$$

The objective is to minimize the cost function $J(\boldsymbol{u})$ while respecting various constraints. By leveraging the objective function presented in eq. (5.9) and the system equation expressed in vector form eq. (5.4),

the generic quadratic programming (QP) problem for NGMPC can be derived as follows:

$$\boldsymbol{u}^* = \underset{\boldsymbol{u},\boldsymbol{e}_\mathrm{s}}{\arg\min}\, J(\boldsymbol{u}) + \boldsymbol{e}_\mathrm{s}^\mathrm{T} \mathcal{Q}_\mathrm{s} \boldsymbol{e}_\mathrm{s} \tag{5.11}$$

$$\text{s.t.} \quad \boldsymbol{x} = \mathcal{A}\mathbf{x}_0 + \mathcal{B}\boldsymbol{u} + \mathcal{E}\boldsymbol{z}, \tag{5.12}$$

$$\boldsymbol{e}_{\min} - \mathcal{J}\boldsymbol{e}_\mathrm{s} < \boldsymbol{x} - \boldsymbol{r} < \boldsymbol{e}_{\max} + \mathcal{J}\boldsymbol{e}_\mathrm{s}, \tag{5.13}$$

$$\boldsymbol{y}_{\min} < \mathcal{C}\boldsymbol{x} < \boldsymbol{y}_{\max}, \tag{5.14}$$

$$\boldsymbol{u}_{\min} < \boldsymbol{u} < \boldsymbol{u}_{\max}, \tag{5.15}$$

$$\Delta\boldsymbol{u}_{\min} < \mathcal{D}\mathbf{u}_{-1} + \mathcal{D}_1\boldsymbol{u} < \Delta\boldsymbol{u}_{\max}, \tag{5.16}$$

where $\mathcal{J} = [\mathrm{I_n}, \ldots\ldots\ldots, \mathrm{I_n}]^\mathrm{T} \in \mathbb{R}^{\mathrm{nN}\times\mathrm{n}}$, I is identity matrix.

The constraints described by Equation (5.13) ensure minimal system deviation from its reference. Integral to this is the slack vector $\boldsymbol{e}_\mathrm{s} \in \mathbb{R}^{\mathrm{n}\times 1}$, which provides flexibility, allowing for minor deviations in the face of measurement inaccuracies, uncertainties, disturbances, or modeling inaccuracies. The slack variables ensure the feasibility of the optimization, acknowledging real-world complexities that might impede perfect state tracking.

Output constraints are specified by eq. (5.14), with $\mathcal{C}$ as the output matrix, while eq. (5.15) and eq. (5.16) represent the limits for the control inputs. The optimization problem is solved using a Quadratic Programming (QP) solver. The cvxgen tool was used to generate the quadratic programming solvers for the developed approaches [Mat12].

In the subsequent sections, NGMPC framework will be employed to develop optimal trajectory stabilization controls for a one-GT system.

5.2. Optimal Trajectory Stabilization for one-GT

Optimal trajectory stabilization for truck and trailer systems ensures safe and efficient navigation, particularly in complex driving scenarios such as reversing or maneuvering in tight spaces. Achieving this involves accurately tracking a desired reference trajectory while maintaining system stability and minimizing control efforts. The objective function determines the system's dynamic behavior in optimal trajectory control. Considering control objectives and desired system responses directs the optimization toward determining optimal control inputs. For instance, emphasizing precision ensures precise tracking of a reference trajectory, while prioritizing control smoothness can minimize actuation energy expenditure, reduce abrupt changes, and provide system stability. A well-formulated objective function must balance these performance metrics to ensure optimal system performance.

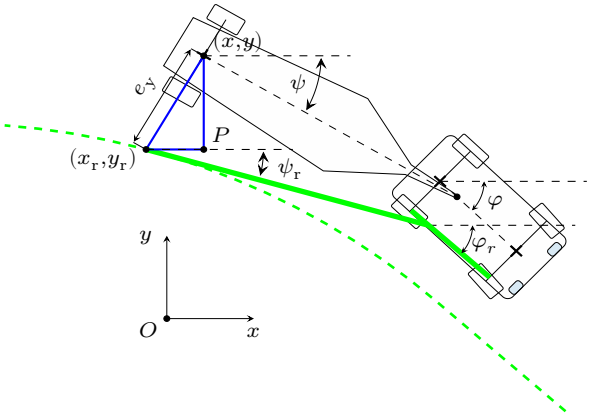


Figure 5.1.: The figure depicts the nominal and the corresponding reference state (x, y, ψ, φ) for an one-GT. The subscript $_\mathrm{r}$ denotes the reference state. The dotted green line depicts the reference trajectory, and the solid green lines illustrate the reference configuration of the truck and trailer system.

Figure 5.1 provides a geometric illustration of the corresponding reference state for a system state in a one-GT. In this figure, x, y, ψ, φ, and δ denote the system states, while x_r, y_r, ψ_r, φ_r, and δ_r represent the reference states. The dotted green line indicates the trajectory the system aims to follow, and the solid green lines depict the reference configuration of the truck and trailer. Here, the states $x, y, \psi, \varphi, \theta$ and δ represent the position, orientations of the trailer and truck, hitch angle, and steering angle, respectively.
Trajectory tracking aims to ensure that the one-GT system follows the desired reference trajectory and remains within an allowed region. Hence, the precision metrics and the constraints to keep the tracking errors within a defined tolerance band for the one-GT system are formulated here:

1 **Lateral Deviation (e_y)**: It quantifies the orthogonal distance between the trailer's position and the desired reference trajectory. It is critical for avoiding off-track behaviors and ensuring safe navigation. The coordinates x_r and y_r denote the intersection points of the reference trajectory with the trailer's lateral axis.

$$\begin{aligned} e_\mathrm{y} &= \sqrt{(x - x_\mathrm{r})^2 + (y - y_\mathrm{r})^2} \\ \text{s.t.} \quad & x - x_\mathrm{r} \in [e_{\mathrm{x,min}}, e_{\mathrm{x,max}}], \\ & y - y_\mathrm{r} \in [e_{\mathrm{y,min}}, e_{\mathrm{y,max}}]. \end{aligned}$$

2 **Trailer Heading Deviation (e_ψ)**: It depicts the angular deviation between the trailer's heading and the desired reference heading.

$$e_\psi = \psi - \psi_\mathrm{r} \quad \text{s.t.} \quad e_\psi \in [e_{\psi,\mathrm{min}}, e_{\psi,\mathrm{max}}].$$

3 **Truck Heading Deviation (e_φ)**: It quantifies the angular difference between the truck's heading and its target reference.

$$e_\varphi = \varphi - \varphi_\mathrm{r} \quad \text{s.t.} \quad e_\varphi \in [e_{\varphi,\mathrm{min}}, e_{\varphi,\mathrm{max}}]. \tag{5.17}$$

4 **Steering Angle Deviation from Reference ($\delta - \delta_{\mathbf{r}}$)**: It measures the difference between the actual steering angle and its reference, promoting alignment with the desired control actions.

$$e_\delta = \delta - \delta_{\mathrm{r}} \quad \text{s.t.} \quad e_\delta \in [e_{\delta,\min}, e_{\delta,\max}]$$

5 **Steering & Hitch Angle Constraints:** These constraints are crucial to maintain actuation limits and prevent jackknifing.

$$\delta \in [\delta_{\min}, \delta_{\max}],$$
$$\theta \in [\theta_{\min}, \theta_{\max}].$$

There are three main approaches developed for trajectory stabilization of one-GT: MPC, VMPC, and FMPC. MPC utilizes a classical system model as the prediction model. The overall goal of all three controls remains the same, as discussed in this section. While the precision metrics are the same for all three developed approaches, the smoothness metrics differ slightly in formulation but aim to achieve the same objective. All three control approaches utilize the NGMPC framework established in section 5.1.

5.2.1. Model Predictive Control (MPC)

This section explores the formulation of MPC for optimal trajectory tracking of one-GT. MPC uses a classical one-GT system dynamic model as the prediction model and trajectory tracking control utilizes the NGMPC framework. The discrete spatial dynamics of the one-GT system, derived from the continuous dynamics of the general one-trailer system as detailed in eq. (3.6) from section 3.2, are obtained using the

Euler discretization method:

$$x_{k+1} = x_k + \mathrm{ds}\cos\psi_k, \tag{5.18a}$$

$$y_{k+1} = y_k + \mathrm{ds}\sin\psi_k, \tag{5.18b}$$

$$\psi_{k+1} = \psi_k + \frac{\mathrm{ds}\,(\mathrm{l_v}\sin\theta_k - \mathrm{l_h}\cos\theta_k \tan\delta_k)}{\mathrm{l_t}\,(\mathrm{l_v}\cos\theta_k + \mathrm{l_h}\sin\theta_k \tan\delta_k)}, \tag{5.18c}$$

$$\varphi_{k+1} = \varphi_k + \frac{\mathrm{ds}\tan\delta_k}{\mathrm{l_v}\cos\theta_k + \mathrm{l_h}\sin\theta_k \tan\delta_k}. \tag{5.18d}$$

Section 3.2 also explains that the state space $\mathbf{x} = [x, y, \psi, \varphi]$ is sufficient to compute all the states of the truck and trailer. The system dynamics in eq. (5.18) can be written as:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \tag{5.19}$$

where the state vector $\mathbf{x}_k \in \mathbb{R}^{4\times 1}$ consists of $[x_k, y_k, \psi_k, \varphi_k]^{\mathrm{T}}$, and the input vector $\mathbf{u}_k \in \mathbb{R}^{1\times 1}$ is $[\delta_k]^{\mathrm{T}}$.

Although the Euler method is computationally efficient, it may introduce discretization errors, especially for larger discrete steps, and sometimes exhibit instability. To address this, the prediction model for the trailer's dynamics is based on a multi-step Euler integration method, where each prediction step is divided into $\mathrm{n_d}$ smaller steps to enhance accuracy and stability. The distance traveled by the trailer between consecutive prediction points, denoted as $\mathrm{ds_{mpc}}$, is given by $\mathrm{ds_{mpc}} = \mathrm{n_d} \cdot \mathrm{ds}$, with $\mathrm{n_d}$ representing the number of discretization steps between prediction steps i and $i+1$. Within each prediction step i, the system input $\mathbf{u}_i = \delta_i$ remains constant.

Starting from the current state $\mathbf{x}_i$, the model iteratively updates the system state using the control input δ_i across the discretization steps, resulting in the predicted state $\mathbf{x}_{i+1}$. This iterative process is encapsulated in the nonlinear function f_{eq}, which defines the relationship $\mathbf{x}_{i+1} = f_{\mathrm{eq}}(\mathbf{x}_i, \mathbf{u}_i)$. By utilizing smaller integration steps, the algorithm

effectively reduces discretization errors commonly associated with larger prediction intervals.

The equivalent function f_{eq} are formulated using the Casadi toolbox [And19]. This symbolic framework allows for efficient computation of the function f_{eq} and its derivatives with respect to $\mathbf{x}_i$ and $\mathbf{u}_i$, enabling fast and accurate analytical Jacobian calculations.

Nominal Trajectory Generation

The cascade control law presented in [Kum22a] is used as the nominal control law, refer appendix A.1 for detail. The algorithm for the nominal trajectory generation is outlined in Algorithm 1. This algorithm establishes the initial nominal state as the current state and iteratively computes the desired curvature using a pure pursuit control law. The desired hitch angle is calculated, followed by determining the desired steering angle using a Simplified Lyapunov control function explained in section 4.1. The reference state vector $\mathbf{r}_i \in \mathbb{R}^{4\times 1}$ is composed of $[x_{\text{i,r}}, y_{\text{i,r}}, \psi_{\text{i,r}}, \varphi_{\text{i,r}}]^{\text{T}}$. The nearest reference state is interpolated using the reference trajectory, and the current nominal state and the next nominal state are computed using system dynamics. The nominal state and input trajectories are updated for each step of the prediction horizon. The reference states are computed using a cubic interpolation of the nearest reference points. The coordinate system is translated to the current trailer frame at each control step to ensure the optimization problem and reference matching are more feasible. The origin of the coordinate system O is defined by the position of the trailer's axle center, x_0 and y_0, and rotated to align with the trailer's longitudinal axis, denoted by the heading angle ψ_0. This translation and rotation of the coordinate system facilitate a more accurate and efficient reference state calculation.

Algorithm 1: Nominal Trajectory Generation

Parameters:

- N: Prediction horizon
- $\mathrm{l_t}$: Trailer length
- $\mathrm{N_r}$: Reference Horizon Length

Input:

- $\mathbf{x}_0$: Initial system state
- $\{\mathbf{r}_k\}_{k=1}^{\mathrm{N_r}}$: Reference trajectory with $\mathrm{N_r}$ number of reference state
- f_{Lyapunov}: Lyapunov control function

Output:

- $\{\bar{\mathbf{x}}_i\}_{i=0}^{N}$: Nominal state trajectory
- $\{\bar{\mathbf{u}}_i\}_{i=0}^{N-1}$: Nominal input trajectory

Initialization:

$\bar{\mathbf{x}}_0 \leftarrow \mathbf{x}_0$

$\theta_{\mathrm{current}} \leftarrow \theta_0$

Begin Algorithm:

for $i = 0$ *to* $\mathrm{N} - 1$ **do**

$\kappa_{\mathrm{desired}} \leftarrow \mathrm{PurePursuitControl}(r_{k\,k=1}^{\mathrm{N_r}}, \bar{\mathbf{x}}_i)$

$\theta_{\mathrm{desired}} \leftarrow \arctan(\kappa_{\mathrm{desired}} \mathrm{l_t})$

$\delta_{\mathrm{desired}} \leftarrow f_{\mathrm{Lyapunov}}(\theta_{\mathrm{desired}}, \theta_{\mathrm{current}})$

$\bar{\mathbf{x}}_{i+1} \leftarrow f_{\mathrm{eq}}(\bar{\mathbf{x}}_i, \delta_{\mathrm{desired}})$

$\mathbf{r}_{i+1} \leftarrow \mathrm{MatchReferenceState}(\{\mathbf{r}_k\}_{k=1}^{\mathrm{N_r}}, \bar{\mathbf{x}}_{i+1})$

$\bar{\mathbf{u}}_i \leftarrow \delta_{\mathrm{desired}}$

$\theta_{\mathrm{current}} \leftarrow g(\bar{\mathbf{x}}_{i+1})$

Objective Function and Optimization Problem

Along with the performance metrics defined in section 5.2, additional metrics focus on the continuity and consistency of control actions. It ensures a control strategy that reduces abrupt changes and maintains system stability. These additional metrics include:

1 **Steering Angle Deviation from Nominal** ($\delta - \bar{\delta}$): Ensures the actual steering angle remains close to a computed nominal value, ensuring accurate linearized model behavior in the vicinity of the nominal trajectory explained in section 5.1.

2 **Steering Angle Rate** ($\delta_i - \delta_{\mathrm{i}-1}$): Captures changes in steering angle over successive time steps, promoting smooth transitions and avoiding abrupt maneuvers.

By balancing these precision and additional metrics, the MPC strategy ensures optimal performance for the one-GT system across various driving scenarios, aligning with both desired trajectories and ensuring refined control actions.

The primary goal of this objective function is to minimize a weighted sum of performance metrics over the prediction horizon. The specific weighting factors reflect the trade-offs between tracking accuracy, control effort, and system stability. The objective function is given by:

$$\begin{aligned} J = \frac{1}{2} \sum_{i=0}^{\mathrm{N}-1} \Big(& w_{\mathrm{e_y}} e_{\mathrm{y},i+1}^2 + w_{\mathrm{e}_\psi} e_{\psi,i+1}^2 + w_{\mathrm{e}_\varphi} e_{\varphi,i+1}^2 \\ & + w_{\delta_\mathrm{r}} (\delta_i - \delta_{\mathrm{i,r}})^2 + w_{\bar{\delta}} (\delta_i - \bar{\delta}_i)^2 + w_{\Delta\delta} (\delta_i - \delta_{\mathrm{i}-1})^2 \Big) . \end{aligned} \tag{5.20}$$

Here, N represents the prediction horizon, and the weighting factors $w_{(\cdot)}$ denote the weights for the respective states.

To find an optimal control sequence $\boldsymbol{u}^* = [\mathbf{u}_0, \dots, \mathbf{u}_{\mathrm{N}-1}]^{\mathrm{T}}$, the objective function is minimized. The quadratic optimization problem for the MPC

is given as:

$$\begin{aligned}\boldsymbol{u}^* = \underset{\mathbf{u},\mathbf{e}_{\text{slack}}}{\arg\min}\, \frac{1}{2} \sum_{i=0}^{\mathrm{N}-1} \Bigg(& (\mathbf{x}_{i+1} - \mathbf{r}_{i+1})^{\mathrm{T}} \mathrm{Q}_{i+1} (\mathbf{x}_{i+1} - \mathbf{r}_{i+1}) \\ & + (\mathbf{u}_i - \mathbf{u}_{\mathrm{r},i})^{\mathrm{T}} \mathrm{R}_i (\mathbf{u}_i - \mathbf{u}_{\mathrm{r},i}) \\ & + (\mathbf{u}_i - \bar{\mathbf{u}}_i)^{\mathrm{T}} \mathrm{P}_i (\mathbf{u}_i - \bar{\mathbf{u}}_i) \\ & + (\mathbf{u}_{i-1} - \mathbf{u}_i)^{\mathrm{T}} \mathrm{S}_i (\mathbf{u}_{i-1} - \mathbf{u}_i) \Bigg) + \mathbf{e}_{\text{slack}}^2 \end{aligned} \tag{5.21a}$$

$$\text{s.t.}_{\text{slack},i} \quad \mathbf{x}_{i+1} = \mathrm{A}_i \mathbf{x}_{\mathrm{i}} + \mathrm{B}_i \mathbf{u}_i + \mathrm{I}\mathbf{z}_i, \tag{5.21b}$$

$$\mathbf{e}_{\min} - \mathrm{I}\mathbf{e}_{\text{slack},i} < \mathbf{x}_i - \mathbf{r}_i < \mathbf{e}_{\max} + \mathrm{I}\mathbf{e}_{\text{slack},i}, \tag{5.21c}$$

$$\theta_{\min} < \mathrm{C}\mathbf{x}_i < \theta_{\max}, \tag{5.21d}$$

$$\mathbf{u}_{\min} < \mathbf{u}_{i-1} < \mathbf{u}_{\max}, \tag{5.21e}$$

$$\Delta\mathbf{u}_{\min} < \Delta\mathbf{u}_i < \Delta\mathbf{u}_{\max}. \tag{5.21f}$$

Here, the weighting matrices for states, input, input deviation from nominal and input smoothness are Q_{i+1}, R_i, S_i, and P_i, respectively. The optimization problem is solved using the NGMPC framework.

5.2.2. Virtual Model Predictive Control (VMPC)

During reverse maneuvers, trailers exhibit inherent nonlinear dynamics accompanied by marked instability. This nonlinear nature and instability pose substantial challenges when applying model predictive control and optimization techniques. Direct linearization is often either ineffective or outright unfeasible for such unstable systems. Moreover, deriving an optimal control formulation based on these unstable dynamics usually leads to non-convergent or non-robust solutions. This section introduces a stable virtual model to address instability, providing a foundation for more efficient formulation of optimal control strategies.

In the standard dynamics of a one-GT system, the primary actuator is the steering wheel, which allows the driver to maneuver the system. Thus, the steering wheel angle is typically the primary input in the mathematical model. However, in the virtual model of the one-GT system, the actuator directly sets a desired hitch angle ϑ, with an internal control system adjusting the steering wheel accordingly.

The state vector for this virtual system is formalized as:

$$\mathbf{x}_k = [x_k, y_k, \psi_k, \varphi_k, \delta_{k-1}]^{\mathrm{T}}. \tag{5.22}$$

The state steering angle $\delta_{\mathrm{k-1}}$ is an algebraic state, while the rest of the system comprises differential states.

The virtual system's input vector, centered around the desired hitch angle, is described as:

$$\mathbf{u}_k = [\vartheta_k]. \tag{5.23}$$

Lyapunov's control law, as explored in section 4.1, is integrated to stabilize the desired hitch angle. Upon the integration of Lyapunov's control law for hitch angle stabilization, the relationship for δ is expressed as:

$$\delta = \arctan\left(\frac{\mathrm{l_v}\sin\theta - K(\theta-\vartheta)\mathrm{l_t}\mathrm{l_v}\cos\theta}{\mathrm{l_t} + \mathrm{l_h}\cos\theta + K(\theta-\vartheta)\mathrm{l_t}\mathrm{l_h}\sin\theta}\right).$$

The virtual model guarantees stable control for forward and reverse one-GT maneuvers. The virtual model system dynamics, using the discrete

spatial vehicle dynamic equation in eq. (5.18), are derived as follows:

$$x_{k+1} = x_k + \mathrm{ds}\cos\psi_k, \tag{5.24}$$

$$y_{k+1} = y_k + \mathrm{ds}\sin\psi_k, \tag{5.25}$$

$$\psi_{k+1} = \psi_k + \frac{\mathrm{ds}(\mathrm{l_v}\sin\theta_k - \mathrm{l_h}\cos\theta_k\tan\delta_k)}{\mathrm{l_t}(\mathrm{l_v}\cos\theta_k + \mathrm{l_h}\sin\theta_k\tan\delta_k)}, \tag{5.26}$$

$$\varphi_{k+1} = \varphi_k + \frac{\mathrm{ds}\tan\delta_k}{\mathrm{l_v}\cos\theta_k + \mathrm{l_h}\sin\theta_k\tan\delta_k}, \tag{5.27}$$

$$\delta_k = \arctan\left(\frac{\mathrm{l_v}\sin\theta_k - K(\theta_k - \vartheta_k)\mathrm{l_t}\mathrm{l_v}\cos\theta_k}{\mathrm{l_t} + \mathrm{l_h}\cos\theta_k + K(\theta_k - \vartheta_k)\mathrm{l_t}\mathrm{l_h}\sin\theta_k}\right). \tag{5.28}$$

Expressing the virtual system as a generalized non linear function:

$$\mathbf{x}_{k+1} = f_v(\mathbf{x}_k, \vartheta_k), \tag{5.29}$$

where $\mathbf{u}_k = [\vartheta_k]$, and ϑ_k is the desired hitch angle, which is the input of the virtual system.

Based on the integrated system dynamics and control strategy, the virtual system's dynamics formulation is provided by eq. (5.29). Similar to section 5.2.1, the equivalent prediction model $f_{\mathrm{v,eq}}(\mathbf{x}_i, \mathbf{u}_i)$ is derived using multi-step Euler integration of the system defined in eq. (5.29). The prediction model dynamics is given as:

$$\mathbf{x_{i+1}} = f_{\mathrm{v,eq}}(\mathbf{x}_i, \mathbf{u}_i), \tag{5.30}$$

where i is the prediction step.

Nominal Trajectory Generation

It is only necessary to compute ϑ, the desired hitch angle for the virtual model. The nominal control algorithm described earlier will be simplified, as no cascade control is required. The

details of this simplified algorithm are outlined in Algorithm 2.

Algorithm 2: Nominal Trajectory Generation Algorithm for VMPC

Parameters:

- N: Prediction horizon
- $\mathrm{l_t}$: Trailer length
- $\mathrm{N_r}$: Reference trajectory length

Input:

- $\mathbf{x}_0$: Initial system state
- $\{\mathbf{r}_k\}_{k=1}^{\mathrm{N_r}}$: Reference state at each time step

Output:

- $\{\bar{\mathbf{x}}_i\}_{i=0}^{N}$: Nominal state trajectory
- $\{\vartheta_i\}_{i=0}^{N-1}$: Desired hitch angle trajectory

Initialization:

$\bar{\mathbf{x}}_0 \leftarrow \mathbf{x}_0$

$\theta_{\text{current}} \leftarrow \theta_0$

Begin Algorithm:

for $i = 0$ *to* $N-1$ **do**

$\kappa_{\text{desired}} \leftarrow \text{PurePursuitControl}(\{\mathbf{r}_k\}_{k=1}^{\mathrm{N_r}}, \bar{\mathbf{x}}_i)$

$\vartheta_i \leftarrow \arctan(\kappa_{\text{desired}} \mathrm{l_t})$

$\bar{\mathbf{x}}_{i+1} \leftarrow f_{\text{v,eq}}(\bar{\mathbf{x}}_i, \vartheta_i)$

$\mathbf{r}_{i+1} \leftarrow \text{MatchReferenceState}(\{\mathbf{r}_k\}_{k=1}^{\mathrm{N_r}}, \bar{\mathbf{x}}_{i+1})$

Objective Function and Optimization Problem for VMPC

Distinct from the metrics in section 5.2, the VMPC places greater emphasis on maintaining a consistent hitch angle rather than the steering angle. The relevant hitch angle metrics for enhancing control precision include:

1 **Desired Hitch Angle Deviation from Reference ($\vartheta - \vartheta_{\mathbf{r}}$)**: Measures the difference between the actual and reference hitch angles.

2 **Deviation from Nominal Hitch Angle** ($\vartheta - \bar{\vartheta}$): Measure the difference from the nominal hitch angle.

3 **Hitch Angle Rate Deviation** ($\vartheta_i - \vartheta_{\mathrm{i}-1}$): Prioritizes angle consistency for a smoother control transition.

4 **Steering Angle Deviation from Nominal** ($\delta - \delta_{\mathrm{r}}$): Ensures proximity to the reference steering angle computed by the motion planner.

While the majority of the constraints remain consistent with MPC for the one-GT system as discussed in section 5.2, the distinct difference in the VMPC context is the introduction of the desired hitch angle (ϑ) constraints. This novel inclusion emphasizes managing the desired hitch angle in the VMPC model, reflecting its importance in optimizing system performance and safety.

Desired Hitch Angle Constraints:

$$\begin{aligned} \vartheta &\in [\vartheta_{\min}, \vartheta_{\max}] \\ \Delta\vartheta &\in [\Delta\vartheta_{\min}, \Delta\vartheta_{\max}] \end{aligned}$$

Stabilizing the hitch angle during reverse driving is essential for a truck-semitrailer system to prevent jackknifing. By focusing on hitch angle metrics aligned with the stable dynamics of the virtual model, precision is enhanced, constraints are managed effectively, and operational hazards are reduced.

The primary goal of this objective function is to minimize a weighted sum of performance metrics over the prediction horizon. The specific weighting factors reflect the trade-offs between tracking accuracy, control

effort, and system stability. The objective function is given by:

$$J = \frac{1}{2}\sum_{i=0}^{\mathrm{N}-1}\Big(w_{\mathrm{e_y}} e_{\mathrm{y},i+1}^2 + w_{\mathrm{e}_\psi} e_{\psi,i+1}^2 + w_{\mathrm{e}_\varphi} e_{\varphi,i+1}^2 + w_{\delta_\mathrm{r}}(\delta_{i+1} - \delta_{\mathrm{i+1,r}})^2 + w_{\vartheta_\mathrm{r}}(\vartheta_i - \vartheta_{\mathrm{r},i})^2 + w_{\bar{\vartheta}}(\vartheta_i - \bar{\vartheta}_i)^2 + w_{\Delta\vartheta}(\vartheta_i - \vartheta_{\mathrm{i-1}})^2 \Big). \tag{5.31}$$

Here, N represents the prediction horizon, and the weighting factors $w_{(\mathrm{keyword})}$ denote the weights for the respective states.

To find an optimal control sequence $\boldsymbol{u}^* = [\mathbf{u}_0, \dots, \mathbf{u}_{\mathrm{N}-1}]^\mathrm{T}$, the objective function is minimized. The quadratic optimization problem for the VMPC is given as:

$$\begin{aligned}
\boldsymbol{u}^* = \underset{\mathbf{u},\mathbf{e}_\mathrm{slack}}{\arg\min}\ & \frac{1}{2}\sum_{i=0}^{\mathrm{N}-1}\Big((\mathbf{x}_{i+1} - \mathbf{r}_{i+1})^\mathrm{T} \mathrm{Q}_{i+1} (\mathbf{x}_{i+1} - \mathbf{r}_{i+1}) \\
& + (\mathbf{u}_i - \mathbf{u}_{\mathrm{r},i})^\mathrm{T} \mathrm{R}_i (\mathbf{u}_i - \mathbf{u}_{\mathrm{r},i}) \\
& + (\mathbf{u}_i - \bar{\mathbf{u}}_i)^\mathrm{T} \mathrm{P}_i (\mathbf{u}_i - \bar{\mathbf{u}}_i) \\
& + (\mathbf{u}_{i-1} - \mathbf{u}_i)^\mathrm{T} \mathrm{S}_i (\mathbf{u}_{i-1} - \mathbf{u}_i) \Big) + \mathbf{e}_\mathrm{slack}^2
\end{aligned} \tag{5.32a}$$

$$\text{s.t.}_{\forall i \in [1,\mathrm{N}]} \quad \mathbf{x}_{i+1} = \mathrm{A}_i \mathbf{x}_\mathrm{i} + \mathrm{B}_i \mathbf{u}_i + \mathrm{I}\mathbf{z}_i, \tag{5.32b}$$

$$\mathbf{e}_\mathrm{min} - \mathrm{I}\mathbf{e}_{\mathrm{slack},i} < \mathbf{x}_i - \mathbf{r}_i < \mathbf{e}_\mathrm{max} + \mathrm{I}\mathbf{e}_{\mathrm{slack},i}, \tag{5.32c}$$

$$\vartheta_\mathrm{min} < \mathrm{C}_\theta \mathbf{x}_i < \vartheta_\mathrm{max}, \tag{5.32d}$$

$$\delta_\mathrm{min} < \mathrm{C}_\delta \mathbf{x}_i < \delta_\mathrm{max}, \tag{5.32e}$$

$$\mathbf{u}_\mathrm{min} < \mathbf{u}_{i-1} < \mathbf{u}_\mathrm{max}, \tag{5.32f}$$

$$\Delta\mathbf{u}_\mathrm{min} < \Delta\mathbf{u}_i < \Delta u_\mathrm{max}, \tag{5.32g}$$

where $\mathrm{C}_\theta = [0, 0, -1, 1, 0]$, $\mathrm{C}_\delta = [0, 0, 0, 0, 1]$. The parameters $\mathbf{e}_\mathrm{min}$ and $\mathbf{e}_\mathrm{max}$ represent the permissible state deviations from the reference trajectory.

5.2.3. Flatness-Based Model Predictive Control (FMPC)

In the context of one-GT systems, utilizing a flat model approach can significantly enhance the efficiency and accuracy of the control strategy. The concept of partial differential flatness transforms system dynamics into a system that does not exhibit numerical instability.

This subsection introduces the Flatness-Based Model Predictive Control (FMPC) for the one-GT system. Leveraging the partially flat model derived in previous sections, FMPC aims to optimize the trajectory of the truck-trailer system while respecting the system's dynamic constraints and ensuring collision-free trajectories.

The flat model system dynamics are defined by eq. (3.15) in section 2.1.2. Here, the model will be discretized to facilitate the application of FMPC. The subsequent states at index k can be represented as:

$$x_{k+1} = x_k + \mathrm{ds}\cos\psi_k, \tag{5.33}$$

$$y_{k+1} = y_k + \mathrm{ds}\sin\psi_k, \tag{5.34}$$

$$\psi_{k+1} = \psi_k + \frac{\mathrm{ds}(\mathrm{l_v}\sin(\varphi_k - \psi_k) - \mathrm{l_h}\cos(\varphi_k - \psi_k)\tan\delta_k)}{\mathrm{l_t}(\mathrm{l_v}\cos(\varphi_k - \psi_k) + \mathrm{l_h}\sin(\varphi_k - \psi_k)\tan\delta_k)}, \tag{5.35}$$

$$\varphi_{k+1} = \varphi_k + \frac{\mathrm{ds}\tan\delta_k}{\mathrm{l_v}\cos(\varphi_k - \psi_k) + \mathrm{l_h}\sin(\varphi_k - \psi_k)\tan\delta_k}, \tag{5.36}$$

$$\delta_k = \arctan\left(\frac{\mathrm{l_v}\sin\theta_k + \mathrm{l_v}\mathrm{l_t}\gamma\cos\theta_k}{\mathrm{l_t} + \mathrm{sign}(\bar{v})\mathrm{l_h}\cos\theta_k - \mathrm{l_t}\mathrm{l_h}\gamma\sin\theta_k}\right). \tag{5.37}$$

With the state vector expressed as:

$$\mathbf{x}_k = \left[x_k, y_k, \psi_k, \varphi_k, \delta_{k-1}\right]^{\mathrm{T}},$$

and the control vector depicted by:

$$\mathbf{u}_k = \left[\gamma_k\right],$$

where γ is the rate of change of the hitch angle with respect to the traveled distance.

Similar to section 5.2.1, the equivalent prediction model $f_{\text{f,eq}}(\mathbf{x}_i, \mathbf{u}_i)$ is derived using multi-step Euler integration of the system defined in eq. (5.33). The prediction model is given by the following equation:

$$\mathbf{x_{i+1}} = f_{\text{v,eq}}(\mathbf{x}_i, \mathbf{u}_i), \tag{5.38}$$

where i is the prediction step.

Nominal Trajectory Generation for FMPC

The algorithm generates the nominal trajectory by iteratively computing γ and updating the system states accordingly. The details of this algorithm are outlined in Algorithm 3.

Objective Function and Optimization Problem for FMPC

The transition to using the rate of change of hitch angle, symbolized by γ, as the system input for the FMPC prediction model is rooted in several critical considerations. Primarily, this shift ensures a continuous hitch angle, which subsequently guarantees a smoother steering wheel angle. Such smoothness is paramount for both vehicle safety and efficient operation.

Algorithm 3: Nominal Trajectory Generation Algorithm for FMPC

Parameters:

- N: Prediction horizon
- $\mathrm{l_t}$: Trailer length
- $\mathrm{N_r}$: Reference trajectory length

Input:

- $\mathbf{x}_0$: Initial system state
- $\{\mathbf{r}_k\}_{k=1}^{\mathrm{N_r}}$: Reference state at each time step

Output:

- $\{\bar{\mathbf{x}}_i\}_{i=0}^{N}$: Nominal state trajectory
- $\{\gamma_i\}_{i=0}^{N-1}$: Desired hitch angle trajectory

Initialization:

$\bar{\mathbf{x}}_0 \leftarrow \mathbf{x}_0$

$\theta_{\text{current}} \leftarrow \theta_0$

Begin Algorithm:

for $i = 0$ *to* $N-1$ **do**

$\kappa_{\text{desired}} \leftarrow \text{PurePursuitControl}(\{\mathbf{r}_k\}_{k=1}^{\mathrm{N_r}}, \bar{\mathbf{x}}_i)$

$\gamma_i \leftarrow -\mathrm{K}(\arctan(\kappa_{\text{desired}}\mathrm{l_t}) - \theta)$

$\bar{\mathbf{x}}_{i+1} \leftarrow f_{\text{f,eq}}(\bar{\mathbf{x}}_i, \gamma_i)$

$\mathbf{r}_{i+1} \leftarrow \text{MatchReferenceState}(\{\mathbf{r}_k\}_{k=1}^{\mathrm{N_r}}, \bar{\mathbf{x}}_{i+1})$

Key metrics for the FMPC are:

1 **Rate of Change of Hitch Angle Deviation from Reference** $(\gamma - \gamma_{\mathbf{r}})$: Measures the variance between the actual and the reference rate of change of hitch angles.

2 **Deviation from Nominal Rate of Change of Hitch Angle** $(\gamma - \bar{\gamma})$: Represents the optimal rate as proposed by the virtual model.

3 **Rate of Change of Hitch Angle Consistency** $(\gamma_i - \gamma_{\mathrm{i-1}})$: Emphasizes the significance of maintaining a consistent rate of change of hitch angle, facilitating seamless control transitions.

4 **Steering Angle Deviation from Nominal ($\delta - \delta_{\mathrm{r}}$)**: Asserts the importance of staying close to the reference steering angle, as elaborated in preceding sections.

By focusing on the rate change of the hitch angle, the system ensures continuity in both the hitch and steering angles. This smoothness in control actions extends the vehicle's lifespan by reducing wear and tear and elevates the safety and predictability of the one-GT system.

The objective for FMPC is expressed below:

$$\begin{aligned} J = \frac{1}{2}\sum_{i=0}^{\mathrm{N}-1}\Big(& w_{\mathrm{e_y}} e_{\mathrm{y},i+1}^2 + w_{\mathrm{e}_\psi} e_{\psi,i+1}^2 + w_{\mathrm{e}_\varphi} e_{\varphi,i+1}^2 \\ & + w_{\delta_{\mathrm{r}}}(\delta_{i+1} - \delta_{\mathrm{i+1,r}})^2 + w_{\gamma_{\mathrm{r}}}(\gamma_i - \gamma_{\mathrm{r},i})^2 \\ & + w_{\bar{\gamma}}(\gamma_i - \bar{\gamma}_i)^2 + w_{\Delta\gamma}(\gamma_i - \gamma_{\mathrm{i-1}})^2 \Big), \end{aligned} \tag{5.39}$$

where N is the prediction horizon, and the weighting factors $w_{()}$ denote the weights for the respective states.

To obtain the optimal control sequence $\boldsymbol{u}^*$, the FMPC minimizes an objective function using a slack variable vector $\mathbf{e}_{\mathrm{slack}} \in \mathbb{R}^5$. The ensuing quadratic optimization problem is:

$$\boldsymbol{u}^* = \underset{\mathbf{u},\mathbf{e}_{\mathrm{slack}}}{\arg\min}\, J(\mathbf{u}) + \mathbf{e}_{\mathrm{slack}}^2 \tag{5.40a}$$

$$\text{s.t.}_{\forall i\in[1,\mathrm{N}]} \quad \mathbf{x}_{i+1} = \mathrm{A}_i\mathbf{x}_{\mathrm{i}} + \mathrm{B}_i\mathbf{u}_i + \mathrm{I}\mathbf{z}_i, \tag{5.40b}$$

$$\mathbf{e}_{\min} - \mathrm{I}\mathbf{e}_{\mathrm{slack},i} < \mathbf{x}_i - \mathbf{r}_i < \mathbf{e}_{\max} + \mathrm{I}\mathbf{e}_{\mathrm{slack},i}, \tag{5.40c}$$

$$\gamma_{\min} < \mathcal{C}_\theta \mathbf{x}_i < \gamma_{\max}, \tag{5.40d}$$

$$\delta_{\min} < \mathcal{C}_\delta \mathbf{x}_i < \delta_{\max}, \tag{5.40e}$$

$$\mathbf{u}_{\min} < \mathbf{u}_{i-1} < \mathbf{u}_{\max}, \tag{5.40f}$$

$$\Delta\mathbf{u}_{\min} < \Delta\mathbf{u}_i < \Delta\mathbf{u}_{\max}. \tag{5.40g}$$

The parameters $\mathbf{e}_{\min}$ and $\mathbf{e}_{\max}$ outline allowable state deviations from the reference. The matrices C_{θ}, and $\mathcal{C}_{\delta}$, and their specific forms, can be cross-referenced with the VMPC section in section 5.2.2 for detailed understanding. The optimization problem is solved using the NGMPC framework.

5.3. System Architecture

This section describes the system architecture for experimentally validating the developed control algorithms. The system has advanced sensing and control hardware to ensure precise and reliable operation during testing and validation.

The truck and trailer have dual-antenna Differential Global Navigation Satellite System (DGNSS) sensors, a hitch angle sensor, and a rapid prototype controller with interfaces to control the actuators. The DGNSS sensors provide critical information about the current state, including the location of the truck and trailer, as well as the hitch angle. The mounting positions of these sensors are depicted in fig. 5.2.

The first DGNSS sensor provides the position of the truck in UTM coordinates along with the heading, while the second DGNSS sensor provides the position of the trailer's middle axle in UTM coordinates. Precise estimation of the hitch angle is crucial, especially during reversing maneuvers. Therefore, a hitch angle sensor measures the hitch angle directly. A PT-1 filter computes the sensor bias during straight-line driving to enhance the accuracy of the hitch measurement within $\pm 1^{\circ}$. This setup is depicted in fig. 1.1, showcasing the MAN truck semi-trailer prototype utilized for automated driving experiments.

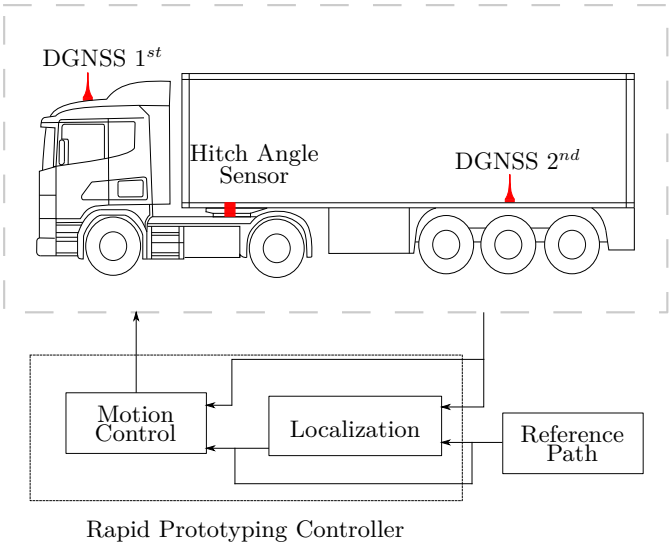


Figure 5.2.: Reduced system architecture for motion control testing and validation

This experimental prototype setup is critical for testing and validating the developed control algorithms. The motion control algorithms are tested and validated with a reference track logged using the DGNSS sensor to eliminate reliance on additional autonomous driving modules, such as motion planning and localization. As claimed by the manufacturer, the positional accuracy of the DGNSS sensors is 2 cm. This high level of accuracy allows for precise localization of the truck semi-trailer on the reference path without needing a separate localization module.

The control system operates on a rapid prototype controller, which includes all necessary software and hardware interfaces for validation. It ensures that the control algorithms can be tested in a realistic environment, allowing for comprehensive evaluation and refinement. The overall system architecture, illustrated in fig. 5.2, provides a robust platform for validating the efficacy of the motion control algorithms developed in this research.

5.4. Experimental Results and Validation

This section presents a detailed analysis of different control strategies for maneuvering a trailer along a predefined DGNSS path. The focus is on comparing the performance of the developed control strategies: Cascade Control consisting of a High-Level Pure Pursuit Control and a Low-Level Lyapunov Control (PPC-LC), Model Predictive Control (MPC), Virtual Model Predictive Control (VMPC), and Flatness-Based Model Predictive Control (FMPC), under various initial conditions. The experiments presented in this section are carried out with Haas during his master thesis [Haa21].

Figure 5.3 illustrates the reference maneuver used in this analysis, composed of a circular path followed by a sharp turn. This track has been specifically chosen to challenge the control strategies in handling both steady-state and dynamic conditions, making it an ideal test case for evaluating the performance of the developed approaches.

Figure 5.3.: Reference Maneuver, composed of a circle and a sharp turn.

Figure 5.4.: Negative initial deviation with PPC-LC. The trailer starts in the second lane from the left while the reference is in the third lane in a circular path. In the last frame, the vehicle jackknifes, indicating the need for driver intervention.

Figure 5.4 demonstrates a scenario with a negative initial deviation using PPC-LC. The trailer begins in the second lane from the left while the reference path is in the third lane. As the sequence progresses, the PPC-LC struggles to maintain the trailer's alignment with the reference path, resulting in a jackknife situation in the final frame, indicating the need for driver intervention. In contrast, fig. 5.5 demonstrates a similar negative initial deviation but with FMPC. Here, the trailer successfully converges to the third lane without requiring driver intervention, showcasing the robustness of FMPC in handling such deviations.

Figure 5.6 presents a scenario where the trailer starts with a positive initial deviation using PPC-LC. The trailer begins in the fourth lane while the reference is in the third lane. The images reveal that the trailer fails to converge to the desired path, highlighting the limitations of PPC-LC in correcting such deviations.

Figure 5.5.: Negative initial deviation with FMPC. The trailer starts in the second lane from the left while the reference is in the third lane in a circular path. The trailer successfully converges to the third lane without driver intervention.

Figure 5.6.: Positive initial deviation with PPC-LC. The trailer starts in the fourth lane while the reference is in the third lane. The trailer fails to converge to the reference lane, indicating that the PPC-LC could not bring the system back to the desired trajectory.

On the other hand, fig. 5.7 demonstrates the performance of FMPC under similar conditions. The trailer successfully converges to the third lane, demonstrating the superior convergence capabilities of FMPC in this context.

Figure 5.7.: Positive initial deviation with FMPC. The trailer starts in the fourth lane while the reference is in the third lane. The trailer successfully converges to the third lane, showing the effectiveness of the FMPC in correcting the trajectory.

Figure 5.8 illustrates a scenario with FMPC where the trailer starts with a positive hitch angle deviation. The trailer begins in the third lane with a positive misalignment of the hitch angle at the start position. Despite this, the system successfully converges to the desired trajectory, indicating the robustness of FMPC in handling initial misalignment. Similarly, fig. 5.9 presents a case with FMPC where the trailer starts with a negative hitch angle deviation. The trailer begins in the third lane with a negative misalignment of the hitch. The system successfully converges to the desired trajectory. This scenario is common, especially when significant differences in hitch angle occur at points requiring a switch from forward to backward motion.

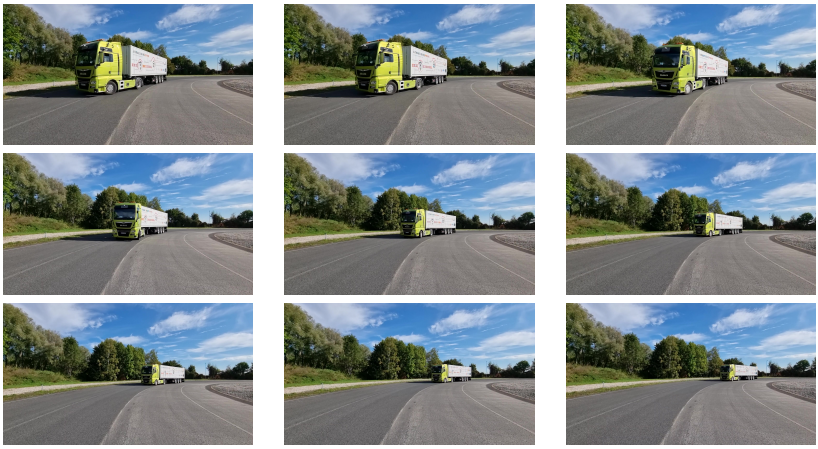


Figure 5.8.: FMPC with positive hitch angle deviation. The trailer starts in the third lane with a slight misalignment at the start position. The system successfully converges to the desired trajectory, indicating the robustness of the FMPC in handling initial misalignment.

Figure 5.9.: FMPC with negative hitch angle deviation. The trailer starts in the third lane with a slight misalignment at the start position. The system successfully converges to the desired trajectory.

To further illustrate the trailer's behavior during sharp turns, figs. 5.10 to 5.12 present sequences from external and internal perspectives.

Figure 5.10 focuses on the external view from a camera positioned outside the curve, showing how the truck and trailer align as they navigate the turn. Over a relatively short distance, the hitch angle changes significantly, as depicted in fig. 5.27, starting from a positive angle, transitioning to a negative angle, and finally returning to zero as the vehicle completes the turn. This sequence highlights the complexity of the trailer's motion and the dynamic adjustments required to maintain control.

Figure 5.10.: Truck and Trailer Maneuvering a Sharp Turn: Camera Positioned Outside of the Curve

Figure 5.11 shifts the perspective to an internal view from a camera positioned inside the curve, allowing for a closer examination of how the trailer follows the truck through the turn. This angle helps to visualize the spatial relationship between the truck and the trailer during the maneuver, emphasizing the importance of precise steering and speed control. The steering effort required to follow the curve is notably high, reflecting the demands of maintaining the desired trajectory through the turn.

Figure 5.11.: Truck and Trailer Maneuvering a Sharp Turn: Camera Positioned Inside of the Curve

Figure 5.12 provides a detailed look at the steering dynamics inside the vehicle. By focusing on the steering wheel angle, these images reveal the direct influence of the control system's input on the vehicle's trajectory through the turn. The steering angle remains consistent during steady-state driving before entering the curve, as seen in the initial frames and depicted in fig. 5.27. However, from the sixth to the tenth frame, significant changes in the steering angle are observed as the automated control system adjusts to navigate the turn. After completing the maneuver, the steering angle returns to zero, indicating the resumption of steady-state driving. This sequence captures the complexity of the steering control required by the autonomous system to maintain stability and direction through a sharp turn.

Figure 5.12.: Steering Angle and Vehicle Dynamics During a Sharp Turn

All three developed approaches (MPC, VMPC, and FMPC) are similar, with some differences in errors, control effort smoothness, and hitch angle dynamics. These differences are difficult to visualize in the video; hence, the visual results presented here focus on FMPC. The following section will present a comparative analysis of all the developed approaches using experimental results to provide a clearer understanding of their performance.

5.4.1. Experimental Results of PPC-LC

The PPC-LC strategy is tested under various conditions to evaluate its stability and tracking performance. Generally, PPC-LC struggles to converge from an initial lateral deviation, often becoming unstable and requiring driver intervention. This strategy serves as the nominal control law within the MPC framework, which includes MPC, VMPC, and FMPC approaches.

In the MPC setup, PPC-LC is used to compute the desired hitch angle, while the SLC method is applied to calculate the steering angle (δ). For VMPC, SLC is integrated directly within the system prediction model, enhancing the stability of the prediction model. Similarly, FMPC employs LC to compute the steering angle, i.e., partial flat model of the one-GT. This section demonstrates the nominal behavior of the system, as the trajectory computed using PPC-LC serves as the linearizing point for the developed NGMPC-based strategies.

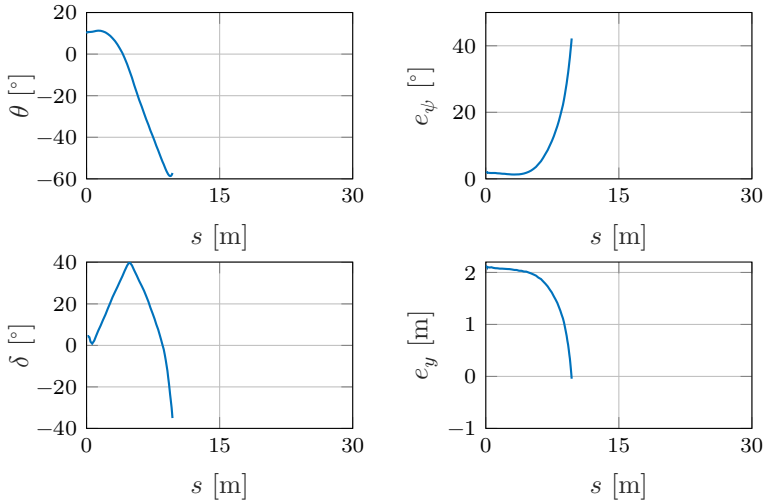


Figure 5.13.: Experimental results for PPC-LC. Maneuver: Reverse circular driving with positive initial lateral deviation.

Figure 5.13 demonstrates the performance of PPC-LC during a reverse circular driving maneuver with a positive initial lateral deviation. The trailer begins in the fourth lane while the reference is in the third lane. The figure illustrates that PPC-LC fails to bring the trailer back to the reference path, leading to instability.

Figure 5.14 presents the results when PPC-LC is tested with a negative initial lateral deviation. The trailer starts in the second lane while the reference is in the third lane. PPC-LC struggles to maintain alignment, resulting in jackknifing and requiring driver intervention.

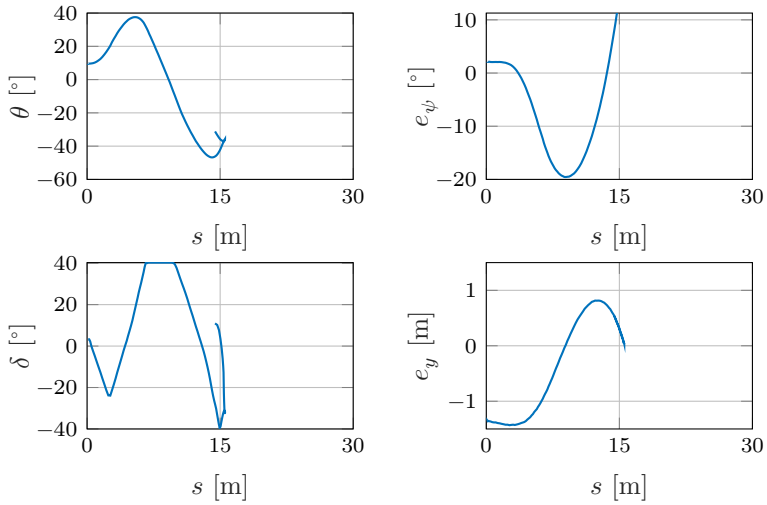


Figure 5.14.: Experimental results for PPC-LC. Maneuver: Reverse circular driving with negative initial lateral deviation.

In scenarios without significant initial deviations, Figure 5.15 demonstrates that PPC-LC can track the trajectory with a maximum error of around 0.6 m. However, the resulting hitch and steering angles exhibit noticeable oscillations, indicating the control strategy's limitations in maintaining smooth and stable control.

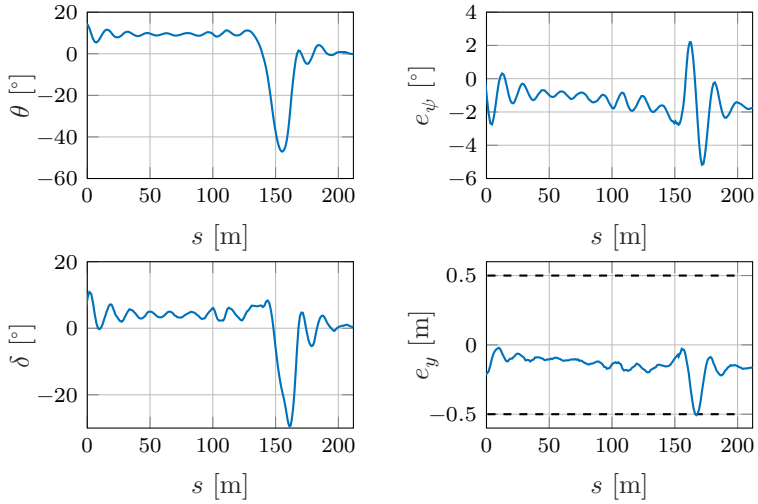


Figure 5.15.: Experimental results for PPC-LC. Maneuver: Reverse circular driving with a sharp turn near the end.

5.4.2. Experimental Results of MPC

The MPC strategy is capable of handling initial deviations from both positive and negative values, as demonstrated in figs. 5.16 and 5.17. However, MPC exhibits some oscillations during convergence, particularly with sudden changes in the hitch and steering angle rates.

Figure 5.16 demonstrates the performance of MPC during a reverse circular driving maneuver with a positive initial lateral deviation. The trailer starts with a deviation, but MPC can handle it, though some oscillations are observed during the convergence.

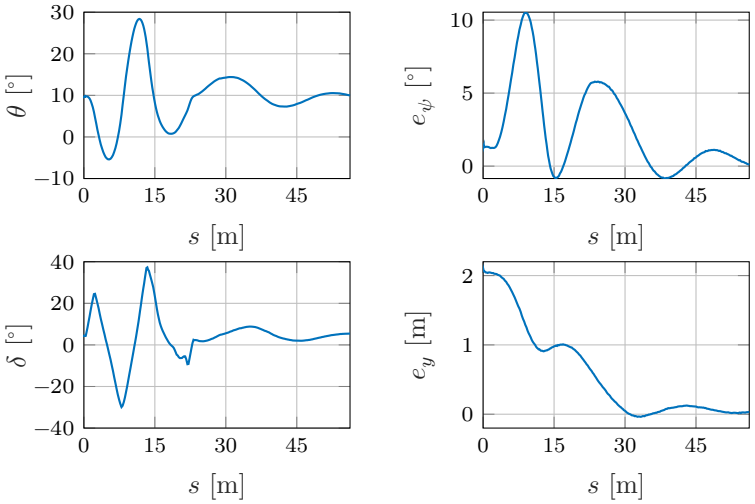


Figure 5.16.: Experimental results for MPC. Maneuver: Reverse circular driving with positive initial lateral deviation.

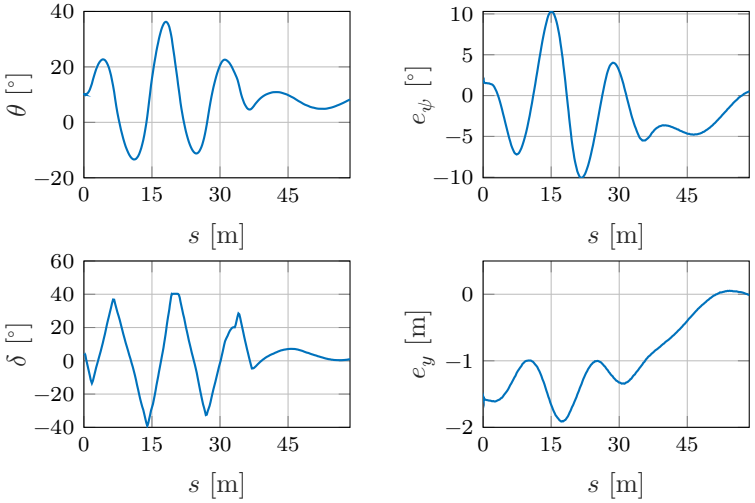


Figure 5.17.: Experimental results for MPC. Maneuver: Reverse circular driving with negative initial lateral deviation.

Figure 5.17 illustrates the scenario where MPC is tested with a negative initial lateral deviation. The system demonstrates similar behavior with oscillations during the convergence phase but maintains the trajectory. Figures 5.18 and 5.19 depict MPC's performance during more complex maneuvers involving sharp turns with 0.5 m and 0.3 m constraints. In these figures, MPC demonstrates the ability to trace the trajectory almost within constraints, though with a slight overshoot of the tolerance. The hitch and steering angles are smoother compared to PPC-LC, making MPC a more reliable option for scenarios requiring steady control.

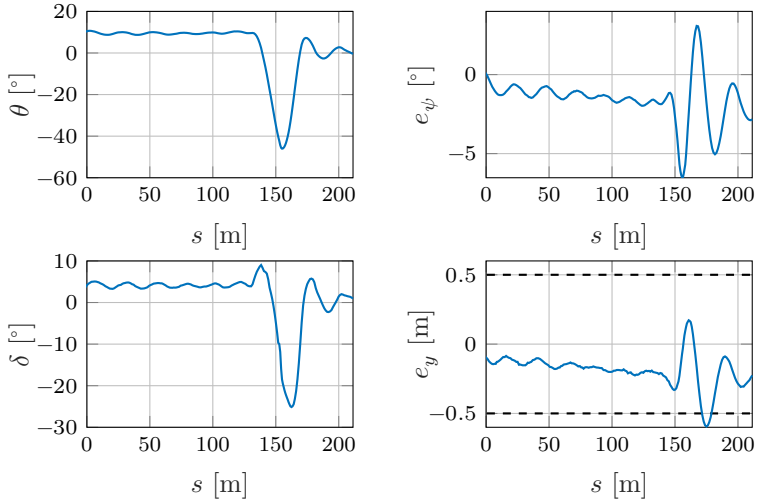


Figure 5.18.: Experimental results for MPC. Maneuver: Reverse circular driving with a sharp turn near the end with 0.5 m constraint.

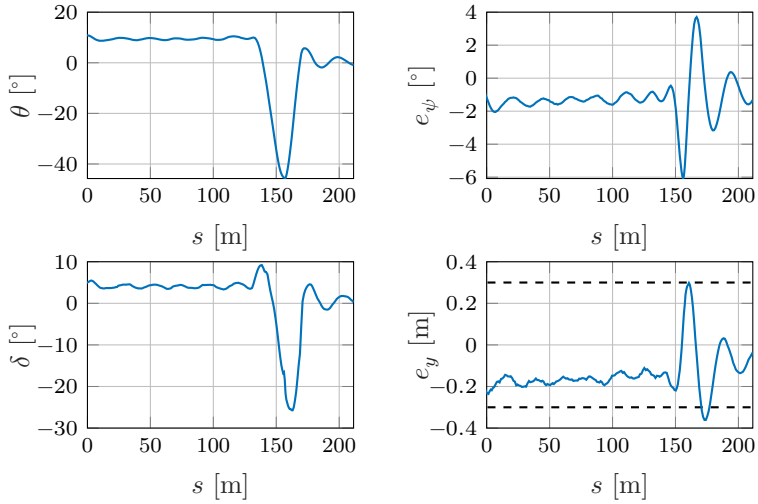


Figure 5.19.: Experimental results for MPC. Maneuver: Reverse circular driving with a sharp turn near the end with 0.3 m constraint.

5.4.3. Experimental Results of VMPC

The VMPC strategy improves upon MPC by ensuring continuous hitch and steering angle rates, leading to smoother transitions during maneuvers. As shown in figs. 5.20 and 5.21, VMPC effectively handles both positive and negative initial lateral deviations, maintaining the trajectory within the specified tolerance.

Figure 5.20 demonstrates the performance of VMPC during a reverse circular driving maneuver with a positive initial lateral deviation. VMPC successfully manages to bring the trailer back to the reference path, demonstrating a smooth and stable convergence.

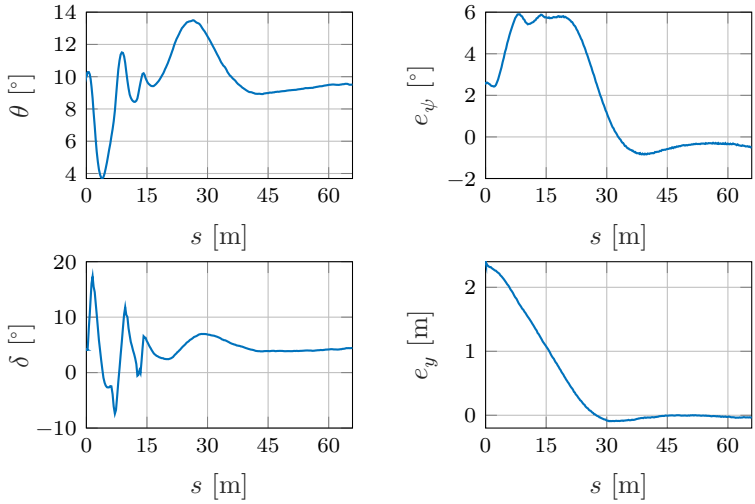


Figure 5.20.: Experimental results for VMPC. Maneuver: Reverse circular driving with positive initial lateral deviation.

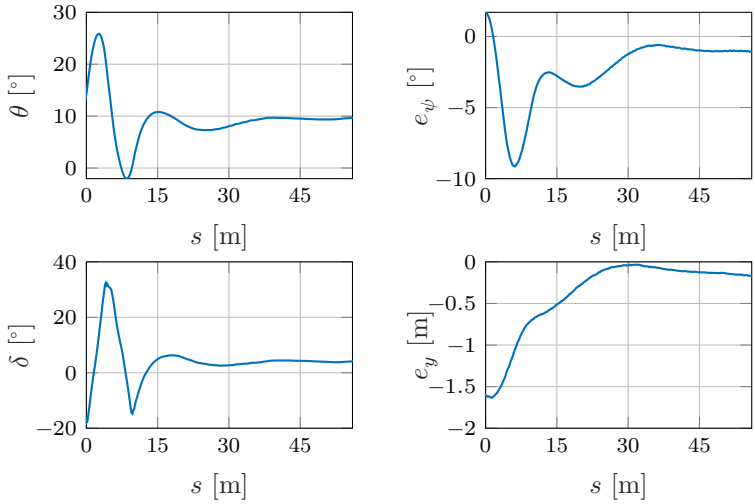


Figure 5.21.: Experimental results for VMPC. Maneuver: Reverse circular driving with negative initial lateral deviation.

Figure 5.21 presents the performance of VMPC with a negative initial lateral deviation. The trailer begins in the second lane while the reference is in the third lane. VMPC effectively handles the deviation and maintains a smooth hitch angle throughout the maneuver. Figures 5.22 and 5.23 further illustrate VMPC's performance during more complex maneuvers involving sharp turns with 0.5 m and 0.3 m constraints. The hitch angle remains relatively steady, and the steering angle is smoother compared to MPC, demonstrating the advantages of VMPC in achieving smoother control.

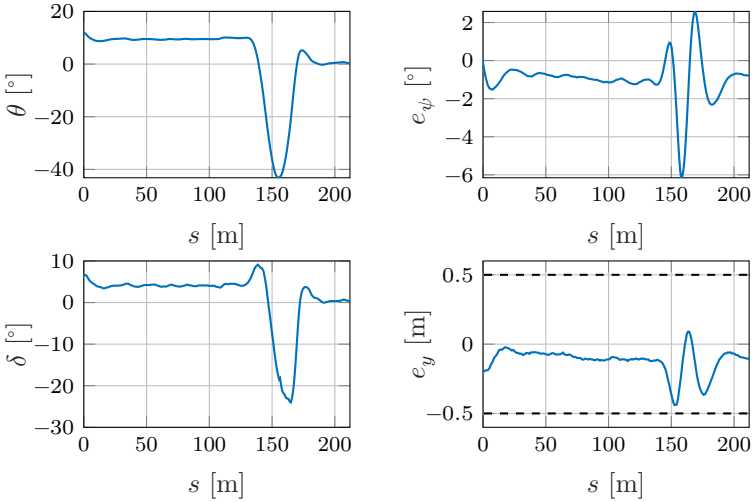


Figure 5.22.: Experimental results for VMPC. Maneuver: Reverse circular driving with a sharp turn near the end with 0.5 m constraint.

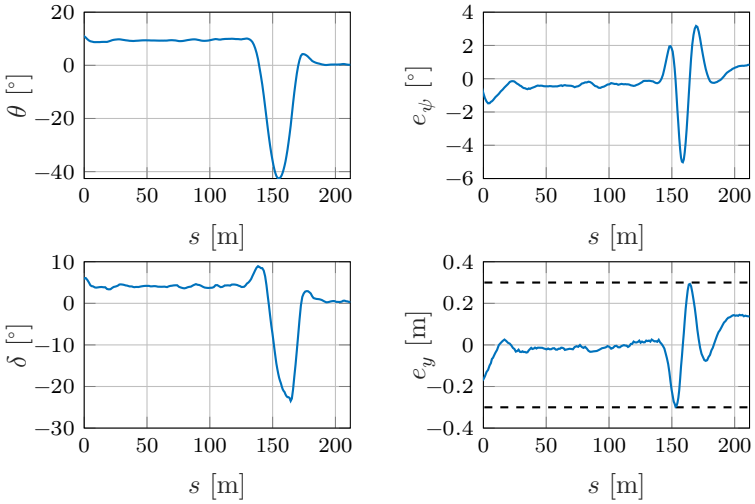


Figure 5.23.: Experimental results for VMPC. Maneuver: Reverse circular driving with a sharp turn near the end with 0.3 m constraint.

5.4.4. Experimental Results of FMPC

The FMPC strategy further enhances control smoothness by including the second derivative of the hitch angle in its cost function, in addition to the hitch angle and its rate. This modification results in even smoother steering and hitch angle dynamics, which are particularly noticeable during convergence.

Figure 5.24 demonstrates the performance of FMPC during a reverse circular driving maneuver with a positive initial lateral deviation. The results demonstrate good convergence and smooth control dynamics, indicating the effectiveness of FMPC in managing such scenarios.

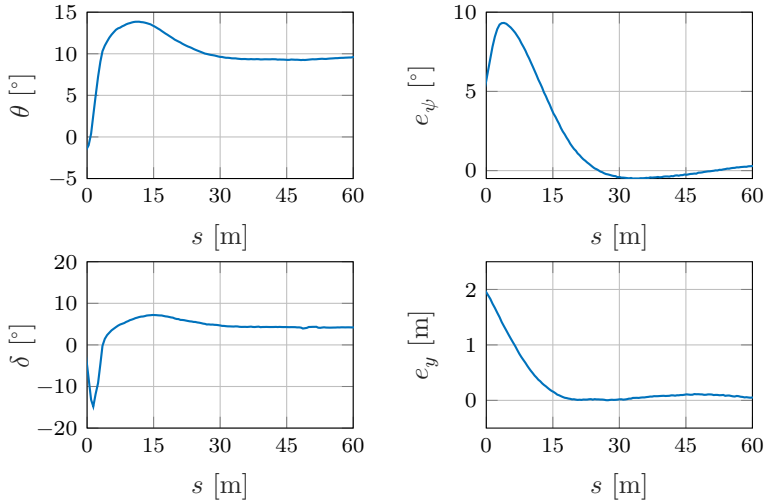


Figure 5.24.: Experimental results for FMPC. Maneuver: Reverse circular driving with positive initial lateral deviation.

Figure 5.25 demonstrates the performance of FMPC during a reverse circular driving maneuver with a negative initial lateral deviation. The trailer starts in the second lane while the reference is in the third lane. As shown, FMPC effectively handles the negative deviation, maintaining smooth steering and hitch angles throughout the maneuver.

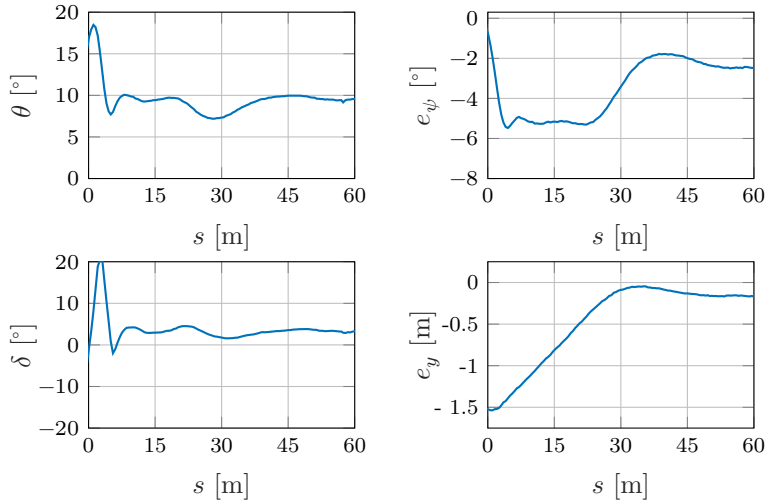


Figure 5.25.: Experimental results for FMPC. Maneuver: Reverse circular driving with negative initial lateral deviation.

Figure 5.26 highlights the performance of FMPC during a more complex maneuver: reverse circular driving with a sharp turn near the end, constrained to a 0.5 m tolerance. The figure demonstrates that FMPC manages this maneuver effectively, maintaining smooth control and adhering to the constraints.

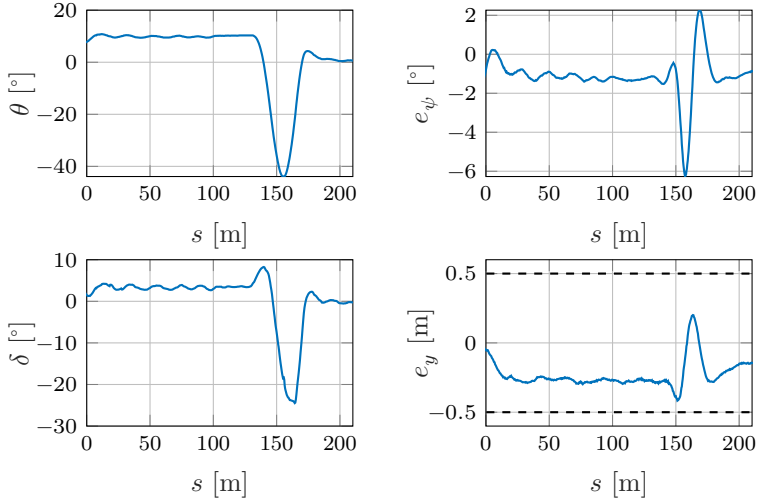


Figure 5.26.: Experimental results for FMPC. Maneuver: Reverse circular driving with a sharp turn near the end with 0.5 m constraint.

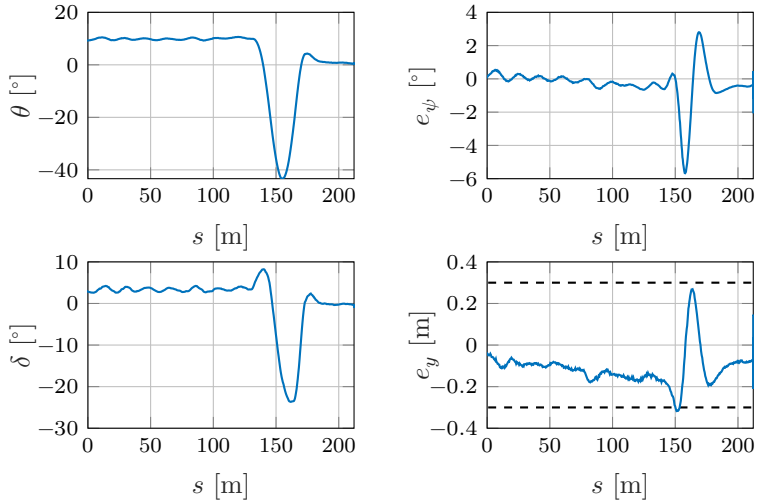


Figure 5.27.: Experimental results for FMPC. Maneuver: Reverse circular driving with a sharp turn near the end with 0.3 m constraint.

Finally, fig. 5.27 depicts the same maneuver as fig. 5.26, but with a tighter constraint of 0.3 m. Despite the increased difficulty, FMPC continues to perform well, showcasing its ability to handle strict constraints while maintaining smooth control dynamics.

5.5. Summary

This chapter introduced the Nominally Guided MPC (NGMPC) framework, a highly efficient tool for addressing the challenges inherent in nonlinear systems, with a primary focus on the one-GT. NGMPC framework is designed to formulate a quadratic programming problem for nonlinear systems efficiently, ensuring optimal trajectory stabilization control.

The chapter explored three developed approaches for trajectory stabilization in one-GT systems:

- Model Predictive Control (MPC): Utilizes the known dynamics of the one-GT system within the NGMPC framework.
- Virtual Model Predictive Control (VMPC): Employs a virtual model where the hitch angle is considered a virtual input, simplifying the control strategy while using the same NGMPC framework.
- Flatness-Based Model Predictive Control (FMPC): Leverages the concept of partial differential flatness, using a partial flat system as the prediction model within the NGMPC framework.

Experimental validation has demonstrated that the system's state variables, including positions and orientations, are maintained within a safe operational zone with a tolerance band around the reference trajectory. This approach accounts for minor deviations while ensuring the vehicle remains stable and secure. Among the strategies, although some oscillations were observed, MPC showed improved handling of initial deviations compared to PPC-LC. VMPC further refined control smoothness by ensuring continuous hitch and steering angle rates, leading to better

stability. FMPC offered the most robust performance, effectively managing complex maneuvers with tight constraints, delivering the smoothest and most stable control. Notably, the FMPC approach, which utilizes a flat model, is not limited to control tasks but also finds applications in trajectory planning and optimization. The next chapter will explore these broader applications of FMPC.

The developed methods have proven highly effective in managing the inherent instability of one-GT systems, particularly during reverse driving maneuvers. Additionally, the real-time capabilities of these methods ensure that the control algorithms can make swift adjustments to maintain the desired trajectory, providing robust performance even in dynamic and uncertain environments. The experimental results highlight the practical applicability and reliability of the NGMPC framework and the developed predictive models, namely virtual and flat system models, establishing their value for real-world implementations in automated driving systems.

6. Trajectory Planning and Optimization for one-GT

Trajectory planning for truck-trailer systems is particularly challenging due to the complex dynamics and non-holonomic constraints inherent to these systems [Ber19, Gua21, Lef05]. The difficulty increases in narrow environments, where optimal control approaches must be progressively constrained to navigate tight spaces [Li20]. Achieving optimal trajectory planning typically requires a good initial guess [Nat23, Zip15]. The initial trajectory guesses' quality significantly influences the optimization process's success and the final trajectory's quality.

Some researchers have utilized the model's symmetry to address numerical instability, which can make model-based optimization methods conservative. Nevertheless, utilizing symmetry in finding trajectories may be ineffective in tight spaces where continuous directional changes are needed to reach the goal. Another approach is to simplify the motion planning process by approximating the one-GT as an one-ST. However, as discussed in section 3.5, the validity of this approximation decreases with increasing hitch length and the trailer's curvature. Therefore, treating the one-GT as an one-ST may not always be feasible.

This chapter addresses trajectory planning and optimization of one-GT. Trajectory planning for different trailer configurations will be compared, and the impact of simplifying the one-GT as an one-ST will be discussed.

6.1. Trajectory Planning: one-GT

Trajectory planning is formulated as a nonlinear programming problem that leverages the concept of partial differential flatness for an one-GT. This approach considers partial flat model dynamics, ensuring a stable system model, particularly during reversing maneuvers. The objective is to generate optimal trajectories that the truck-trailer system can accurately follow. Trajectory planning aims to determine a feasible trajectory that meets the system's kinematic constraints, ensures smooth and efficient motion, and is collision-free.

6.1.1. Collision Constraints

Extensive research has been conducted on collision constraints in trajectory planning. Researchers have explored various geometrical objects, such as circles, polygons, rectangles, ellipsoids, and sigmoid functions, to formulate constraints for collision avoidance [Pau22, Ple17, Smi17, Ger11, Bri19, Gut17].

Several studies have focused on collision avoidance and optimal path planning for vehicles with multiple trailers [Moh17, Gue19]. Various approaches, including lattice-based frameworks and neural network algorithms, have been developed to generate kinematically feasible and collision-free paths for these vehicles [Lju19, Qur19].

[Lam03] presents a method for path optimization in highly constrained environments, specifically for complex kinematic systems like trucks with two trailers. Collision avoidance was managed by defining a potential that increases as the distance from an obstacle decreases. Moreover, [Bus95] introduces an obstacle avoidance algorithm for a car pulling trailers, focusing on designing a collision-free path for a mobile robot configured with trailers. By incorporating a trailer correction factor and upper bounds on the off-tracking of the trailers, the algorithm aims to ensure safe path planning by accounting for the unique dynamics of a vehicle pulling trailers.

Most works handle trajectory planning using sample-based planners, incorporating collision checks and stabilizing the system on the sample-based trajectory. In this work, collision constraints are considered within the nonlinear programming problem. The collision avoidance constraints consist of two parts: defining the geometry of the truck and trailer and defining the obstacles in the occupancy. This section begins by exploring how the obstacles are formulated.

As illustrated in fig. 6.1, the plot demonstrates a typical parking lot designed for truck-trailers. The figure showcases the layout and numbered parking locations, providing a visual representation of how such a parking lot is structured.

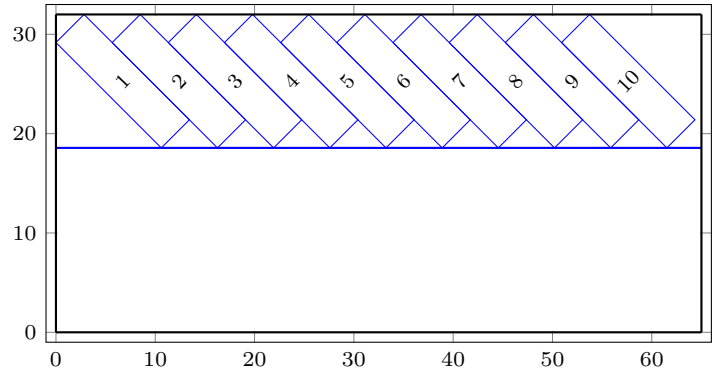


Figure 6.1.: The plot illustrates an example parking lot for truck-trailers.

Figure 6.2 converts the parking lot into constraints and obstacles for trajectory planning. This figure details the geometrical constraints and the obstacles within the parking area, which are essential for formulating collision avoidance strategies and ensuring safe navigation for truck trailers.

In the figure, all the parking spaces, whether occupied or free, are represented by overlapping circles. These circles simplify the space taken up by potential obstacles, making it easier to compute collision constraints. Additionally, the parking lot boundary is discretized and sampled with

circles of a smaller radius. This approach ensures that no part of the truck trailer collides with the boundary, as defining upper and lower bounds alone would not be sufficient for large systems. Any occupied grid cell or clusters of cells can be mapped as a cicles.

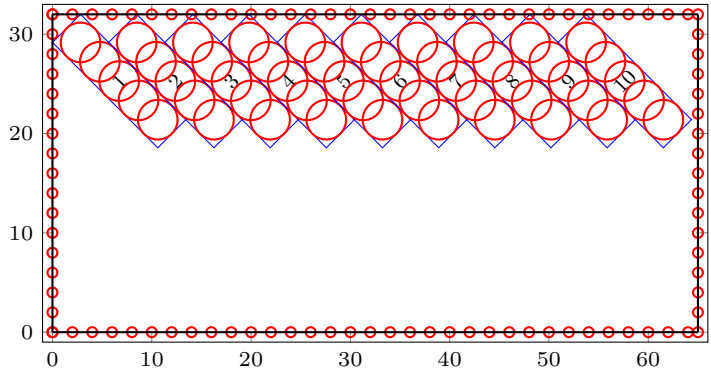


Figure 6.2.: The plot illustrates the conversion of a parking lot layout into circular obstacles for defining collision constraints for trajectory planning.

Figure 6.3 illustrates the selection of relevant circular obstacles within the parking lot for trajectory planning. Rather than converting the entire parking lot into circular obstacles, this figure highlights the specific circles crucial for the selected parking spot. This focused approach ensures that only the pertinent obstacles are considered, reducing the complexity within the trajectory planning problem and enhancing the efficiency of collision avoidance strategies.

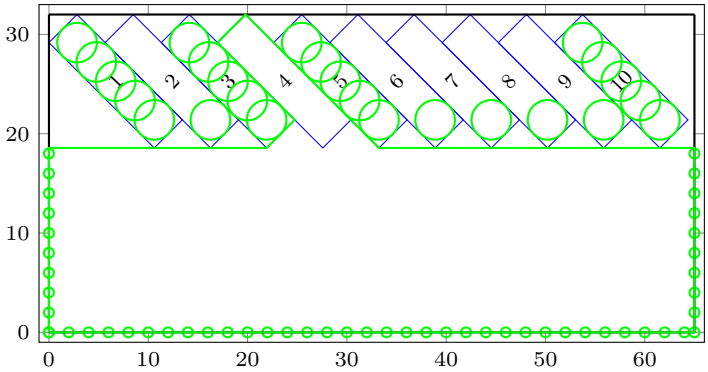


Figure 6.3.: The plot illustrates selection of relevant obstacles for a specific parking spot that must be considered to ensure collision-free navigation.

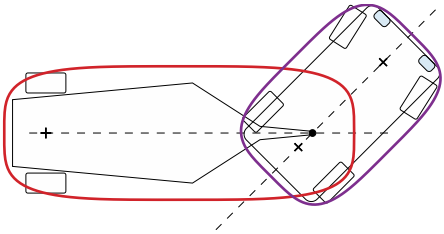


Figure 6.4.: The plot demonstrates the selected geometry for the truck and trailer as super ellipses.

With the geometry of the objects within the environment defined, the focus will now shift to how the system is geometrically defined.

Figure 6.4 demonstrates the selected geometry for the truck and trailer as super ellipses and defines the collision-free criteria between the super ellipse and a circle.

To ensure a collision-free trajectory, the following constraint must be satisfied for all circular obstacles:

$$\left(\frac{x_c}{a+r+\epsilon}\right)^n + \left(\frac{y_c}{b+r+\epsilon}\right)^n > 1, \tag{6.1}$$

where (x_c, y_c) are the coordinates of the circle center in the coordinate system of the ellipse, and r is the radius of the circle. For the trailer, $a = \frac{\mathrm{l}_{\mathrm{trailer}}}{2}$ and $b = \frac{\mathrm{w}_{\mathrm{trailer}}}{2}$, where $\mathrm{l}_{\mathrm{trailer}}$ is the length of the trailer and $\mathrm{w}_{\mathrm{trailer}}$ is the width of the trailer. Similarly, for the truck, $a = \frac{\mathrm{l}_{\mathrm{truck}}}{2}$ and $b = \frac{\mathrm{w}_{\mathrm{truck}}}{2}$, where $\mathrm{l}_{\mathrm{truck}}$ is the length of the truck and $\mathrm{w}_{\mathrm{truck}}$ is the width of the truck. The parameter ϵ is a small positive value for safety margin. This constraint ensures that the truck and trailer, modeled as super ellipses, avoid all circular obstacles in the environment, maintaining a safe and collision-free trajectory.

6.1.2. Problem Formulation

This section focuses on the trajectory planning for the one-GT system. The primary objective is to find an optimal path that navigates the trailer from a given start position to a specified goal pose while avoiding obstacles.

Trajectory planning considers circular obstacles defined in the map, ensuring the planned path remains collision-free while respecting the trailer's kinematic constraints. This trajectory planning framework aims to generate smooth, feasible paths that adhere to the physical constraints of the one-GT system and navigate through complex environments without collisions.

Consider, the state vector for the one-GT system as:

$$\mathbf{x}_i = [x_i, y_i, \psi_i, \theta_i, \delta_i] \in \mathbb{R}^5, \tag{6.2}$$

where x_i and y_i denote the position of the trailer at time step i, ψ_i is the heading angle, θ_i is the hitch angle, and δ_i is the steering angle.

The system input is:

$$\mathbf{u}_i = [\gamma_i, ds_i] \in \mathbb{R}^2, \tag{6.3}$$

with γ_i being the rate change of the hitch angle w.r.t. distance traveled by trailer and ds_i representing the distance traveled at time step i.

The dynamics of the one-GT system were derived using the continuous flat model defined by the eq. (3.15) in section 3.4 and discretized using the fourth-order Runge-Kutta (RK4) integration method, resulting in the following discrete-time equations:

$$\mathbf{x}_{i+1} = f(\mathbf{x}_i, \mathbf{u}_i), \tag{6.4}$$

where f encapsulates the kinematic relationships between the truck and the trailer.

The initial state $\mathbf{x}_0 \in \mathbb{R}^5$ is given as:

$$\mathbf{x}_0 = \mathbf{x}(0). \tag{6.5}$$

To guide the system to a desired final position and orientation, a goal state $\mathbf{x}_\mathrm{N} \in \mathbb{R}^5$ is defined. The terminal cost J_f for achieving this goal is:

$$J_\mathrm{f} = (\mathbf{x}_\mathrm{N} - \mathbf{x}_g)^\mathrm{T} \mathrm{Q}_\mathrm{f} (\mathbf{x}_\mathrm{N} - \mathbf{x}_g), \tag{6.6}$$

where $\mathbf{x}_\mathrm{N}$ represents the state at the final time step N, $\mathbf{x}_g \in \mathbb{R}^5$ is the goal state, and $\mathrm{Q}_\mathrm{f} \in \mathbb{R}^{5\times5}$ is a weight matrix that prioritizes accuracy in reaching the goal state. To ensure proximity to the goal, a terminal constraint is imposed:

$$\|\mathbf{x}_\mathrm{N} - \mathbf{x}_g\| \leq \epsilon. \tag{6.7}$$

The running cost, which penalizes deviations in the state and input vectors over time, is given by:

$$J_{\mathrm{r}} = \sum_{i=0}^{\mathrm{N}-1} \left(\mathrm{Q}_{\theta} \theta_i^2 + \mathrm{Q}_{\delta} \delta_i^2 \right), \tag{6.8}$$

where $\mathrm{Q}_{\theta} \in \mathbb{R}$ and $\mathrm{Q}_{\delta} \in \mathbb{R}$ are weight coefficients for θ_i and δ_i, respectively. Similarly, the running costs for the inputs J_{u} and the change in inputs $J_{\Delta \mathrm{u}}$ are:

$$J_{\mathrm{u}} = \sum_{i=0}^{\mathrm{N}-1} \mathbf{u}_i^{\mathrm{T}} \mathrm{R}_{\mathrm{u}} \mathbf{u}_i. \tag{6.9}$$

$$J_{\Delta \mathrm{u}} = \sum_{i=0}^{N-2} (\mathbf{u}_{i+1} - \mathbf{u}_i)^{\mathrm{T}} \mathrm{R}_{\Delta \mathrm{u}} (\mathbf{u}_{i+1} - \mathbf{u}_i), \tag{6.10}$$

where $\mathrm{R}_{\mathrm{u}} \in \mathbb{R}^{2 \times 2}$ is the weight matrix for the control inputs, and $\mathrm{R}_{\Delta \mathrm{u}} \in \mathbb{R}^{2 \times 2}$ is the weight matrix for the changes in control inputs.

Input constraints ensure control inputs stay within feasible ranges throughout the trajectory:

$$\mathbf{u}_{\min} \leq \mathbf{u}_i \leq \mathbf{u}_{\max}, \quad \mathbf{u}_{\min}, \mathbf{u}_{\max} \in \mathbb{R}^2. \tag{6.11}$$

Additionally, a collision constraint ensures the system maintains a safe distance from obstacles:

$$g(\mathbf{x}_i) > 1, \tag{6.12}$$

where $g(\mathbf{x}_i)$ represents the distance to obstacles at time step i.

The complete Nonlinear Programming (NLP) problem is formulated as follows:

$$
\begin{aligned}
\min_{\mathbf{x},\mathbf{u}} \quad & J = J_{\mathrm{f}} + J_{\mathrm{r}} + J_{\mathrm{u}} + J_{\Delta \mathrm{u}} \\
\text{s.t.}_{\forall i \in [0,\mathrm{N}-1]} \quad & \mathbf{x}_0 = \mathbf{x}(0), \\
& \|\mathbf{x}_{\mathrm{N}} - \mathbf{x}_g\| \leq \epsilon, \\
& \mathbf{u}_{\min} \leq \mathbf{u}_i \leq \mathbf{u}_{\max}, \\
& \theta_{\min} \leq \theta_i \leq \theta_{\max}, \\
& \delta_{\min} \leq \delta_i \leq \delta_{\max}, \\
& g(\mathbf{x}_i) > 1, \quad i = 0, \ldots, \mathrm{N}-1, \\
& \mathbf{x}_{i+1} = f(\mathbf{x}_i, \mathbf{u}_i).
\end{aligned}
\tag{6.13}
$$

Incorporating these elements ensures the one-GT system can effectively navigate complex environments with smooth, collision-free trajectories, adhering to all physical and operational constraints. The trajectory planning problem was formulated and solved using Casadi, a symbolic framework for numerical optimization [And19].

6.2. Results and Discussions

The following section presents the results of the planned trajectories for the one-GT under various parking maneuvers. This analysis demonstrates the versatility and effectiveness of the developed trajectory planning algorithm, showcasing its applicability across different configurations. The algorithm's ability to plan paths requiring continuous adjustment in the direction of travel to reach the goal for various parking maneuvers is discussed. The focus is on the area swept by the one-GT, illustrating that the planned trajectories are collision-free. Additionally, trajectories for the same maneuver across different configurations are compared, providing insights into the hitch and steering angles and their compliance with constraints. Each figure illustrates the area swept

by the one-GT for different configurations of trailer connections: in front of the rear axle ($l_h = -1\,m$), at the rear axle ($l_h = 0\,m$), and behind the rear axle ($l_h = 1\,m$). The results for parallel, angled, and perpendicular parking are presented.

Table 6.1.: Model parameters and dimensions for one-GT

Parameter	Description	Value (m)
l_v	Wheelbase of the Truck	3.6
l_t	Length of the Trailer	8
l_h	Hitch Length for Trailer	-1, 0, 1
$l_{trailer}$	Trailer's Geometric Length	12
l_{truck}	Truck's Geometric Length	5
$l_{rear,t}$	Trailer's Rear Overhang	3.1
$l_{rear,v}$	Truck's Rear Overhang	0.7
w	Width of Truck	2.5

6.2.1. Parallel Parking

In parallel parking maneuvers, the one-GT is required to park parallel to the curb, which involves a combination of forward and backward movements to align the vehicle within the parking space.

Trailer Connected in Front of the Rear Axle ($l_h = -1\,m$): The area swept by the one-GT during parallel parking with the trailer connected in front of the rear axle is illustrated in fig. 6.5. This configuration results in a specific trajectory pattern that must be followed to successfully park the vehicle.

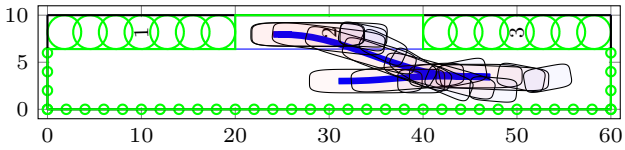


Figure 6.5.: The figure illustrates the area swept by the one-GT for parallel parking and the trailer connected in front of the rear axle ($l_h = -1\,\mathbf{m}$).

Trailer Connected at the Rear Axle ($l_h = 0\,m$): The area swept by the one-GT during parallel parking with the trailer connected at the rear axle is shown in fig. 6.6. This setup showcases a different trajectory due to the altered pivot point of the trailer connection.

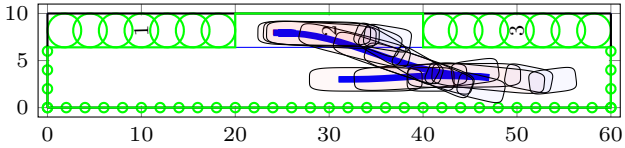


Figure 6.6.: The figure illustrates the area swept by the one-GT for parallel parking and the trailer connected at the rear axle ($l_h = 0\,m$).

Trailer Connected Behind the Rear Axle ($l_h = 1\,m$): Figure 6.7 demonstrates the area swept by the one-GT during parallel parking with the trailer connected behind the rear axle. This configuration results in yet another unique trajectory, reflecting the extended pivot point.

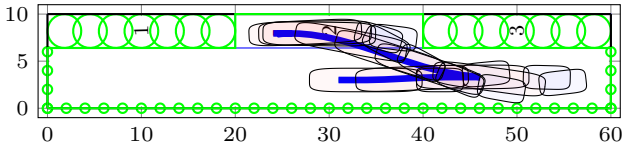


Figure 6.7.: The figure illustrates the area swept by the one-GT for parallel parking and the trailer connected behind the rear axle ($l_h = 1\,m$).

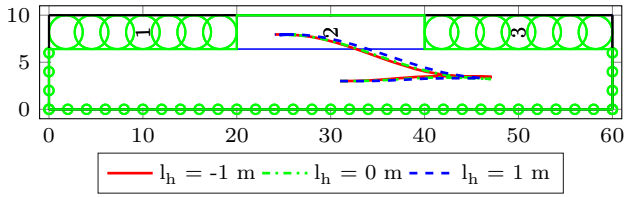


Figure 6.8.: The figure illustrates the trajectories planned for one-GT for parallel parking with different trailer connections: in front of the rear axle ($l_h = -1\,m$), at the rear axle ($l_h = 0\,m$), and behind the rear axle ($l_h = 1\,m$).

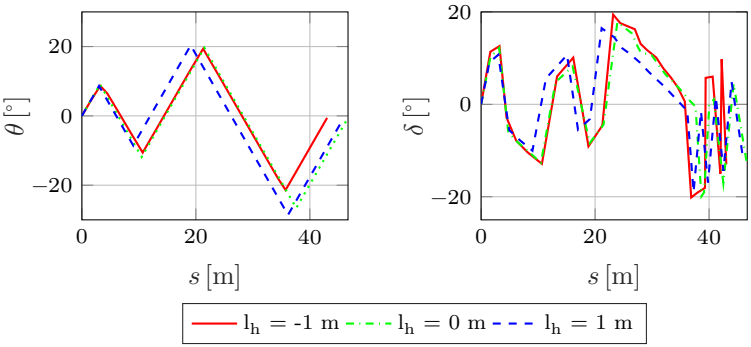


Figure 6.9.: The figures demonstrate the hitch angle θ and steering angle δ vs. distance s for parallel parking with different trailer connections: in front of the rear axle ($l_h = -1\,m$), at the rear axle ($l_h = 0\,m$), and behind the rear axle ($l_h = 1\,m$).

6.2.2. Angled Parking

Angled parking requires the one-GT to enter the parking spot at an angle, typically around 45 degrees. This maneuver can be more challenging due to the increased complexity of the trajectory.

Trailer Connected in Front of the Rear Axle ($l_h = -1\,m$): The area swept by the one-GT during angled parking with the trailer connected in front of the rear axle is depicted in fig. 6.10.

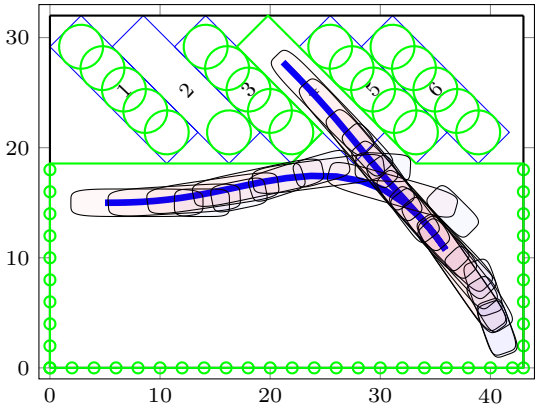


Figure 6.10.: The figure illustrates the area swept by the one-GT for angled parking and the trailer connected in front of the rear axle ($l_h = -1\,\mathbf{m}$).

Trailer Connected at the Rear Axle ($l_h = 0\,m$): fig. 6.11 demonstrates the area swept by the one-GT during angled parking with the trailer connected at the rear axle.

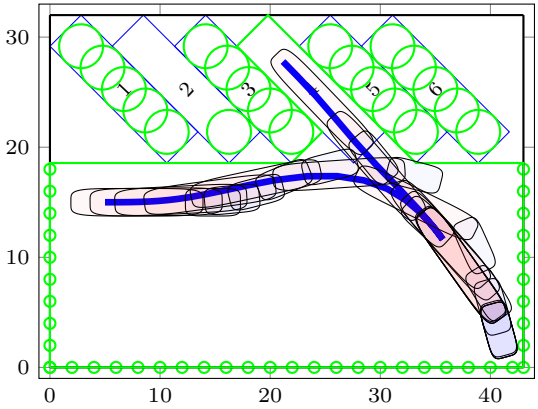


Figure 6.11.: The figure illustrates the area swept by the one-GT for angled parking and the trailer connected at the rear axle ($l_h = 0\,m$).

Trailer Connected Behind the Rear Axle ($l_h = 1\,m$): 6.12 demonstrates the area swept by the one-GT during angled parking with the trailer connected behind the rear axle.

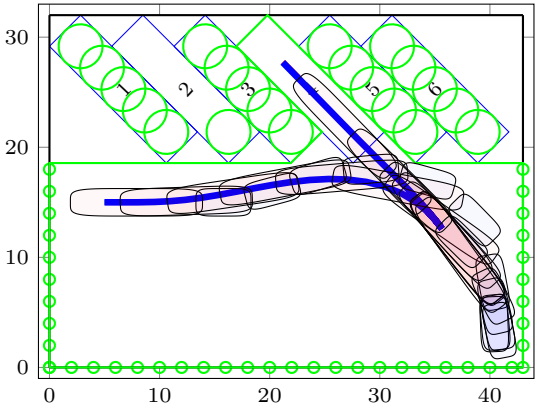


Figure 6.12.: The figure illustrates the area swept by the one-GT for angled parking and the trailer connected behind the rear axle ($l_h = 1\,\mathbf{m}$).

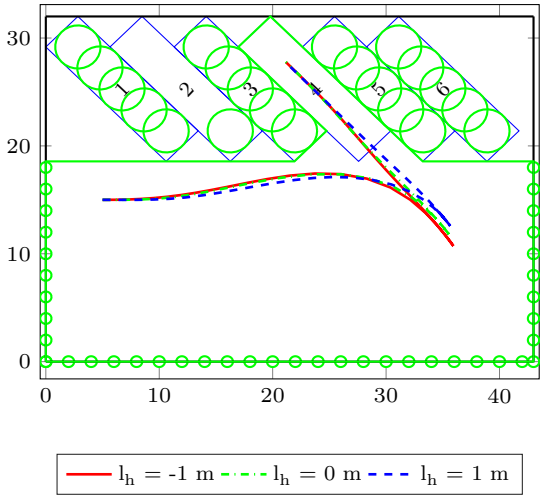


Figure 6.13.: The figure illustrates the area swept by the one-GT for angled parking with different trailer connections: in front of the rear axle ($l_h = -1\,m$), at the rear axle ($l_h = 0\,m$), and behind the rear axle ($l_h = 1\,m$).

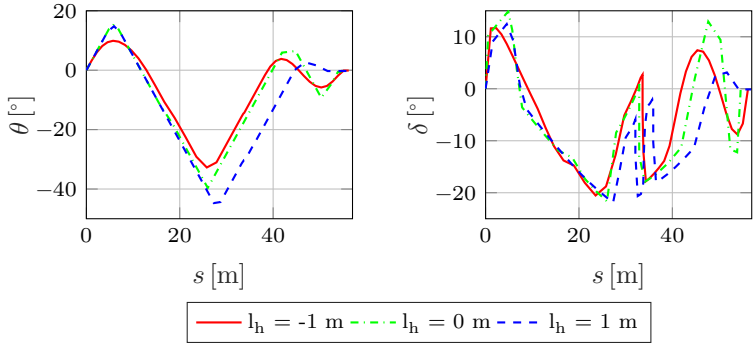


Figure 6.14.: The figures demonstrate the hitch angle θ and steering angle δ vs. distance s for angled parking with different trailer connections: in front of the rear axle ($l_h = -1\,m$), at the rear axle ($l_h = 0\,m$), and behind the rear axle ($l_h = 1\,m$).

6.2.3. Perpendicular Parking

Perpendicular parking involves positioning the vehicle at a right angle to the parking space, requiring precise maneuvering.

Trailer Connected in Front of the Rear Axle ($l_h = -1\,m$): The area swept by the one-GT during perpendicular parking with the trailer connected in front of the rear axle is illustrated in fig. 6.15.

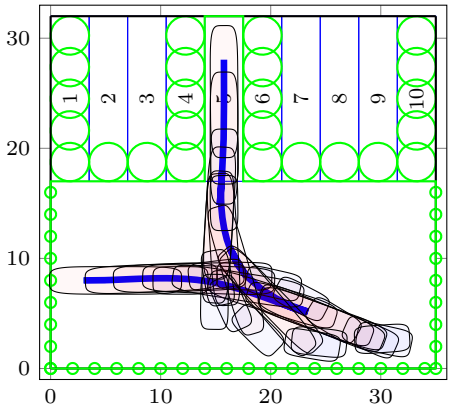


Figure 6.15.: The figure illustrates the area swept by the one-GT for perpendicular parking and the trailer connected in front of the rear axle ($l_h = -1m$).

Trailer Connected at the Rear Axle ($l_h = 0\,m$): fig. 6.16 demonstrates the area swept by the one-GT during perpendicular parking with the trailer connected at the rear axle.

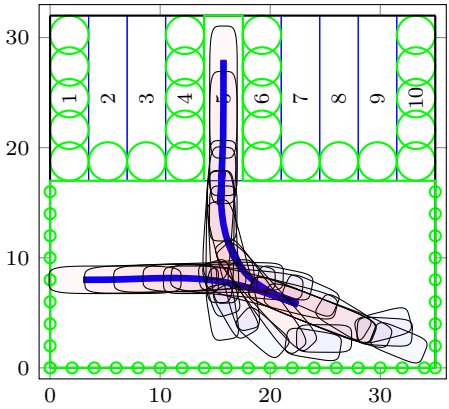


Figure 6.16.: The figure illustrates the area swept by the one-GT for perpendicular parking and the trailer connected at the rear axle ($l_h = 0$m).

Trailer Connected Behind the Rear Axle ($l_h = 1$ m): Figure 6.17 demonstrates the area swept by the one-GT during perpendicular parking with the trailer connected behind the rear axle.

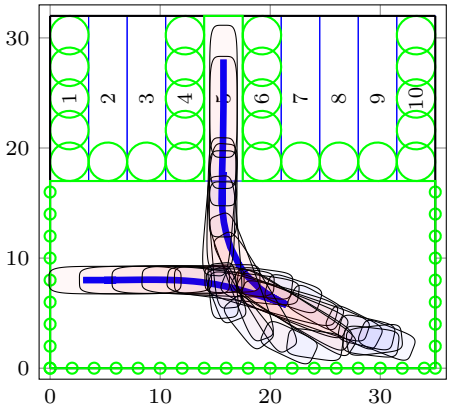


Figure 6.17.: The figure illustrates the area swept by the one-GT for perpendicular parking and the trailer connected behind the rear axle ($l_h = 1m$).

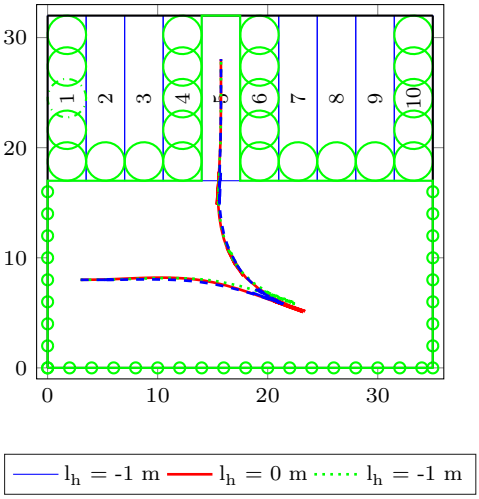


Figure 6.18.: The figure illustrates the area swept by the one-GT for perpendicular parking with different trailer connections: in front of the rear axle ($l_h = -1\,m$), at the rear axle ($l_h = 0\,m$), and behind the rear axle ($l_h = 1\,m$).

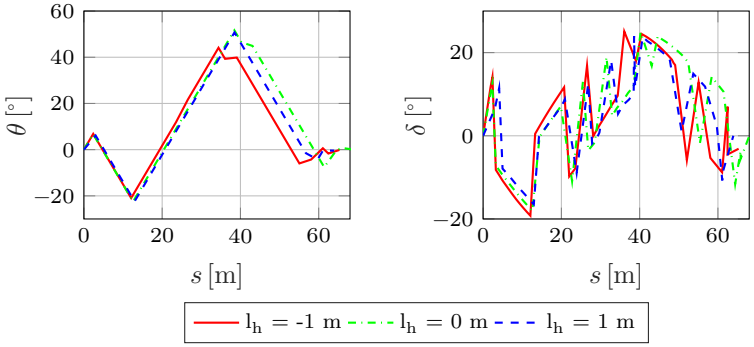


Figure 6.19.: The figures demonstrate the hitch angle θ and steering angle δ vs. distance s for perpendicular parking with different trailer connections: in front of the rear axle ($l_h = -1\,m$), at the rear axle ($l_h = 0\,m$), and behind the rear axle ($l_h = 1\,m$).

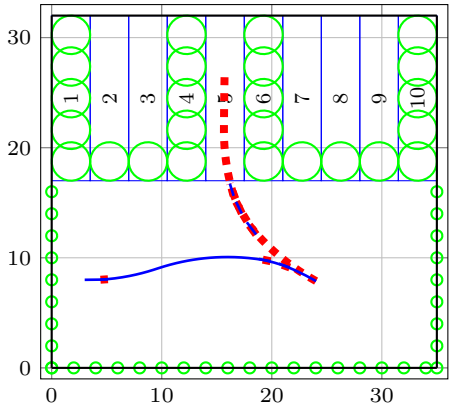


Figure 6.20.: The figure illustrates the planned trajectory for perpendicular parking for the one-GT system. The dashed trajectory denotes the reverse maneuver, while the solid trajectory indicates the forward maneuver.

Figure 6.20 illustrates the planned trajectory for perpendicular parking using the one-GT system. In the figure, the dashed trajectory denotes the reverse maneuver, while the solid trajectory indicates the forward maneuver.

This figure highlights the effectiveness of the trajectory planning algorithm in managing complex parking maneuvers that require continuous direction changes. By seamlessly transitioning between forward and reverse movements, the algorithm ensures that the truck and trailer navigate efficiently into the parking space without collisions. This capability is crucial for practical applications where space constraints necessitate precise and adaptive maneuvering. The ability to plan trajectories that incorporate continuous direction changes demonstrates the robustness and versatility of the planning algorithm in handling real-world parking scenarios.

Despite the optimality of planned trajectories, they can sometimes exhibit high jerks and may not be feasible due to significant discretization steps and model linearizations. To address these issues, the second part

of this chapter focuses on trajectory optimization. The goals of trajectory optimization are twofold:

1. Drivability: The trajectory optimization aims to achieve a high-order continuity for the trajectory. For a non-holonomic system such as a truck trailer, achieving minimum G^3 continuity is essential for effectively navigating the path even with an infinite steering rate.

2. Feasibility Check: The developed trajectory optimization algorithm can ensure feasibility for a given trajectory.

6.3. Trajectory Optimization

Dubins and Reeds-Shepp paths are known for creating minimal-length steering paths for nonholonomic vehicles by combining arc-shaped turns and straight line segments [Gim22]. These paths are designed to be continuously differentiable (G^1) and typically consist of no more than three segments between positions [Día19]. Dubins paths, for example, are made up of three trajectory segments, which can be either circular arcs or straight lines [Lug14]. While these paths offer the shortest travel distance, they lack curvature continuity, leading to abrupt changes in trajectory curvature at the transition points between arcs and straight segments [Nua].

Clothoid paths create curvature continuous paths but do not achieve G^3 continuity [Gim22]. As shown by Fraichard and Scheuer, curvature continuous paths approximate Reeds-Shepp paths, which are known to be the shortest for connecting points with heading continuity [Lam21]. Lima et al. developed an optimization algorithm to compute sparse, curvature continuous trajectories by maximizing the zeros in the second derivative of the curvature vector. Oliveira et al. introduced a method for maintaining sharpness continuity using cubic curvature paths, which respect steering limitations for articulated vehicles. Still, their approach is limited to connecting only two arbitrary positions.

In Section 3.5, the relationship between steering and hitch angle continuity and the flat point was examined, concluding that for a truck-trailer system to maintain continuity, the trajectory should be at least G^3 continuous. This section presents an optimization method designed to generate more feasible driving paths.

Inspired by the findings, this section presents an optimization method that generates paths that are not only feasible but also optimally sharp and continuous. Inspired by the work of Oliveira et al. [Oli18] and Lima et al. [Lim15, Lim16, Lim17], this method aims to ensure G^4 continuity, thereby enhancing the feasibility and smoothness of the truck-trailer system's motion.

The optimization presented here extends our previous work detailed in [Kum22b]. In that study, the focus was on vehicle dynamics without trailers, utilizing clothoids to define system dynamics and sharpness as the rate of change of curvature. Here, this approach is extended to optimize paths for a vehicle with a trailer.

Two models are utilized to optimize the trajectory: one-ST, defined by eq. (3.10) in section 3.3.1, and one-GT, defined in section 3.4. The continuous spatial models are discretized using the fourth-order Runge-Kutta (RK4) integration method.

The first model is an extension of the clothoid model from [Kum22b], which computes the steering and hitch angles for an one-ST using the system's differential flatness.

The second model is based on partial flatness, as described by eq. (3.15) in section 3.4. Here, sharpness is defined by the rate of change of the hitch angle (γ) instead of curvature. This model aims to find a trajectory where the hitch angle is continuous up to the second derivative. This approach ensures smoother transitions and enhances the overall stability of the truck-trailer system during maneuvers.

The state vector for both models is $\mathbf{x} = [x, y, \psi, \kappa, \theta, \delta]$ consisting of the trailer's position (x, y), heading (ψ), curvature (κ), hitch angle (θ), and steering angle (δ).

In a nutshell, objective aims to minimize the l_0 norm of the third derivative of the system input $\boldsymbol{u} = [\mathbf{u}_0^{\mathrm{T}}, \mathbf{u}_1^{\mathrm{T}}, \dots, \mathbf{u}_{\mathrm{N}-1}^{\mathrm{T}}]_{\mathrm{N}\times 1}^{\mathrm{T}}$ where N is the number of reference waypoints.

The optimization problem seeks to find an optimal SC trajectory near a known reference path, consisting of N waypoints defined by $[x_{\mathrm{ref}}, y_{\mathrm{ref}}, \psi_{\mathrm{ref}}, \kappa_{\mathrm{ref}}, s_i] \forall i \in [0, \mathrm{N}]$. The optimization is subject to constraints ensuring the system states (x, y, ψ) stay within a tolerance band around the reference trajectory and the hitch and steering angles remain within valid regions.

$$
\begin{aligned}
&\min_{\mathbf{u}} \|\mathcal{D}_3 \mathbf{u}\|_0 \\
&\text{s.t.}_{\forall i \in [1,\mathrm{N}]} \quad \mathbf{x}_{i+1} = f(\mathbf{x}_i, \mathbf{u}_i), \\
&|x_i - x_{r,i}| \leq e_x, \\
&|y_i - y_{r,i}| \leq e_y, \\
&|\psi_i - \psi_{r,i}| \leq e_\psi, \\
&|\theta_i| \leq \theta_{\max}, \\
&|\delta_i| \leq \delta_{\max}, \\
&|\delta_i'| \leq \delta'_{\max}, \\
&|\mathbf{u}_i| \leq \max.
\end{aligned}
\tag{6.14}
$$

The constraints on the steering angle δ and its rate of change δ' collectively handle the actuation constraints and vehicle dynamics, ensuring that both the steering angle and the steering angle rate remain within their respective limits. Additionally, the constraint on the hitch angle θ ensures that it stays within its maximum allowable range.

The distance between consecutive waypoints is defined by the vector

$$
\mathbf{l} = [l_0, l_1, \dots, l_{\mathrm{N}-1}]^{\mathrm{T}}. \tag{6.15}
$$

Suppose $\mathbf{F}(\mathbf{l})$ is a matrix-valued function that maps $\mathbf{l} \in \mathbb{R}^{\mathrm{N}}$ to $\mathbb{R}^{(\mathrm{N}-1)\times \mathrm{N}}$ and is defined as:

$$\mathbf{F}(\mathbf{l}) = \begin{bmatrix} \frac{1}{l_0} & -\frac{1}{l_0} & & \\ & \ddots & \ddots & \\ & & \frac{1}{l_{\mathrm{N}-1}} & -\frac{1}{l_{\mathrm{N}-1}} \end{bmatrix}. \tag{6.16}$$

The matrix operators for the first, second, and third derivatives are then constructed using $\mathbf{F}(\mathbf{l})$:

$$\begin{aligned} \mathcal{D}_1 &= \mathbf{F}(\mathbf{l}), \\ \mathcal{D}_2 &= \mathbf{F}(\mathbf{l}_1)\mathbf{F}(\mathbf{l}), \\ \mathcal{D}_3 &= \mathbf{F}(\mathbf{l}_2)\mathbf{F}(\mathbf{l}_1)\mathbf{F}(\mathbf{l}), \end{aligned}$$

where

$$\begin{aligned} \mathbf{l}_1 &= \left[\frac{l_0 + l_1}{2}, \frac{l_1 + l_2}{2}, \dots, \frac{l_{\mathrm{N}-2} + l_{\mathrm{N}-1}}{2}\right]^{\mathrm{T}}, \\ \mathbf{l}_2 &= \left[\frac{l_0 + 2l_1 + l_2}{4}, \frac{l_1 + 2l_2 + l_3}{4}, \dots, \frac{l_{\mathrm{N}-3} + 2l_{\mathrm{N}-2} + l_{\mathrm{N}-1}}{4}\right]^{\mathrm{T}}. \end{aligned}$$

The objective is now transformed to $||\boldsymbol{w}_3 \odot \mathcal{D}_3 \boldsymbol{u}||_1$. The constraints need to be linearized to solve the l_1 norm optimization. Following the approach defined in section 5.1.1, the model is linearized around the nominal state and input paths $[\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}]$, serving as the initial guess.

When using a sample-based planner, the curvature or hitch angle of the reference trajectory might not be accurate. In some cases, this information may not be available. Trying to fit piece-wise curvature to the trajectory can also lead to inaccuracies. In such cases, the nominal state and control input can be computed as defined in section 5.1. It aids in improving convergence by ensuring that the optimization starts in the vicinity of the reference, thereby providing a more reliable initial guess and enhancing the overall stability of the optimization process.

The linearized equation for the state vector $\boldsymbol{x} = [\mathbf{x}_1^{\mathrm{T}}, \dots, \mathbf{x}_{\mathrm{N}}^{\mathrm{T}}]$ as a function of the input vector $\boldsymbol{u}$ is given by eq. (5.4) and is written as follows:

$$\boldsymbol{x} = \mathcal{A}\mathbf{x}_0 + \mathcal{B}\boldsymbol{u} + \mathcal{E}\boldsymbol{z}, \tag{6.17}$$

where $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{u}}$ are the nominal state and input vectors.

The linearized optimization problem is given as follows:

$$\begin{aligned}
\min_{\mathbf{u}} \quad & ||\boldsymbol{w}_3 \odot \mathcal{D}_3\boldsymbol{u}||_1 + ||\varphi||_2 && \text{(6.18a)}\\
\text{s.t.} \quad & \boldsymbol{x} = \mathcal{A}\mathbf{x}_0 + \mathcal{B}\boldsymbol{u} + \mathcal{E}\boldsymbol{z}, && \text{(6.18b)}\\
& |\mathcal{C}\boldsymbol{x} - \boldsymbol{r}| \leq \boldsymbol{\varepsilon} + \boldsymbol{\varphi}, && \text{(6.18c)}\\
& |\mathcal{C}_\delta\boldsymbol{x}| \leq \delta_{\max}, && \text{(6.18d)}\\
& |\mathcal{D}_1\boldsymbol{\delta}| \leq \boldsymbol{\delta}'_{\max}, && \text{(6.18e)}\\
& |\mathcal{C}_\theta\boldsymbol{x}| \leq \theta_{\max}, && \text{(6.18f)}\\
& |\boldsymbol{u}| \leq \boldsymbol{u}_{\max}, && \text{(6.18g)}
\end{aligned}$$

where $\boldsymbol{\varepsilon} = [\varepsilon_1^{\mathrm{T}}, \dots, \varepsilon_{\mathrm{N}}^{\mathrm{T}}]^{\mathrm{T}}$ and $\varepsilon_i = [\varepsilon_{\mathrm{x}}, \varepsilon_{\mathrm{y}}, \varepsilon_\psi]^{\mathrm{T}}$, $\boldsymbol{r} = [\mathbf{r}_1^{\mathrm{T}}, \dots, \mathbf{r}_{\mathrm{N}}^{\mathrm{T}}]^{\mathrm{T}}$ and $\mathbf{r}_i = [x_{\mathrm{ref,i}}, y_{\mathrm{ref,i}}, \psi_{\mathrm{ref,i}}]^{\mathrm{T}}$.

$\mathcal{C}_\delta$ and $\mathcal{C}_\theta$ map the steering and hitch angles from the state vector, respectively, while $\mathcal{C}$ maps the state to the desired output $[x, y, \psi]$.

6.3.1. Sparsification of the Optimized Path

Assume $\boldsymbol{u}^*$ is the optimal input vector. Consider a vector $\boldsymbol{\mu}$ representing the weighted third-order derivative of $\boldsymbol{u}$, defined as follows:

$$\begin{aligned}
&\boldsymbol{\mu} = [\mu_0, \dots, \mu_{\mathrm{N}-3}]^{\mathrm{T}},\\
&\boldsymbol{\mu} = |\boldsymbol{w}_3\mathcal{D}_3\boldsymbol{u}^*|.
\end{aligned}$$

Kink points are identified as the points where the elements of $\boldsymbol{\mu}$ exceed a threshold ϵ. Once the $\mathrm{N_k}$ kink points are selected, the optimized

trajectory is constructed using $\mathrm{N_k} - 1$ cubic segments of input. Each section of the trajectory is defined by the states

$$[x_k, y_k, \psi_k, \kappa_k, \theta_k, \delta_k, \mathbf{u}_k, \mathbf{u}'_k]_{s=0} \tag{6.19}$$

and input $\mathbf{u}''_k$ with the length of the section s_k, where k indicates the segment. The optimized trajectory for a section of the sparsified trajectory is calculated iteratively using the following discrete system of equations:

$$\mathbf{u}_{i+1} = \mathbf{u}'_i + \mathbf{u}''_i \delta s, \tag{6.20a}$$

$$\mathbf{u}'_{i+1} = \mathbf{u}_i + \mathbf{u}'_i s + \frac{1}{2}\mathbf{u}''_i \delta s^2, \tag{6.20b}$$

$$\mathbf{x}_{i+1} = f(\mathbf{x}_i, \mathbf{u}_i), \tag{6.20c}$$

where δs is the discretization step, and $\mathbf{u}''_\mathrm{s}$ is the system input determined using the relation $\mathbf{u}'' = \mathcal{D}_2 \boldsymbol{u}^*$ for the kink points.

6.3.2. Applications and Outlook

The system dynamics model for one-ST is defined for the trailer using the clothoid equations of motion, with the steering angle and hitch angle calculated using differential flatness. If steering and hitch angles are not considered, the optimization method can still efficiently compute a clothoid trajectory or a higher-order continuous trajectory by adjusting the optimization objective in eq. (6.21), as explained in [Kum22b]. Unlike iterative methods that fit G^1 splines between consecutive points, this

algorithm solves the problem for the entire horizon as a single optimization problem. The optimization problem is defined by:

$$\begin{aligned}
&\min_{\mathbf{u}} \beta_3\|\mathcal{D}_3\mathbf{u}\|_0 + \beta_2\|\mathcal{D}_2\mathbf{u}\|_0 + \beta_1\|\mathcal{D}_1\mathbf{u}\|_0 \\
&\text{s.t.}_{\forall i\in[1,\mathrm{N}]} \quad \mathbf{x}_{i+1} = f(\mathbf{x}_i, \mathbf{u}_i), \\
&|x_i - x_{r,i}| \leq e_x, \\
&|y_i - y_{r,i}| \leq e_y, \\
&|\psi_i - \psi_{r,i}| \leq e_\psi, \\
&|\kappa| \leq \kappa_{\max}, \\
&|\alpha_i| \leq \alpha_{\max}.
\end{aligned} \tag{6.21}$$

where $\mathbf{x} = [x, y, \psi, \kappa]^{\mathrm{T}}$ and $\mathbf{u} = \alpha$, i.e., sharpness of the trajectory and f defines the system dynamics of a clothoid.

Table 6.2.: Type of continuity achieved by adjusting objective

β_1	β_2	β_3	Trajectory Type
0	0	1	Cubic Curvature G^4
0	1	0	Quadratic Curvature G^3
1	0	0	Clothoid Trajectory G^2

As per the table 6.2, the method finds various applications beyond trajectory optimization for truck-trailer systems. Sparsified trajectories can help efficiently store offline computed trajectories and facilitate communication between trajectory planning and control modules. It was demonstrated in [Kum22b] that a relatively large DGNSS trajectory can be effectively represented with a compact number of waypoints without losing details. Additionally, this method has practical applications in compactly storing road information for autonomous driving. By efficiently computing and representing paths, it is possible to create high-definition (HD) maps that store extensive road network data in a more manageable format. This compact representation aids in efficient data

storage and communication between modules in autonomous driving systems.

[Gal22] used a piece-wise clothoid fit for road line markings, demonstrating the utility of clothoidal mapping for HD maps. However, the developed method in this work offers the advantage of solving the entire trajectory optimization as a single problem, avoiding the limitation of piece-wise fitting approaches.

[Zha24a] employed a linearized clothoid model to map the roads as clothoid segments and used the system model derived in [Lim17]. The approximation error in such linearized representations can be significant. This work has demonstrated the method's effectiveness in reducing this error, showing that the approximated model is precise and that the deviation between the approximated and actual models converges [Kum22b]. The results presented in the next section highlight this modeling accuracy even for a complex truck-trailer system.

6.3.3. Results and Discussions

This section presents a comprehensive analysis of the optimization results for the one-ST system, focusing on various aspects of the trajectory optimization process. First, the optimization of a G^1 figure-of-eight trajectory for an one-ST system is examined. This reference trajectory is meticulously selected, with a figure-of-eight pattern chosen for its continuous heading and discontinuous curvature. The trajectory mimics a Dubins path, consisting of two straight lines and two circular arcs, providing a challenging test case for the optimization method. This reference highlights the optimizations' ability to handle paths with significant curvature changes and transitions between trajectory segments.

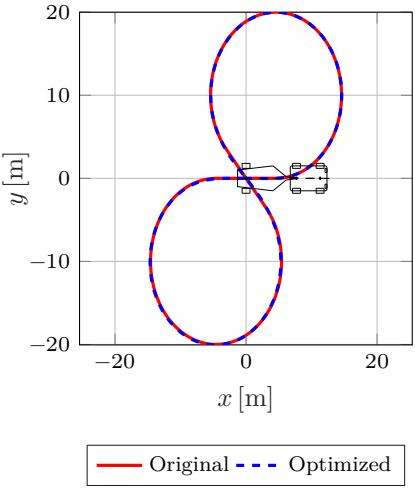


Figure 6.21.: Original vs Optimized Trajectory. The figure demonstrates the reference trajectory and the optimized trajectory to illustrate the improvement.

Next, the influence of the optimized hitch angle for an one-ST system on the behavior of the one-GT system is explored. This analysis highlights the necessity of optimizing paths specifically for the one-GT system to ensure stable and accurate trajectory tracking. The trajectory optimization results for the one-GT system are then presented, demonstrating the effectiveness of the optimization method in producing feasible and smooth paths for complex one-GT systems.

Furthermore, the impact of varying steering angles and their rate constraints on the optimization results is investigated. This analysis provides insights into how different constraints affect trajectory feasibility and system performance.

Finally, results for optimizing a trajectory computed by a planner, which consists of both forward and reverse maneuvers, are provided. These results demonstrate the versatility of the optimization method in handling different driving scenarios.

Trajectory Optimization for one-ST

Figure 6.21 illustrates the original reference trajectory and the optimized trajectory. The optimization process results in a trajectory that maintains the state deviations in the allowed tolerance from the reference but also ensures a smoother transition, making it more feasible. Figure 6.22 illustrate the deviations in the (x,y) coordinates. These figures highlight the actual and approximated deviation in optimization between the reference and optimized trajectories, demonstrating that both remain within the defined tolerance.

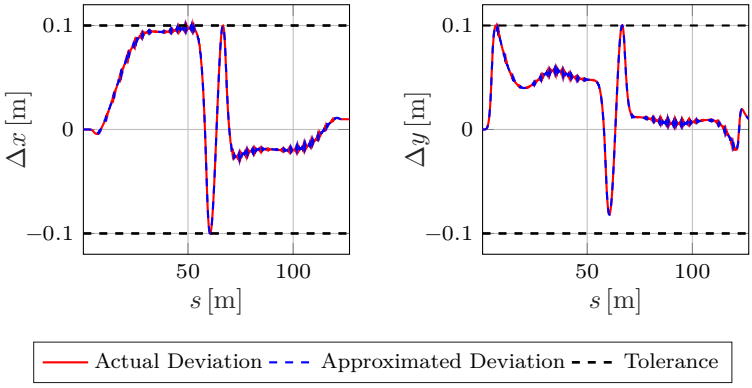


Figure 6.22.: This figure highlights the actual and approximated deviation in the (x,y) coordinates for optimization of one-ST.

The similarity between the approximated deviations obtained during the optimization process and the actual deviations observed validates the convergence and effectiveness of the algorithm. This close match indicates that the optimization method reliably produces accurate and feasible paths that adhere to the specified constraints and objectives.

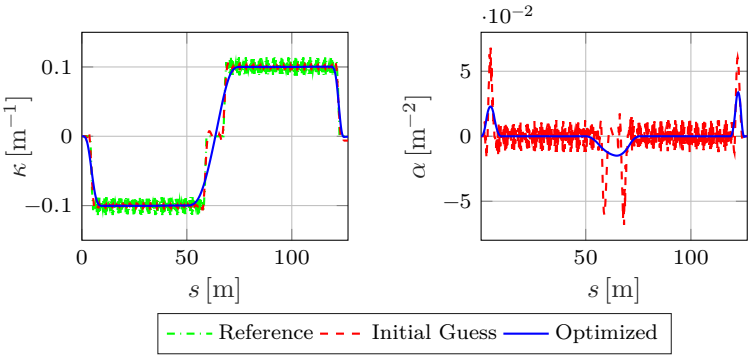


Figure 6.23.: The figure compares the reference, initial guess, and optimized for curvature, and rate change of curvature.

Figure 6.23 compares the reference, initial, and optimized curvature (κ). This analysis helps to understand the changes made during the optimization to achieve a smoother path. Figure 6.23 demonstrates the comparison between the initial and optimized sharpness, while Figure 6.24 illustrate the first and second derivatives of the optimized sharpness. These figures provide insights into resulting trajectory smoothness.

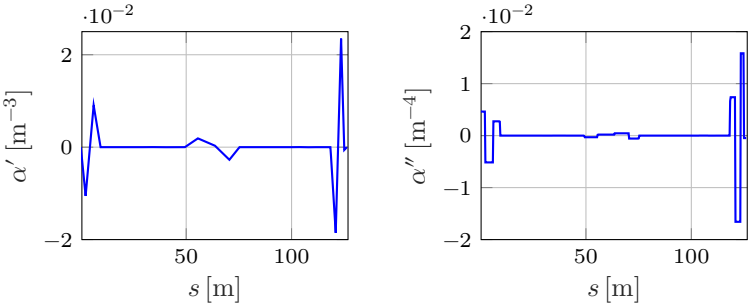


Figure 6.24.: This figure demonstrates the first and second derivatives of the optimized alpha values.

Figure 6.25 illustrates the hitch and steering angles in degrees. The figure highlights the continuity of the resulting steering wheel angle.

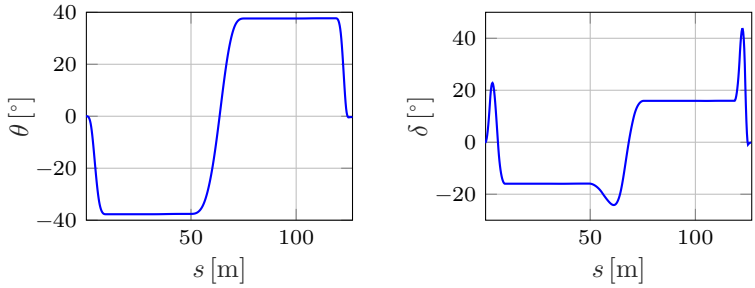


Figure 6.25.: The figure demonstrates the optimized hitch(θ) and steering(δ) angle in degrees for one-ST.

Finally, Figure 6.26 illustrates the reference, optimized, and truck trajectories.

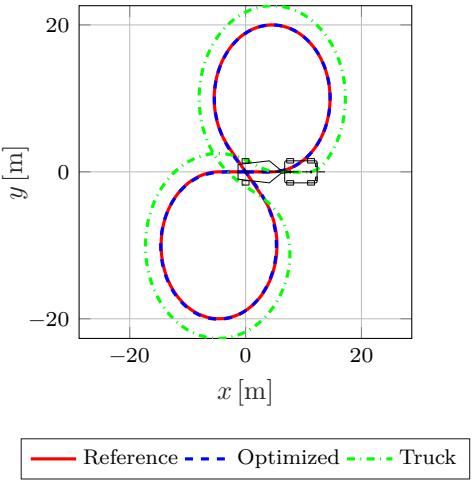


Figure 6.26.: The figure demonstrates reference trajectory, optimized trailer's, and the truck's trajectory for a one-ST.

The results presented in this section demonstrate the effectiveness of the optimization algorithm for one-ST. The optimization improved the

feasibility and drivability of the trajectory for a one-ST by keeping the vehicle actuation constraints within the permissible operational range.

Trajectory Optimization for one-GT

The section first compares the one-GT and one-ST systems. A common assumption in path planning and optimization for the one-GT system is that the trailer connects at the truck's rear axle, i.e., as an one-ST system. The states of the one-GT system are examined as it follows the optimized hitch path for an one-ST system. The following figures illustrate the optimized trajectory for an one-ST system followed by different configurations of the one-GT system. Figure 6.27 demonstrates the trajectories for various configurations of the one-GT.

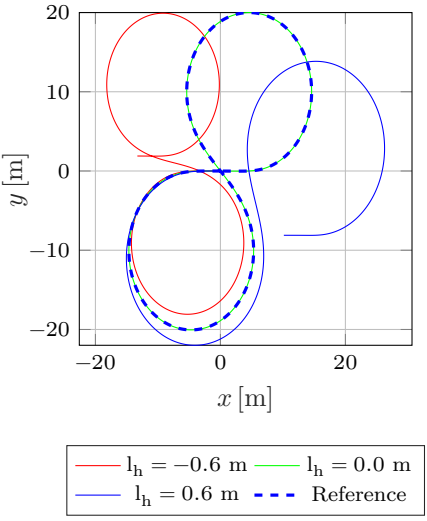


Figure 6.27.: Trajectories of one-GT for different configurations. The figure demonstrates how the flatness-based trajectory is followed by different configurations of one-GT.

The trajectory varies significantly even for small hitch distances, such as 0.6 meters, for negative and positive off-axle hitches. Although stabilization of the one-GT on the optimized path for an one-ST is possible using trajectory tracking control, the hitch and steering constraints might not be satisfied. Therefore, trajectory optimization for the one-GT system improves feasibility and drivability. Next, the results for trajectory optimization of the one-GT system will be presented.

Figure 6.28 illustrates the original reference trajectory, the optimized trajectory, and the initial guess. The optimization can converge to an optimal solution even if the initial guess is far from the reference trajectory. The introduction of slack variables in the optimization problem handles the initial deviation and iteratively converges to optimal zero. Without the slack variables, the initial guess influences the optimization's success.

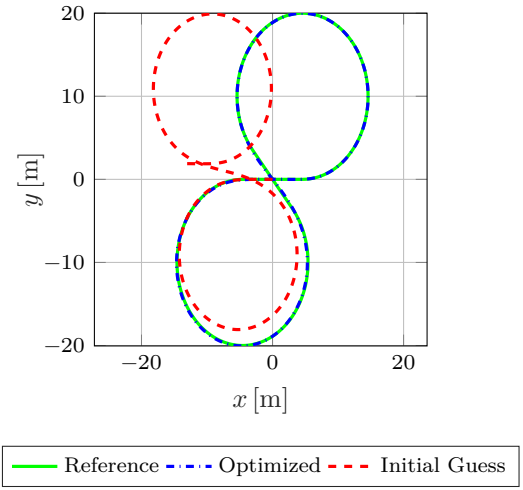


Figure 6.28.: The figure illustrates reference, optimized trajectory, and initial guess for one-GT.

Figure 6.29 illustrates the deviations in the (x,y) coordinates. The deviations within the allowed tolerance affirm the optimization's ability

to converge to an optimal solution. While a good initial guess speeds up the optimization process, the algorithm is robust enough to handle a poor initial guess.

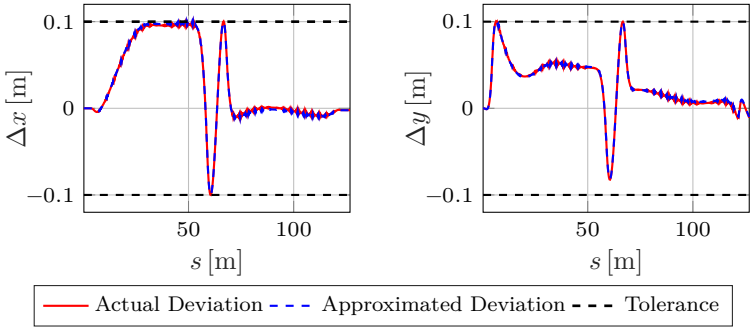


Figure 6.29.: The figure highlights the actual and approximated deviations in the x,y coordinates for one-GT.

Figure 6.30 demonstrates how the rate change of hitch γ changes from their initial state to their optimized state and it's first derivative. These figures provide insights into the optimization process and the resulting trajectory smoothness. The rate change of hitch γ is crucial as it influences the vehicle's steering behavior. The optimized γ values result in a smoother and more accurate path, indicating the effectiveness of the optimization process. A continuous first derivative implies less abrupt changes in steering, contributing to a more stable and controlled vehicle trajectory.

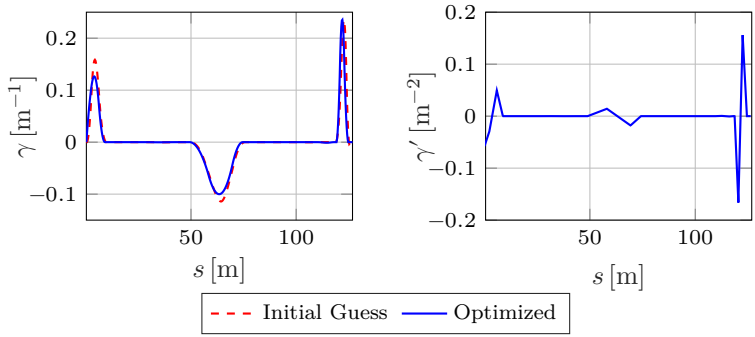


Figure 6.30.: The figure illustrates the initial and optimized rate change of hitch angle γ and the difference in the rate change of the hitch angle between two consecutive steps for one-GT.

Figure 6.31 illustrates the initial and optimized hitch and steering angles in degrees.

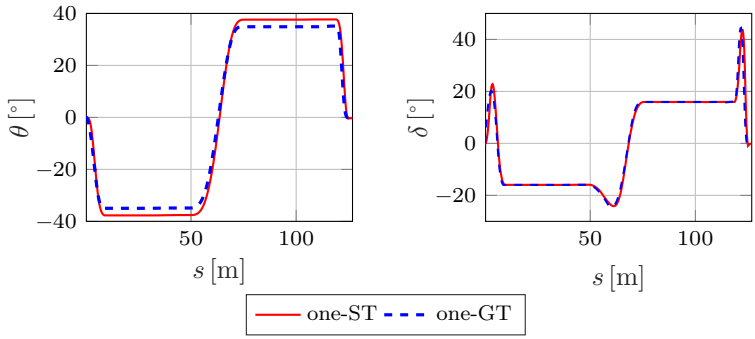


Figure 6.31.: The figures illustrate the reference and optimized hitch(θ) and steering(δ) angles in degrees for one-GT

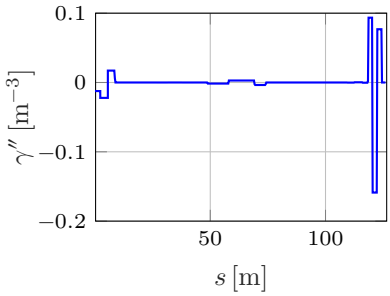


Figure 6.32.: The figure illustrates the initial and optimized second derivative of the hitch angle rate γ for one-GT.

Figure 6.32 illustrates the initial and optimized second derivative of the hitch angle rate γ for the one-GT, as well as the difference in the second derivative between two consecutive steps.

The figure demonstrates a profile similar to a bar diagram, where each bar represents a trajectory section. The corners of these bars, known as kink points. Each bar's width corresponds to the trajectory length that can be defined as one segment. It implies that the trajectory between two kink points can be represented as a single section, simplifying the trajectory representation.

The results presented in this section demonstrate the effectiveness of the optimization algorithm in improving the feasibility of trajectory for one-GT.

Trajectory Optimization with Varying Constraints

The optimization process effectively manages varying constraints, as demonstrated in Figure 6.33. This figure highlights the deviations in the x and y coordinates for two optimizations with different steering angle constraints for the one-GT. The figures illustrate how the optimized trajectory closely follows the reference path, ensuring minimal deviations and high accuracy in trajectory planning.

Figure 6.34 and Figure 6.35 illustrate the reference and optimized hitch θ and steering δ angles, as well as their rates, for the one-GT. The figures represent the results of two optimization scenarios with different steering constraints: 30° and 40°. These figures demonstrate how the optimization process adapts to different constraints, optimizing the trajectory while respecting the given limits.

The results highlight the algorithm's capability to manage varying steering constraints, maintaining smooth and feasible trajectories. By comparing the reference and optimized angles, the optimization's adjustments to achieve the best possible trajectory within the specified constraints can be observed. This flexibility is crucial for practical applications where different vehicles and scenarios may impose varying limitations.

In conclusion, the optimization effectively handles different steering constraints, ensuring accurate and feasible trajectories for the one-GT. This adaptability demonstrates the robustness of the trajectory optimization, making it suitable for a wide range of practical applications.

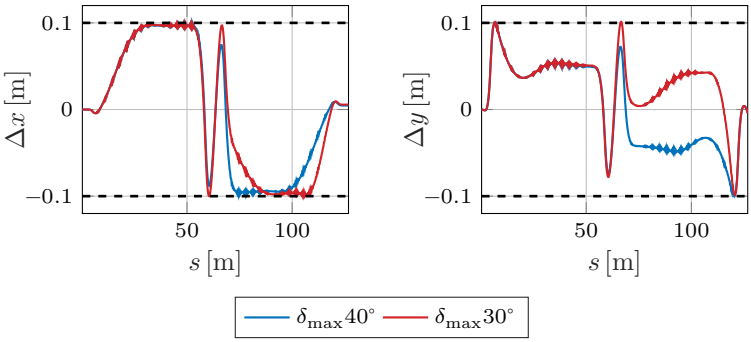


Figure 6.33.: The figure highlights the actual and approximated deviations in the x, y coordinates for one-GT.

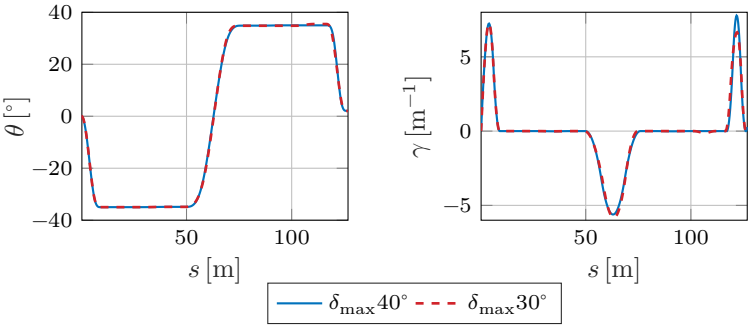


Figure 6.34.: The figures illustrate the reference and optimized hitch θ and steering δ angles in degrees for one-GT.

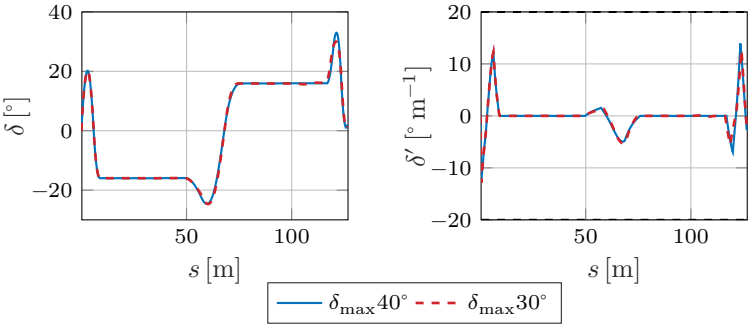


Figure 6.35.: The figures illustrate the reference and optimized steering δ angles and their rates in degrees for one-GT.

Optimization of Planned Trajectory

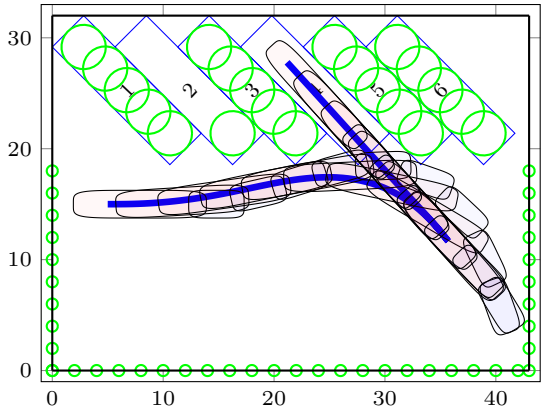


Figure 6.36.: The figure illustrates the optimized trajectory for the one-GT during an angled parking maneuver. The area swept by the optimized trajectory is also shown, highlighting that the trajectory is collision-free in the vicinity of the original reference.

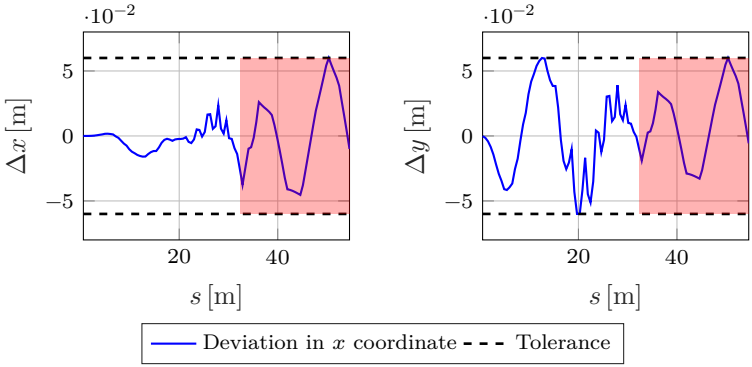


Figure 6.37.: The figure highlights the actual and approximated deviations in the x,y coordinates between the optimized and original reference for one-GT.

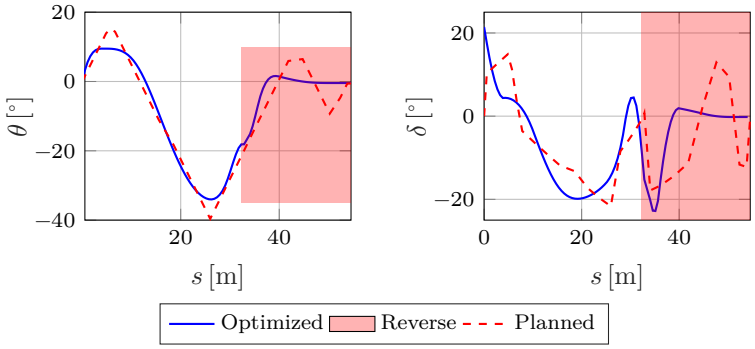


Figure 6.38.: The figures illustrate the planned and optimized hitch (θ) and steering (δ) angles in degrees for one-GT.

The figures in this section demonstrate the results of optimizing a planned trajectory using the trajectory planner described in this work. Figure 6.36 illustrates the optimized trajectory for the one-GT during an angled parking maneuver. The area swept by the optimized trajectory is shown, highlighting that the trajectory remains collision-free near the original reference. The trajectory includes forward and reverse driving maneuvers, demonstrating optimization robustness in handling complex parking scenarios.

Figure 6.37 highlights the actual and approximated deviations in the x, y coordinates between the optimized and original reference trajectories for the one-GT. These figures demonstrate the ability of the optimization process to maintain the trajectory within a narrow tolerance around the planned path, ensuring collision avoidance.

Figure 6.38 illustrates the planned and optimized hitch (θ) and steering (δ) angles in degrees for the one-GT. The optimization improves the overall feasibility and smoothness of the path, adjusting the hitch and steering angles to create a more practical and navigable trajectory.

In conclusion, trajectory optimization significantly enhances the feasibility and smoothness of the path while ensuring it remains collision-free.

The optimization process improves path quality by maintaining the trajectory within a narrow tolerance around the planned path. However, conducting an additional collision check after optimization can further enhance safety by addressing any potential local collisions that may arise during the optimization process.

6.4. Summary

This chapter addressed the challenges of trajectory planning and optimization for truck-trailer systems, focusing on one-GT. It presented a comprehensive approach to trajectory planning, incorporating the complexities of truck-trailer dynamics, non-holonomic constraints, and collision constraints. The chapter further introduced a trajectory optimization method aimed at achieving high-order continuity (G^4) for the planned paths, ensuring smooth and feasible trajectories.

The key contributions of this chapter are:

- Application of partial differential flat model for trajectory planning and optimization of one-GT.
- Optimal trajectory planning framework for one-GT systems.
- Trajectory optimization method achieving G^4 continuity for smooth and feasible paths.

7. Generalization for n-General Trailer (n-GT)

In this work, the development of trajectory tracking planning and control methods has primarily focused on One-General Trailer (one-GT) configurations. However, many practical applications, particularly logistics and transportation, involve complex n-GT systems. The ability to generalize control and planning strategies to n-General Trailer (n-GT) systems would significantly enhance the utility and flexibility of the developed methods, enabling their application to a broader range of real-world scenarios.

The primary objective of this chapter is to generalize the concept of partial differential flatness and the stable flat model developed for a one-GT to the n-General Trailer (n-GT). Then, the chapter would like to extend the Lyapunov Control (LC) and trajectory optimization based on a partial flat model of two-GT. After the Truck Semitrailer, the Two-General Trailer (two-GT) system, commonly known as the Truck-Dolly-Trailer, is a widely used configuration in logistics. In this setup, the dolly is connected off-axle to the truck, and the trailer is connected to the dolly. In the mathematical modeling, the dolly represents the first trailer and the system is represented by a two-GT.

7.1. Partial Flatness for n-GT

The n-General Trailer (n-GT) is inherently not differentially flat. The concept of partial differential flatness was explored in section 2.1.2 and proven to be applicable in section 3.4 for a one-GT.

The concept of partial differential flatness, as explored in section 2.1.2, also finds application in the n-General Trailer (n-GT) system. This subsection proposes a partially differential flat model for n-GT; this approach introduces the flat state $\mathbf{s} = [\theta_1, \theta_2, \theta_3, ..., \theta_\mathrm{n}]^\mathrm{T}$, where θ_j represents the hitch angle between the j-th and (j+1)-th trailers. The remaining system states are designated as the non-flat state $\mathbf{r} = [x_\mathrm{n}, y_\mathrm{n}, \psi_\mathrm{n}]^\mathrm{T}$, where x_n and y_n denote the position coordinates of the n-th trailer, and ψ_n represents its heading angle.

The evolution of the non-flat state $\mathbf{r}$, given by $\mathbf{r}' = f(\mathbf{s}, \mathbf{r})$, allows for the determination of all $\mathbf{r}$ states through integration, provided that initial conditions are known. As an example, this work will now derive a flat model and extend the trajectory planning for two-GT systems.

7.2. Two-General Trailer (two-GT): System Modeling

In this section, the Truck-Dolly-Trailer system, a type of Two-General Trailer (two-GT) system as illustrated in fig. 7.1, will be discussed. The state vector is defined as $\mathbf{x} = [x_2, y_2, \psi_2, \psi_1, \psi_0]$, with the input $\mathbf{u} = \delta$.

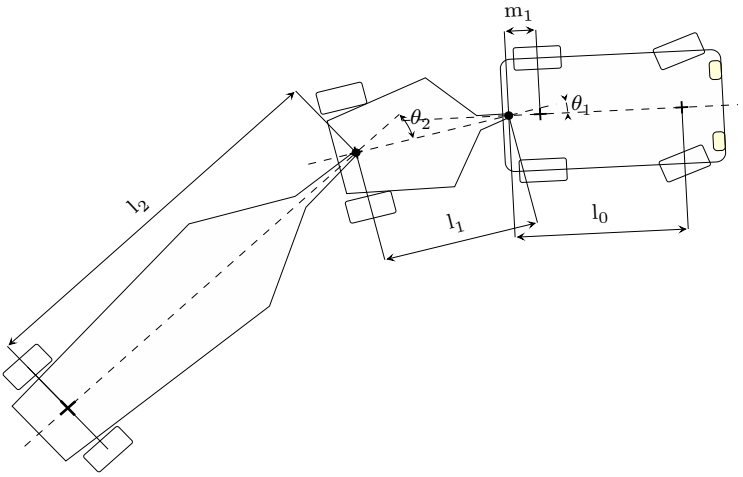


Figure 7.1.: Geometric illustration of a Two-General Trailer (two-GT). The parameters l_2 and l_1 represent the lengths of the trailers, while m_1 denote the distances between the axle of the dolly and its respective coupling points.

In this context, the state vector $\mathbf{x} = [x_2, y_2, \psi_2, \psi_1, \psi_0]$ represents the following components:

- x_2: The x-coordinate of the trailer.
- y_2: The y-coordinate of the trailer.
- ψ_2: The orientation of the trailer.
- ψ_1: The orientation of the dolly.
- ψ_0: The orientation of the truck.

Here, the subscript $_2$ denotes variables associated with the trailer, $_1$ denotes variables associated with the dolly, and $_0$ denotes variables associated with the truck. In the truck-dolly-trailer configuration, the trailer

is connected to the dolly's axle; if the trailer is hitched off the axle, the equation will be more complex, but the partial flatness will remain valid.

The input $\mathbf{u} = \delta$ represents the steer angle δ, which is the control input for the system.

The hitch angles is given as follows:

$$\theta_1 = \psi_0 - \psi_1, \tag{7.1}$$
$$\theta_2 = \psi_1 - \psi_2. \tag{7.2}$$

Here, θ_1 is the hitch angle between the truck and the dolly, and θ_2 is the hitch angle between the dolly and the trailer.

The velocities' relationship can be established using the chain form eq. (3.2). The truck and dolly velocities have been derived as functions of the trailer velocity:

$$v_0 = \frac{v_2 \mathrm{l}_0}{\cos\theta_2 \left(\mathrm{l}_0 \cos\theta_1 + m_1 \tan\delta \sin\theta_1\right)}, \tag{7.3}$$
$$v_1 = \frac{v_2}{\cos\theta_2}. \tag{7.4}$$

Finally, the spatial system dynamics $\mathbf{x}'$ is derived using eq. (3.1), and is given by:

$$\mathbf{x}' = \begin{bmatrix} x_2' \\ y_2' \\ \psi_2' \\ \psi_1' \\ \psi_0' \end{bmatrix} = \begin{bmatrix} \cos\psi_2 \\ \sin\psi_2 \\ \frac{\tan\theta_2}{l_2} \\ \frac{l_0 \sin\theta_1 - m_1 \tan\delta\cos\theta_1}{l_1 \cos\theta_2 (l_0 \cos\theta_1 + m_1 \tan\delta \sin\theta_1)} \\ \frac{\tan\delta}{\cos\theta_2 (l_0 \cos\theta_1 + m_1 \tan\delta \sin\theta_1)}. \end{bmatrix} \tag{7.5}$$

7.3. Partial Flatness: two-GT

The concept of partial differential flatness, as explored in section 7.1, is applicable for n-GT system. This subsection derives a partially differential flat model for truck-dolly-trailer configurations.

The system state are divide as the flat state $\mathbf{s} = [\theta_1, \theta_2]^{\mathrm{T}}$ while designating the remaining system states as the non-flat state $\mathbf{r} = [x_2, y_2, \psi_2]^{\mathrm{T}}$.

Utilizing eq. (7.5) to substitute expressions for ψ_1' and ψ_2', the rate of change of the hitch angle, θ_2, is derived as follows:

$$\theta_2' = \psi_1' - \psi_2'. \tag{7.6}$$

The derivative of the hitch angle, γ_2, is given by:

$$\gamma_2 = \frac{\mathrm{l}_0 \sin\theta_1 - m_1 \tan\delta \cos\theta_1}{\mathrm{l}_1 \cos\theta_2 \left(\mathrm{l}_0 \cos\theta_1 + m_1 \tan\delta \sin\theta_1\right)} - \frac{\tan\theta_2}{\mathrm{l}_2}. \tag{7.7}$$

The steering angle, δ, can then be formulated as a function of the hitch angle rates, γ_2, and the hitch angles, θ_1 and θ_2:

$$\delta = \arctan\left(\frac{\mathrm{l}_0 \left(\mathrm{l}_2 \sin\theta_1 - \mathrm{l}_1 \cos\theta_1 \sin\theta_2 - \gamma_2 \mathrm{l}_1 \mathrm{l}_2 \cos\theta_1 \cos\theta_2\right)}{m_1 \left(\mathrm{l}_2 \cos\theta_1 + \mathrm{l}_1 \sin\theta_1 \sin\theta_2 + \gamma_2 \mathrm{l}_1 \mathrm{l}_2 \cos\theta_2 \sin\theta_1\right)}\right). \tag{7.8}$$

The evolution of the non-flat state $\mathbf{r}$, given by $\mathbf{r}' = f(\mathbf{s},\mathbf{r})$, allows for the determination of all $\mathbf{r}$ states through integration, provided that initial conditions are known. The flat model of the system can be derived using the eq. (7.5) and substituting steering angle δ using the relation derived in eq. (7.8). The flat model states is defined as $\mathbf{x} = [x_2, y_2, \psi_2, \psi_1, \psi_0]$ and input as $\mathbf{u} = \gamma_2$.

7.4. Lyapunov Control Law: two-GT

Utilizing Lyapunov stability criteria as discussed in section 2.1.3, a Lyapunov function, $V(\theta) = \frac{1}{2}(\theta_2 - \vartheta)^2$, is introduced to assess stability similar to two-GT discussed in section 4.1.

Here, ϑ represents the reference angle, and θ_2 signifies the hitch angle between dolly and trailer.
The system is stable within the region $\mathcal{D} = [\theta_{\min},\theta_{\max}] \in \mathrm{R}$ if $V(\theta_2) > 0$ for $\theta_2 \neq \vartheta$ and $V(\theta_2) = 0$ at the equilibrium point $\theta_2 = \vartheta$.

To achieve asymptotic stability within $\mathcal{D}$, the derivative of $V(\theta_2)$, denoted as $V'(\theta_2)$, must be negative for all $\theta_2 \neq \vartheta$, resulting in the control law for θ_2' as:

$$\theta_2' = \vartheta' - K(\theta_2 - \vartheta), \tag{7.9}$$

where K is a positive proportional control gain. The resulting $V'(\theta_2) = -K(\theta_2 - \vartheta)^2$ is always negative for all positive K and $\theta \neq \vartheta$.

The steering angle input, δ, for the Two-General Trailer (two-GT) system is given in eq. (7.8) as:

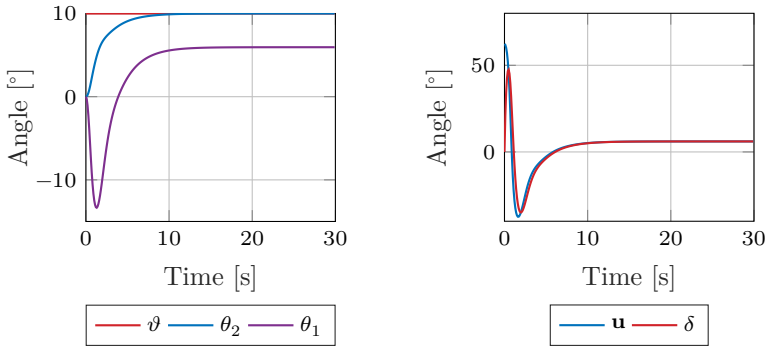
$$\delta = \arctan\left(\frac{l_0\left(l_2 \sin\theta_1 - l_1 \cos\theta_1 \sin\theta_2 - \gamma_2 l_1 l_2 \cos\theta_1 \cos\theta_2\right)}{m_1\left(l_2 \cos\theta_1 + l_1 \sin\theta_1 \sin\theta_2 + \gamma_2 l_1 l_2 \cos\theta_2 \sin\theta_1\right)}\right), \tag{7.10}$$

with $\gamma_2 = \vartheta' - K(\theta_2 - \vartheta)$. The control strategy dynamically adjusts for errors between the current and desired hitch angles. The control approach will track the hitch angle rate as the control error $(\theta_2 - \vartheta)$ diminishes to zero.

7.4.1. Experimental Setup and Results

Similarly to section 4.1, the concept of Lyapunov control is extended to the Truck-Dolly-Trailer system. The simulation setup is similar to

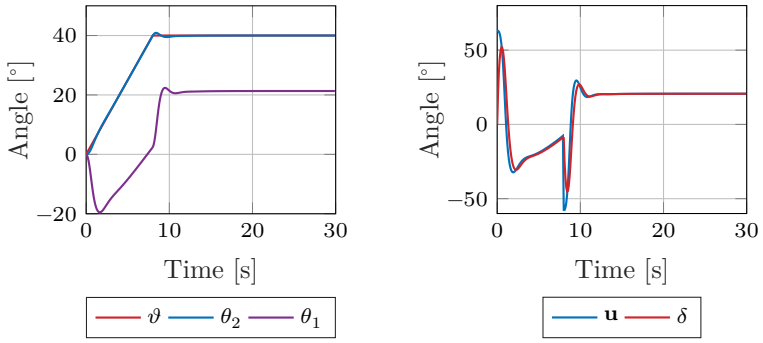
section 4.3 for the Truck-Dolly-Trailer system. The control has been validated for step, ramp, and sinusoidal inputs.



(a) Hitch angles θ_2, θ_1. **(b)** Control input and steering angle δ.

Figure 7.2.: Comparative analysis of hitch angles θ_2 and θ_1 response and control input u for step reference hitch angle input in the Truck-Dolly-Trailer system.

The figures illustrate the response of a Truck-Dolly-Trailer system to various reference hitch angle inputs and the corresponding control efforts. Figure 7.2 demonstrates the system's response to a step change in the reference hitch angle θ. Figure 7.2a depicts the hitch angle of the dolly and trailer, demonstrating how closely the actual angles follow the step input. Figure 7.2b presents the combined control input effort and the steering angle δ, highlighting the system's adjustments to maintain the desired hitch angle.

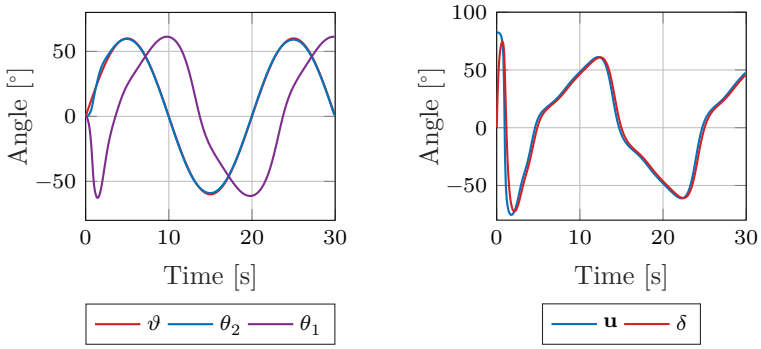


(a) Hitch angles θ_2, θ_1.

(b) Control input and steering angle δ.

Figure 7.3.: Comparative analysis of hitch angles θ_2 and θ_1 response and control input u for ramp reference hitch angle input in the Truck-Dolly-Trailer system.

Similarly, Figure 7.3 presents the system's response to a ramp change in the reference hitch angle. Figure 7.3a demonstrates the hitch angles for both the dolly and the trailer, while Figure 7.3b displays the control input and steering angle required to track the ramp input. These results illustrate the system's ability to handle gradually increasing reference angles.



(a) Hitch angles θ_2, θ_1. **(b)** Control input and steering angle δ.

Figure 7.4.: Comparative analysis of hitch angles θ_2 and θ_1 response and control input u for sinusoidal reference hitch angle input in the Truck-Dolly-Trailer system.

Finally, Figure 7.4 illustrates the system's response to a sinusoidal reference hitch angle input. Figure 7.4a figures the hitch angles, demonstrating the system's capacity to follow a continuously varying reference. Figure 7.4b demonstrates the control inputs and steering angles used to achieve this tracking. These figures collectively demonstrate the effectiveness of the control strategies in maintaining desired hitch angles under different input conditions in the Truck-Dolly-Trailer system.

7.5. Trajectory Planning: two-GT

Trajectory planning for the two general trailers (two-GT) system is formulated similarly to the one-trailer (one-GT) system, as described in section 6.1. Most of the optimization problem remains the same, with some additional costs and constraints to account for the dolly. The dolly is modeled as a superellipse, and collision constraints are applied accordingly.

The state vector for the two-GT system is defined as:

$$\mathbf{x}_i = [x_{2,i}, y_{2,i}, \psi_{n,i}, \theta_{2,i}, \theta_{1,i}, \delta_i] \in \mathbb{R}^6, \tag{7.11}$$

where $x_{2,i}$ and $y_{2,i}$ represent the position of the n-th trailer at time step i, $\psi_{n,i}$ is the heading angle, $\theta_{2,i}$ and $\theta_{1,i}$ are the hitch angles for the second and first trailers, respectively, and δ_i is the steering angle.

The inputs to the system are:

$$\mathbf{u}_i = [\gamma_{2,i}, ds_{2,i}] \in \mathbb{R}^2, \tag{7.12}$$

where $\gamma_{2,i}$ is the control input rate change of hitch angle for the second trailer and $ds_{2,i}$ denotes the distance traveled by the second trailer at time step i.

The dynamics of the two-GT system were derived using the continuous flat model defined in section 7.3 and discretized using the fourth-order Runge-Kutta (RK4) integration method, resulting in the following discrete-time equations:

$$\mathbf{x}_{i+1} = f(\mathbf{x}_i, \mathbf{u}_i), \tag{7.13}$$

where f represents the system's dynamic equations of the flat model of the truck, dolly, and trailer system.

The initial state $\mathbf{x}_0 \in \mathbb{R}^6$ is defined as the starting position of the system:

$$\mathbf{x}_0 = \mathbf{x}(0). \tag{7.14}$$

To ensure that the system reaches a goal position and orientation, the goal state $\mathbf{x}_\mathrm{N} \in \mathbb{R}^6$ is defined. To achieve this, a terminal cost J_f is introduced:

$$J_\mathrm{f} = (\mathbf{x}_\mathrm{N} - \mathbf{x}_g)^\mathrm{T} Q_\mathrm{f} (\mathbf{x}_\mathrm{N} - \mathbf{x}_g), \tag{7.15}$$

where $\mathbf{x}_{\mathrm{N}}$ is the state of the system at the final time step N, $\mathbf{x}_g \in \mathbb{R}^6$ is the goal state, and $Q_{\mathrm{f}} \in \mathbb{R}^{6\times6}$ is a weight matrix that defines the importance of reaching the goal state accurately. In addition to the terminal cost, a terminal constraint is imposed to ensure that the final state of the system is at the goal state or within an acceptable tolerance range:

$$\|\mathbf{x}_{\mathrm{N}} - \mathbf{x}_g\| \leq \epsilon. \tag{7.16}$$

The running cost penalizes deviations in the system's state and inputs at each time step, ensuring smooth transitions and avoiding large deviations. The running cost for the states J_{r} is defined as:

$$J_{\mathrm{r}} = \sum_{i=0}^{\mathrm{N}-1} \left(Q_{\theta_1}\theta_{1,i}^2 + Q_{\theta_2}\theta_{2,i}^2 + Q_{\delta}\delta_i^2 \right), \tag{7.17}$$

where $Q_{\theta_1} \in \mathbb{R}$, $Q_{\theta_2} \in \mathbb{R}$, and $Q_{\delta} \in \mathbb{R}$ are weight coefficients for $\theta_{1,i}$, $\theta_{2,i}$, and δ_i, respectively. Similarly, the running cost for the inputs J_{u} and the difference in the inputs $J_{\Delta\mathrm{u}}$ are defined as:

$$J_{\mathrm{u}} = \sum_{i=0}^{\mathrm{N}-1} \mathbf{u}_i^{\mathrm{T}} \mathrm{R}_{\mathrm{u}} \mathbf{u}_i. \tag{7.18}$$

$$J_{\Delta\mathrm{u}} = \sum_{i=0}^{N-2} (\mathbf{u}_{i+1} - \mathbf{u}_i)^{\mathrm{T}} \mathrm{R}_{\Delta\mathrm{u}} (\mathbf{u}_{i+1} - \mathbf{u}_i), \tag{7.19}$$

where $\mathrm{R}_{\mathrm{u}} \in \mathbb{R}^{2\times2}$ is the weight matrix for the control input, and $\mathrm{R}_{\Delta\mathrm{u}} \in \mathbb{R}^{2\times2}$ is the weight matrix for the change in control input.

The input constraints ensure that the control inputs remain within their feasible ranges throughout the trajectory:

$$\mathbf{u}_{\min} \leq \mathbf{u}_i \leq \mathbf{u}_{\max}, \quad \mathbf{u}_{\min}, \mathbf{u}_{\max} \in \mathbb{R}^2. \tag{7.20}$$

Additionally, the collision constraint ensures that the system avoids obstacles by maintaining a safe distance from them. This constraint is represented by a function $g(\mathbf{x}_i) > 1$, indicating that the system is outside the collision region:

$$g(\mathbf{x}_i) > 1, \tag{7.21}$$

where $g(\mathbf{x}_i)$ is a function that represents the normalized distance to obstacles at the i-th time step.

The trajectory planning problem is formulated as Nonlinear Programming (NLP) problem and is summarized as follows:

$$\begin{aligned}
\min_{\mathbf{x},\mathbf{u}} \quad & J = J_\mathrm{f} + J_\mathrm{r} + J_\mathrm{u} + J_{\Delta\mathrm{u}} \\
\text{s.t.}_{\forall i \in [0,\mathrm{N}-1]} \quad & \mathbf{x}_0 = \mathbf{x}(0), \\
& \|\mathbf{x}_\mathrm{N} - \mathbf{x}_g\| \leq \epsilon, \\
& \mathbf{u}_{\min} \leq \mathbf{u}_i \leq \mathbf{u}_{\max}, \\
& \theta_{1,\min} \leq \theta_{1,i} \leq \theta_{1,\max}, \\
& \theta_{2,\min} \leq \theta_{2,i} \leq \theta_{2,\max}, \\
& \delta_{\min} \leq \delta_i \leq \delta_{\max}, \\
& g(\mathbf{x}_i) > 1, \quad i = 0, \dots, \mathrm{N}-1 \\
& \mathbf{x}_{i+1} = f(\mathbf{x}_i, \mathbf{u}_i).
\end{aligned} \tag{7.22}$$

The trajectory planning problem was formulated and solved using Casadi, a symbolic framework for numerical optimization [And19].

7.6. Results and Discussions

This section presents the results for the planned trajectories of the two-GT system during diverse parking maneuvers; parallel, perpendicular, and angled parking. The results highlight the area swept to demonstrate that the planned trajectories are collision-free. Additionally, the angles θ_1, θ_2, δ, and γ_2 are examined to ensure that the trajectory planner respects all constraints. Finally, it is demonstrated that the trajectory planner can continuously adjust direction when required, showcasing its ability to navigate tight spaces efficiently.

The model parameters and dimensions used in this study are summarized in table 7.1. It is assumed that the front and rear overhangs are equal. Equal overhangs on both sides are assumed, which can be adapted with actual values as they do not significantly affect optimization.

Table 7.1.: Model parameters and dimensions for Truck-Dolly-Trailer

Parameter	**Description**	**Value (m)**
l_0	Wheelbase of the Truck	4.6
l_1	Length of the Dolly	3.8
l_2	Length of the Trailer	7.6
m_1	Hitch Length for Dolly	0.8
$l_{trailer}$	Trailer's Geometric Length	9.0
l_{truck}	Truck's Geometric Length	6.2
l_{dolly}	Dolly's Geometric Length	3.8
w	Width of Truck	2.5

Table 7.2 summarizes the constraints for the angles θ_1, θ_2, δ, and γ_2 used in the optimization process. These angles are chosen to be within the permissible range for safe and efficient maneuvering of the two-GT system. Specifically, the steering angle constraint of $\pm 42^\circ$ is deduced from the truck's curvature constraint of 0.2m^{-1}.

Table 7.2.: Constraints for θ_1, θ_2, δ, and γ_2

Parameter	Constraint
θ_1	$[-40°, 40°]$
θ_2	$[-40°, 40°]$
δ	$[-42°, 42°]$
γ_2	$[-2.86°\text{m}^{-1}, 2.86°\text{m}^{-1}]$

7.6.1. Parallel Parking

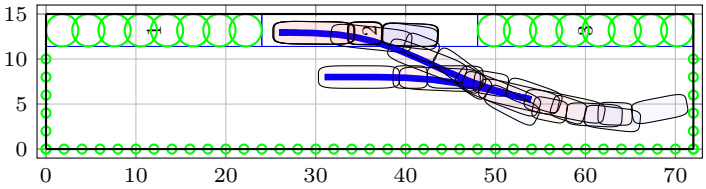


Figure 7.5.: The figure illustrates the area swept by the two-GT system for parallel parking.

In fig. 7.5, the two-GT system successfully optimizes and finds a parallel parking trajectory. The figure demonstrates that the planned reference trajectory is free of collisions, ensuring the trailers navigate into the parking space without any obstacles. The trajectory is optimized to minimizes the angles $[\theta_2, \theta_1, \delta, \gamma_2]$ while maintaining a collision-free path. Figure 7.6 further confirms that the angles θ_1, θ_2, δ, and γ_2 remain within the specified constraints during the maneuver.

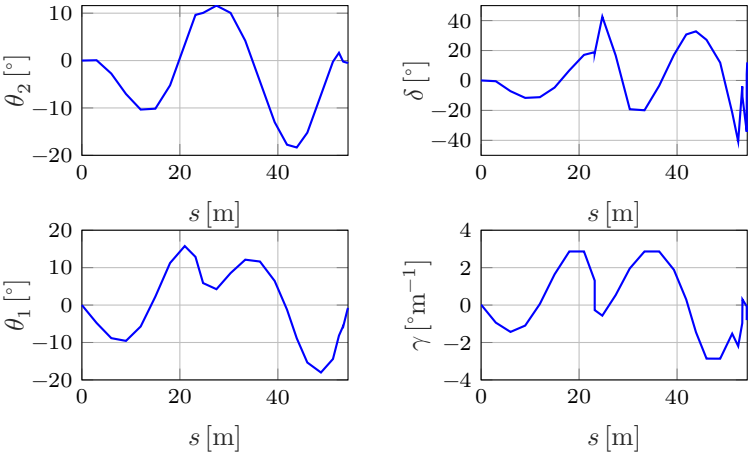


Figure 7.6.: The figure illustrates the angles θ_1, θ_2, δ, and γ_2 for parallel parking, showing that all values remain within constraints.

7.6.2. Angled Parking

Figure 7.7 illustrates the two-GT system performing an angled parking maneuver.

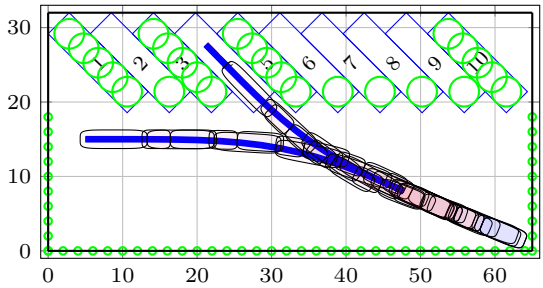


Figure 7.7.: The figure illustrates the area swept by the two-GT system for angled parking.

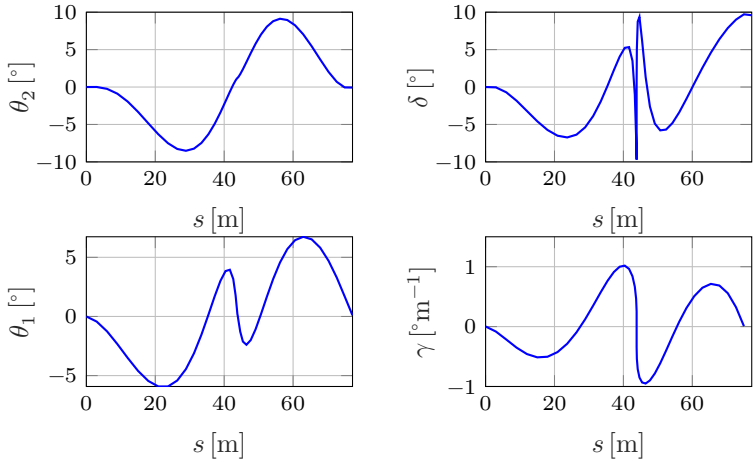


Figure 7.8.: The figure illustrates the angles θ_1, θ_2, δ, and γ_2 for angled parking, showing that all values remain within constraints.

The swept area and the planned reference trajectory indicate that the system effectively manages the spatial constraints of angled parking, ensuring a collision-free maneuver. The trajectory planning accounts for the unique challenges the angle poses, successfully positioning the trailers within the designated space. As shown in fig. 7.8, the angles θ_1, θ_2, δ, and γ_2 remain within constraints, validating the maneuver's feasibility.

7.6.3. Perpendicular Parking

Figure 7.9 demonstrates the area swept by the two-GT system during a perpendicular parking maneuver. The planned trajectory ensures that the trailers are guided precisely into the parking spot. The trajectory planning avoids potential collisions, proving the system's robustness in navigating tight spaces typical of perpendicular parking. Figure 7.10 illustrates that the angles θ_1, θ_2, δ, and γ_2 remain within the constraints.

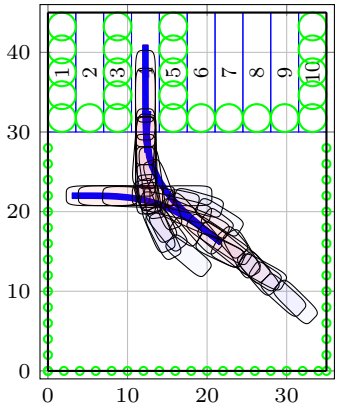


Figure 7.9.: The figure illustrates the area swept by the two-GT system for perpendicular parking.

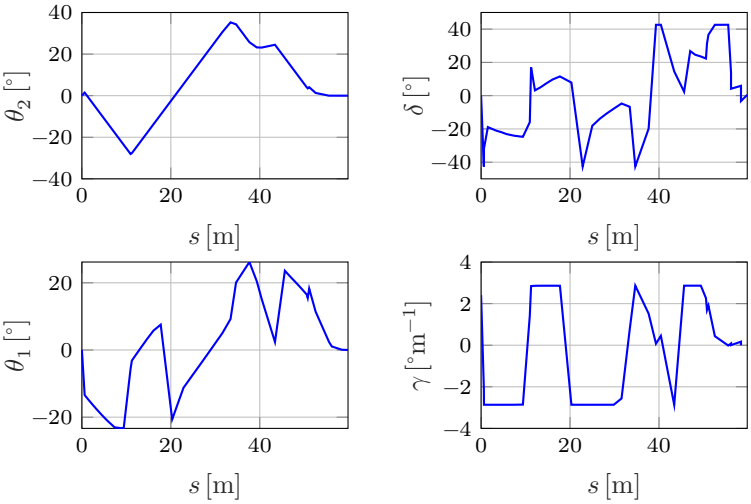


Figure 7.10.: The figure illustrates the angles θ_1, θ_2, δ, and γ_2 for perpendicular parking, showing that all values remain within constraints.

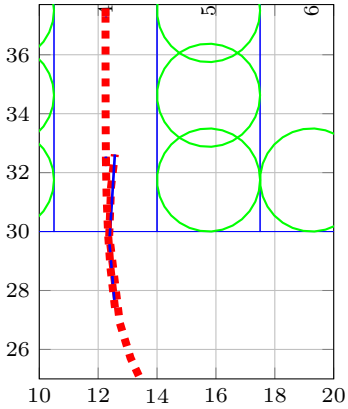


Figure 7.11.: The figure illustrates the optimized trajectory with different line styles for reverse and forward maneuvers during perpendicular parking. The dashed trajectory denotes the reverse maneuver, and the solid trajectory indicates the forward maneuver.

Additionally, fig. 7.11 highlights the optimized trajectory using different line styles for reverse and forward maneuvers, demonstrating the system's ability to adjust direction to navigate tight spaces. This maneuver is a perfect example of how optimization effectively manages direction changes near the entrance to the parking spot. Initially, the trailer moves backward; it then changes direction to move forward and subsequently reverses to manage collision constraints effectively.

Overall, the results provide clear visual evidence that the planned reference trajectories for the two-GT system are effective in various parking scenarios. The system demonstrates high accuracy and reliability, ensuring that the parking maneuvers are free from collisions. The angles θ_1, θ_2, δ, and γ_2 remain within their respective constraints, further validating the feasibility and safety of the maneuvers. This capability is critical for practical applications where precise and safe maneuvering of trailers is essential.

7.7. Summary

This chapter generalized the concept of partial differential flatness to n-General Trailer (n-GT). It then derived a partially flat model for the Two-General Trailer (two-GT) system. The chapter extended the Lyapunov Control (LC) and trajectory planning to the two-GT configuration, utilizing a partial flat two-GT model. Finally, the effectiveness of these methods in handling the complexities associated with multi-trailer systems was demonstrated, showcasing their ability to perform various parking maneuvers while respecting all constraints and maintaining collision-free trajectories.

The key contributions of this chapter are:

- Generalization of partial differential flatness to n-GT systems.
- Derivation of a partially flat model for the two-GT system.
- Extension of Lyapunov Control (LC) and trajectory optimization methods to the two-GT configuration.
- Demonstration of the effectiveness of these methods through simulation results, highlighting the system's capability to manage complex maneuvers and ensure collision-free trajectories.

8. Conclusion and Outlook

This work explored optimal trajectory planning and control strategies for truck-trailer systems, focusing on One-General Trailer (one-GT) systems and extending these methods to n-General Trailer (n-GT) configurations. The research began by examining the concept of differential flatness, a widely used approach for developing trajectory control and planning methods. It was identified that while n-Standard Trailer (n-ST) systems are differentially flat, n-General Trailer (n-GT) systems generally are not, except the one-GT system, which exhibits differential flatness. However, the resulting model for one-GT is complex and unsuited for model-based trajectory optimization and control. To address these limitations, the concept of partial differential flatness was introduced.

Building on this foundation, a partial differential flat model was derived and it was first utilized for Lyapunov Control (LC). The model was later integrated into model-based planning and control, enhancing the system's capability to handle complex maneuvers and optimize trajectories. Lyapunov Control (LC) and Linear Quadratic Regulator (LQR) strategies were formulated for hitch angle stabilization and validated through simulations under various input conditions, including step, ramp, and sinusoidal responses.

Nonlinear Model Predictive Control (NMPC) poses significant challenges for real-time applications, as it requires solving complex and computationally intensive optimization problems. Additionally, quadratic programming (QP) solvers typically require a linear system model, which limits the direct application of LQ-MPC for controlling nonlinear systems like truck trailers. The NGMPC framework was formulated to

overcome these challenges, enabling precise trajectory tracking while maintaining computational efficiency.

Within the NGMPC framework, two novel prediction models were developed to address the trajectory stabilization problem for one-GT: Virtual Model Predictive Control (VMPC) and Flat Model Predictive Control (FMPC). The Virtual Model assumes a virtual input, precisely the hitch angle, rather than the steering angle, with a control law integrated within the prediction model. In contrast, the Flat Model was derived using partial differential flatness, where the rate of change of the hitch angle serves as the system input. Both models are designed to be stable against the inherent numerical instability of truck-trailer systems, providing robust solutions for trajectory tracking in complex, dynamic environments.

The developed control methods were experimentally validated, and the results were presented. Practical applicability was a key consideration throughout the research. The motion control algorithms developed here were applied within the Hamburg TruckPilot project, where they were tested in experimental setups. These strategies proved robust and effective, highlighting their potential for real-world deployment in autonomous truck-trailer systems.

Traditionally, optimization-based methods for truck-trailer systems were primarily confined to trajectory stabilization due to the numerical instability inherent in these systems. Researchers have often employed symmetry in trajectory planning, where the reverse motion is treated as equivalent to forward motion, effectively planning trajectories from the goal to the start point in forward motion. Model-based approaches typically linearize the system around a reference trajectory generated through sampling-based planning methods to mitigate numerical instability; however, this often results in an inaccurate representation of the system.

In addressing these limitations, this work employed a partially differential flat model for trajectory planning. This model facilitated the generation of complex maneuvers requiring continuous direction changes,

ensuring that the generated paths were feasible and drivable. The trajectory planning algorithm was validated through parallel, angled, and perpendicular parking maneuvers where collision and confined space constraints were meticulously formulated and tested.

The extension of the partially flat model to n-GT systems with an example of two-GT systems marked a significant advancement. This work expanded the trajectory planning framework to address the complexities of two-GT configurations, including increased non linearity and the challenges of hitch angle stabilization. The framework effectively managed dynamic interactions between multiple trailers by integrating partial differential flatness with trajectory planning formulated as NMPC.

The two-trailer system case study highlighted the framework's ability to navigate constrained environments, execute precise maneuvers, and maintain stability across all trailers. The results demonstrated the robustness and practical applicability of the developed methods in real-world logistics and transportation.

This extension lays the groundwork for future research on even more complex two-GT configurations. The developed framework has significant potential to enhance the efficiency, safety, and reliability of n-GT systems in industries such as freight transport and automated logistics.

8.1. Outlook

Automating truck-trailer systems within logistics hubs will require significant infrastructure adaptations, including installing sensors, V2X communications, and efficient layouts to facilitate autonomous navigation. This thesis presented an approach to address static obstacles within the trajectory planning problem. In structured environments, where dynamic planning needs are limited, trucks can wait until the routes are clear if a dynamic obstacle appears. V2X communication can provide crucial information about other participants in occluded areas, enhancing safety and efficiency.

Trajectory stabilization and planning can be further simplified by employing a high-level mission planner in combination with the proposed methods. Initial trajectories could be computed offline and stored as map data, guiding trucks from entry points to designated parking spots. The mission planner would break down the overall journey into a sequence of goal poses, significantly reducing the complexity of local planning and stabilization tasks. For instance, parking maneuvers could be divided into three distinct missions:

1 Navigating from the entry point to a location near the parking spot.

2 Moving from the initial parking pose to a forward position in preparation for reversing.

3 Executing the reverse maneuver into the parking spot.

This approach simplifies the planning problem by considering only one direction at a time, reducing the need for excessive sensors and computational power, and making it an economically viable solution for the price-sensitive logistics industry.

Looking ahead, this research identifies several areas for further exploration. While the NGMPC framework and developed control strategies offer significant improvements, challenges remain. Future work could enhance these strategies through adaptive and learning-based approaches, such as Reinforcement Learning (RL), which has shown promise in dynamically tuning Model Predictive Controllers (MPC) to adapt to changing conditions. RL could further improve the adaptability and robustness of the control strategies developed in this research, ensuring effectiveness across a broader range of scenarios.

Moreover, while the control strategies were validated on prototypes, real-world implementation will require further investigation into the impact of varying system parameters and environmental conditions. Weight distribution, center of gravity, and sensor accuracy can significantly influence system performance. Addressing these variables by integrating

real-time estimation and adaptive control methods will ensure the reliability and safety of autonomous truck-trailer operations.

In conclusion, integrating the developed planning and control strategies with advanced perception systems and communication technologies offers significant benefits. In highly structured environments like logistics hubs, automation can significantly benefit from enhanced V2X communication and sensor fusion, which enable more precise and efficient navigation. Additionally, employing high-level mission planners to simplify local trajectory planning, combined with the optimization methods developed in this work, could reduce computational demands, making autonomous systems more accessible and cost-effective for the logistics industry.

This research has substantially contributed to trajectory planning and control for truck-trailer systems, addressing theoretical and practical challenges. The methods developed here provide a solid foundation for future advancements, potentially impacting autonomous logistics systems' efficiency, safety, and reliability.

Bibliography

[Ali10] Aliyev, TL and Gatzke, EP: "Prioritized constraint handling NMPC using Volterra series models". In: *Optimal Control Applications and Methods* 31.5 (2010), pp. 415–432 (cit. on p. 75).

[Alt01] Altafini, Claudio: "Some properties of the general n-trailer". In: vol. 74. 4. Taylor & Francis, 2001, pp. 409–424 (cit. on p. 40).

[Alt02] Altafini, Claudio: "Following a path of varying curvature as an output regulation problem". In: *IEEE Transactions on Automatic Control* 47.9 (2002), pp. 1551–1556 (cit. on p. 40).

[Alt98] Altafini, Claudio: "The general n-trailer problem: conversion into chained form". In: *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No. 98CH36171)*. Vol. 3. IEEE. 1998, pp. 3129–3130 (cit. on p. 40).

[And19] Andersson, Joel A E; Gillis, Joris; Horn, Greg; Rawlings, James B and Diehl, Moritz: "CasADi – A software framework for nonlinear optimization and optimal control". In: *Mathematical Programming Computation* 11.1 (2019), pp. 1–36. DOI: 10.1007/s12532-018-0139-4 (cit. on pp. 87, 133, 177).

[Ast01] Astolfi, A.; Bolzern, P. and Locatelli, A.: "Path-tracking of a tractor-trailer vehicle via Lyapunov techniques". In: *2001 European Control Conference (ECC)*. 2001, pp. 451–456. DOI: 10.23919/ECC.2001.7075948 (cit. on p. 29).

[Ast04] Astolfi, Alessandro; Bolzern, Paolo and Locatelli, A.: "Path-Tracking of a Tractor-Trailer Vehicle Along Rectilinear and Circular Paths: A Lyapunov-Based Approach". In: *IEEE Transactions on Robotics and Automation* 20 (1 2004), pp. 154–160. doi: 10.1109/tra.2003.820928 (cit. on p. 61).

[Aug09] Augusto, Jorge; Dalte, Marcos Rosendo; Lages, Walter Fetter and Alves, Vasconcelos: "A Library Tailored for Real-Time Implementation of Model Predictive Control". In: *IFAC Proceedings Volumes* 42.16 (2009), pp. 329–334 (cit. on p. 74).

[Bel11] Bellamy, Drew and Pravica, Luka: "Assessing the impact of driverless haul trucks in Australian surface mining". In: vol. 36. 2. Elsevier. 2011, pp. 149–158 (cit. on p. 2).

[Bel57] Bellman, Richard: Dynamic Programming. 1st ed. USA, 1957 (cit. on p. 28).

[Ber19] Bergman, Kristoffer; Ljungqvist, Oskar; Glad, Torkel and Axehill, Daniel: "An Optimization-Based Receding Horizon Trajectory Planning Algorithm". In: (2019). doi: 10.48550/arxiv.1912.05259 (cit. on p. 125).

[Ber20] Bergman, Kristoffer; Ljungqvist, Oskar; Glad, Torkel and Axehill, Daniel: "An optimization-based receding horizon trajectory planning algorithm". In: *IFAC-PapersOnLine* 53.2 (2020), pp. 15550–15557 (cit. on p. 32).

[Ber95] Bertsekas, Dimitri P: Dynamic programming and optimal control. Athena Scientific, 1995 (cit. on p. 20).

[Bou22] Bourelius, Edvin: Real-time Model Predictive Control with Complexity Guarantees Applied on a Truck and Trailer System. 2022 (cit. on pp. 35, 37).

[Bri19] Brito, Bruno; Floor, Boaz; Ferranti, Laura and Alonso-Mora, Javier: "Model predictive contouring control for collision avoidance in unstructured dynamic

environments". In: *IEEE Robotics and Automation Letters* 4.4 (2019), pp. 4459–4466 (cit. on p. 126).

[Bro83] Brockett, Roger W et al.: "Asymptotic stability and feedback stabilization". In: *Differential geometric control theory* 27.1 (1983), pp. 181–191 (cit. on p. 43).

[Bry18] Bryson, Arthur Earl: Applied optimal control: optimization, estimation and control. Routledge, 2018 (cit. on p. 14).

[Bul96] Bullo, Francesco and Murray, Richard M.: "Experimental Comparison of Trajectory Trackers for a Car With Trailers". In: *Ifac Proceedings Volumes* (1996). doi: 10.1016/s1474-6670(17)58101-x (cit. on p. 44).

[Bus95] Bushnell, L.G.: "An obstacle avoidance algorithm for a car pulling trailers with off-axle hitching". In: *Proceedings of 1995 34th IEEE Conference on Decision and Control*. Vol. 4. 1995, 3837–3842 vol.4. doi: 10.1109/CDC.1995.479196 (cit. on p. 126).

[DeS98] DeSantis, RM: "Path-tracking for articulated vehicles via exact and Jacobian linearization". In: *IFAC Proceedings Volumes* 31.3 (1998), pp. 159–164 (cit. on p. 29).

[Día19] Díaz, Jean and Ayala, José Molina: "Census of Bounded Curvature Paths". In: *Geometriae Dedicata* (2019). doi: 10.1007/s10711-019-00444-2 (cit. on p. 144).

[Eve16] Evestedt, Niclas; Ljungqvist, Oskar and Axehill, Daniel: "Path tracking and stabilization for a reversing general 2-trailer configuration using a cascaded control approach". In: IEEE. 2016, pp. 1156–1161 (cit. on p. 30).

[Fli95] Fliess, Michel; Lévine, Jean; Martin, Philippe and Rouchon, Pierre: "Flatness and defect of non-linear systems: introductory theory and examples". In: *International journal of control* 61.6 (1995), pp. 1327–1361 (cit. on pp. 12–14, 16, 44, 56).

[Föl88] Föllinger, Otto and Roppenecker, Günter: Optimierung dynamischer Systeme: eine Einfuehrung fuer Ingenieure; mit 7 Tabellen und 16 Uebungsaufgaben mit genauer Darstellung des Loesungsweges. Oldenbourg, 1988 (cit. on pp. 21, 22).

[Gal22] Gallazzi, Barbara; Cudrano, Paolo; Frosi, Matteo; Mentasti, Simone and Matteucci, Matteo: "Clothoidal Mapping of Road Line Markings for Autonomous Driving High-Definition Maps". In: *2022 IEEE Intelligent Vehicles Symposium (IV)*. 2022, pp. 1631–1638. DOI: 10.1109/IV51971.2022.9827028 (cit. on p. 151).

[Gan18] Gan, Mi; Li, Dandan; Wang, Mingfei; Zhang, Guangyuan; Yang, Shuai and Liu, Jiyang: "Optimal Urban Logistics Facility Location With Consideration of Truck-Related Greenhouse Gas Emissions: A Case Study of Shenzhen City". In: *Mathematical Problems in Engineering* (2018). DOI: 10.1155/2018/8439582 (cit. on p. 3).

[Gan20] Gan, Mi; Li, Xinyuan; Fadong, Zhang and He, Zhenggang: "Regional Logistics Network Design in Mitigating Truck Flow-Caused Congestion Problems". In: *Journal of Advanced Transportation* (2020). DOI: 10.1155/2020/5197025 (cit. on pp. 3, 4).

[Ger11] Gerdts, Matthias; Henrion, René; Hömberg, Dietmar and Landry, Chantal: "Path planning and collision avoidance for robots". In: (2011) (cit. on p. 126).

[Gim22] Gim, Suhyeon; Lee, Suk-Han and Adouane, Lounis: "Safe and Efficient Lane Change Maneuver for Obstacle Avoidance Inspired From Human Driving Pattern". In: *Ieee Transactions on Intelligent Transportation Systems* (2022). DOI: 10.1109/tits.2020.3034099 (cit. on p. 144).

[Gon09] González-Cantos, A. and Ollero, A.: "Backing-up maneuvers of autonomous tractor-trailer vehicles using the qualitative theory of nonlinear dynamical systems". In: *The*

International Journal of Robotics Research 28 (1 2009), pp. 49–65. DOI: 10.1177/0278364908097588 (cit. on p. 60).

[Gua21] Guan, Chenyong and Jiang, Yu: "A Tractor-Trailer Parking Control Scheme Using Adaptive Dynamic Programming". In: *Complex & Intelligent Systems* (2021). DOI: 10.1007/s40747-021-00330-z (cit. on p. 125).

[Gue19] Guevara, Leonardo; Torres-Torriti, Miguel and Cheein, Fernando Auat: "Collision-free navigation of N-trailer vehicles with motion constraints". In: *2019 12th International Workshop on Robot Motion and Control (RoMoCo)*. 2019, pp. 118–123. DOI: 10.1109/RoMoCo.2019.8787374 (cit. on p. 126).

[Gut17] Gutjahr, Benjamin; Gröll, Lutz and Werling, Moritz: "Lateral vehicle trajectory optimization using constrained linear time-varying MPC". In: vol. 18. 6. IEEE. 2017, pp. 1586–1595 (cit. on p. 126).

[Haa21] Haas, Andreas: "Lateral Trajectory Optimziation and Control of a Truck & Trailer System". Masters Thesis. Karlsruhe Institute of Technology, 2021 (cit. on p. 102).

[Hao03] Hao, Yaoyao; Laxton, B.; Benson, E. R. and Agrawal, Sunil K.: "Robotic Simulation of the Docking and Path Following of an Autonomous Small Grain Harvesting System". In: (2003). DOI: 10.13031/2013.14059 (cit. on p. 44).

[Hao05] Hao, Yaoyao and Agrawal, Sunil K.: "Formation Planning and Control of UGVs With Trailers". In: *Autonomous Robots* (2005). DOI: 10.1007/s10514-005-4750-7 (cit. on pp. 15, 31, 44).

[Ink20] Inkinen, Tommi and Hämäläinen, Esa: "Reviewing Truck Logistics: Solutions for Achieving Low Emission Road Freight Transport". In: *Sustainability* (2020). DOI: 10.3390/su12176714 (cit. on pp. 1, 3).

[Jac70] Jacobson, D.H. and Mayne, D.Q.: Differential dynamic programming. Modern analytic and computational methods in science and mathematics. 1970 (cit. on p. 28).

[Jaz17] Jazar, Reza N: Vehicle dynamics: theory and application. springer, 2017 (cit. on pp. 59, 60).

[Jea96] Jean, Frédéric: "The car with n trailers: characterization of the singular configurations". In: *ESAIM: Control, Optimisation and Calculus of Variations* 1 (1996), pp. 241–266 (cit. on p. 38).

[Kha02] Khalil, Hassan K: Nonlinear systems. Prentice hall, 2002 (cit. on pp. 19, 22, 29).

[Kum20] Kumar, Mohit; Hildebrandt, Arne-Christoph; Strauss, Peter; Kraus, Sven; Stiller, Christoph and Zimmermann, Andreas: "Lateral Trajectory Stabilization of an Articulated Truck during Reverse Driving Maneuvers". In: *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2020, pp. 744–751 (cit. on p. 62).

[Kum22a] Kumar, Mohit; Haas, Andreas; Strauss, Peter; Kraus, Sven; Taş, Ömer Şahin and Stiller, Christoph: "Conception and experimental validation of a model predictive control (MPC) for lateral control of a truck-trailer". In: *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2022, pp. 1550–1557 (cit. on pp. 62, 71, 87).

[Kum22b] Kumar, Mohit; Strauss, Peter; Kraus, Sven; Taş, Ömer Şahin and Stiller, Christoph: "Sharpness Continuous Path optimization and Sparsification for Automated Vehicles". In: *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2022, pp. 1473–1479 (cit. on pp. 145, 149–151).

[Kva19] Kvarnfors, Karl: Motion Planning for Parking a Truck and Trailer System. 2019 (cit. on p. 46).

[Lam03] Lamiraux, Florent; Bonnafous, David and Van Geem, Carl: "Path Optimization for Nonholonomic Systems: Application to Reactive Obstacle Avoidance and Path Planning". In: *Control Problems in Robotics*. Ed. by Bicchi, Antonio; Prattichizzo, Domenico and Christensen, Henrik Iskov. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 1–18. doi: 10.1007/3-540-36224-X_1. url: https://doi.org/10.1007/3-540-36224-X_1 (cit. on pp. 31, 60, 126).

[Lam21] Lambert, Edward Derek; Romano, Richard and Watling, David: "Optimal Smooth Paths Based on Clothoids for Car-Like Vehicles in the Presence of Obstacles". In: *International Journal of Control Automation and Systems* (2021). doi: 10.1007/s12555-020-0179-1 (cit. on p. 144).

[Lau93] Laumond, J-P: "Controllability of a multibody mobile robot". In: *IEEE Transactions on Robotics and Automation* 9.6 (1993), pp. 755–763 (cit. on p. 38).

[Lef05] Lefebvre, Olivier; Lamiraux, Florent and Bonnafous, David: "Fast computation of robot-obstacle interactions in nonholonomic trajectory deformation". In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE. 2005, pp. 4612–4617 (cit. on p. 125).

[Li20] Li, Bai; Acarman, Tankut; Zhang, Youmin; Zhang, Liangliang; Yaman, Cagdas and Kong, Qi: "Tractor-Trailer Vehicle Trajectory Planning in Narrow Environments With a Progressively Constrained Optimal Control Approach". In: *Ieee Transactions on Intelligent Vehicles* (2020). doi: 10.1109/tiv.2019.2960943 (cit. on p. 125).

[Lim15] Lima, Pedro F; Trincavelli, Marco; Mårtensson, Jonas and Wahlberg, Bo: "Clothoid-based model predictive control for autonomous driving". In: IEEE. 2015, pp. 2983–2990 (cit. on p. 145).

[Lim16] Lima, Pedro: "Predictive control for autonomous driving: With experimental evaluation on a heavy-duty construction truck". PhD thesis. KTH Royal Institute of Technology, 2016 (cit. on p. 145).

[Lim17] Lima, Pedro F; Nilsson, Mattias; Trincavelli, Marco; Mårtensson, Jonas and Wahlberg, Bo: "Spatial model predictive control for smooth and accurate steering of an autonomous truck". In: *IEEE Transactions on Intelligent Vehicles* 2.4 (2017), pp. 238–250 (cit. on pp. 145, 151).

[Liz01] Lizárraga, David A; Morin, Pascal and Samson, Claude: "Chained form approximation of a driftless system. Application to the exponential stabilization of the general N-trailer system". In: *International Journal of Control* 74.16 (2001), pp. 1612–1629 (cit. on p. 41).

[Lju15] Ljungqvist, Oskar: Motion planning and stabilization for a reversing truck and trailer system. Department of Electrical Engineering Linköpings universitet, 2015 (cit. on p. 30).

[Lju16] Ljungqvist, Oskar; Axehill, Daniel and Helmersson, Anders: "Path following control for a reversing general 2-trailer system". In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE. 2016, pp. 2455–2461 (cit. on pp. 30, 34, 69).

[Lju19] Ljungqvist, Oskar; Evestedt, Niclas; Axehill, Daniel; Cirillo, Marcello and Pettersson, Henrik: "A Path Planning and Path-following Control Framework for a General 2 trailer With a Car-like Tractor". In: *Journal of Field Robotics* 36.8 (2019), pp. 1345–1377. doi: 10.1002/rob.21908 (cit. on pp. 31, 32, 36, 126).

[Lju20] Ljungqvist, Oskar and Axehill, Daniel: "A predictive path-following controller for multi-steered articulated vehicles". In: *IFAC-PapersOnLine* 53.2 (2020), pp. 15725–15732 (cit. on pp. 31, 34, 36, 69).

[Lug14] Lugo-Cardenas, Israel; Flores, Gerardo; Salazar, Sergio and Lozano, Rogelio: "Dubins Path Generation for a Fixed Wing UAV". In: (2014). doi: 10.1109/icuas.2014.6842272 (cit. on p. 144).

[Lya92] Lyapunov, Aleksandr Mikhailovich: "The general problem of the stability of motion". In: *International journal of control* 55.3 (1992), pp. 531–534 (cit. on pp. 18, 19).

[Mar01] Martin, Philippe; Murray, Richard M and Rouchon, Pierre: "Flat systems, equivalence and feedback". In: *Advances in the control of nonlinear systems*. Springer. 2001, pp. 5–32 (cit. on p. 12).

[Mat12] Mattingley, Jacob and Boyd, Stephen: "CVXGEN: A code generator for embedded convex optimization". In: *Optimization and Engineering* 13.1 (2012), pp. 1–27 (cit. on p. 82).

[May00] Mayne, David Q; Rawlings, James B; Rao, Christopher V and Scokaert, Pierre OM: "Constrained model predictive control: Stability and optimality". In: vol. 36. 6. Elsevier. 2000, pp. 789–814 (cit. on p. 23).

[Mic17] Michałek, Maciej Marcin: "Cascade-Like Modular Tracking Controller for Non-Standard N-Trailers". In: *Ieee Transactions on Control Systems Technology* (2017). doi: 10.1109/tcst.2016.2557232 (cit. on p. 45).

[Moc23] Mochklas, Mohamad; ., Noerchoidah; Mauliddah, Nurullaili and M. Sari, Tyasha A.: "The Existence of Truck Driver in the Spirit of Work". In: *Jurnal Manajemen* (2023). doi: 10.24912/jm.v27i1.1038 (cit. on p. 3).

[Moh17] Mohamed, Amr; Ren, Jing; Lang, Haoxiang and El-Gindy, Moustafa: "Optimal collision free path planning for an autonomous articulated vehicle with two trailers". In: *2017 IEEE International Conference on Industrial Technology (ICIT)*. 2017, pp. 860–865. doi: 10.1109/ICIT.2017.7915472 (cit. on p. 126).

[Moh22] Mohan, Aniruddh and Vaishnav, Parth: "Impact of Automation on Long Haul Trucking Operator-Hours in the United States". In: *Humanities and Social Sciences Communications* (2022). doi: 10.1057/s41599-022-01103-w (cit. on p. 3).

[Mor88] Morari, Manfred; Garcia, Carlos E and Prett, David M: "Model predictive control: theory and practice". In: *IFAC Proceedings Volumes* 21.4 (1988), pp. 1–12 (cit. on p. 74).

[Mur17] Murray, Richard M; Li, Zexiang and Sastry, S Shankar: A mathematical introduction to robotic manipulation. CRC press, 2017 (cit. on p. 14).

[Nat23] Nath, Asoke and Jisha, V R: "Trajectory Tracking Control of a Truck Trailer Robot Using Feedback Linearization". In: (2023). doi: 10.1109/iccc57789.2023.10165610 (cit. on pp. 31, 125).

[Nie98] Nieuwstadt, Michiel van; Rathinam, Muruhan and Murray, Richard M: "Differential flatness and absolute equivalence of nonlinear control systems". In: *SIAM Journal on Control and Optimization* 36.4 (1998), pp. 1225–1239 (cit. on p. 14).

[Nua] Nuaimi, Mohanad Al: "Analysis and Comparison of Clothoid and Dubins Algorithms for UAV Trajectory Generation". In: (). doi: 10.33915/etd.7059 (cit. on p. 144).

[Oli18] Oliveira, Rui; Lima, Pedro F.; Cirillo, Marcello; Mårtensson, Jonas and Wahlberg, Bo: "Trajectory Generation using Sharpness Continuous Dubins-like Paths with Applications in Control of Heavy-Duty Vehicles". In: *2018 European Control Conference (ECC)*. 2018, pp. 935–940. doi: 10.23919/ECC.2018.8550279 (cit. on p. 145).

[Pad16] Paden, Brian; Čáp, Michal; Yong, Sze Zheng; Yershov, Dmitry and Frazzoli, Emilio: "A survey of motion planning and control techniques for self-driving urban vehicles".

In: *IEEE Transactions on intelligent vehicles* 1.1 (2016), pp. 33–55 (cit. on p. 15).

[Par16] Park, Hyeongjun; Zagaris, Costantinos; Virgili Llop, Josep; Zappulla, Richard; Kolmanovsky, Ilya and Romano, Marcello: "Analysis and experimentation of model predictive control for spacecraft rendezvous and proximity operations with multiple obstacle avoidance". In: *AIAA/AAS astrodynamics specialist conference*. 2016, p. 5273 (cit. on p. 75).

[Pau22] Pauls, Jan-Hendrik; Boxheimer, Mario and Stiller, Christoph: "Real-time cooperative motion planning using efficient model predictive contouring control". In: *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2022, pp. 1495–1503 (cit. on p. 126).

[Ple17] Plessen, Mogens Graf; Bernardini, Daniele; Esen, Hasan and Bemporad, Alberto: "Spatial-based predictive control and geometric corridor planning for adaptive cruise control coupled with obstacle avoidance". In: *IEEE Transactions on Control Systems Technology* 26.1 (2017), pp. 38–50 (cit. on p. 126).

[Pon62] Pontryagin, Lev Semenovich; Boltyanskii, Vladimir Grigorevich; Gamkrelidze, Revaz Valerianovich and Mishchenko, Evgenii Frolovich: The Mathematical Theory of Optimal Processes. 1962 (cit. on p. 21).

[Pra07] Pradalier, Cédric and Usher, Kane: "A Simple and Efficient Control Scheme to Reverse a Tractor-Trailer System on a Trajectory". In: (2007). DOI: 10.1109/robot.2007.363648 (cit. on pp. 61, 62).

[Pra08] Pradalier, Cédric and Usher, Kane: "Robust Trajectory Tracking for a Reversing Tractor Trailer". In: *Journal of Field Robotics* (2008). DOI: 10.1002/rob.20241 (cit. on p. 61).

[Qur19] Qureshi, Ahmed H.; Simeonov, Anthony; Bency, Mayur J. and Yip, Michael C.: "Motion Planning Networks". In: *2019 International Conference on Robotics and Automation (ICRA)*. 2019, pp. 2118–2124. DOI: 10.1109/ICRA.2019.8793889 (cit. on p. 126).

[Ram14] Ramasamy, Suresh; Wu, Guofan and Sreenath, Koushil: "Dynamically Feasible Motion Planning through Partial Differential Flatness." In: *Robotics: Science and Systems*. 2014 (cit. on pp. 15–17).

[Raw09] Rawlings, James B and Mayne, David Q: Model predictive control: Theory and design. 2009 (cit. on p. 23).

[Rig20] Rigatos, Gerasimos; Siano, Pierluigi; Wira, Patrice; Busawon, Krishna and Binns, Richard: "A Nonlinear H-Infinity Control Approach for Autonomous Truck and Trailer Systems". In: *Unmanned Systems* (2020). DOI: 10.1142/s2301385020500041 (cit. on p. 61).

[Rou93] Rouchon, Pierre; Fliess, Michel; Lévine, Jean and Martin, Philippe: "Flatness, motion planning and trailer systems". In: *Proceedings of 32nd IEEE Conference on Decision and Control*. IEEE. 1993, pp. 2700–2705 (cit. on pp. 12, 45–48, 56).

[Ryu08] Ryu, Ji-Chul; Agrawal, Sunil K. and Franch, Jaume: "Motion Planning and Control of a Tractor With a Steerable Trailer Using Differential Flatness". In: *Journal of Computational and Nonlinear Dynamics* (2008). DOI: 10.1115/1.2908178 (cit. on p. 45).

[Sal23] Salamah, Yasser Bin: "Sliding Mode Controller for Autonomous Tractor-Trailer Vehicle Reverse Path Tracking". In: *Applied Sciences* (2023). DOI: 10.3390/app132111998 (cit. on p. 31).

[Sha23] Shamuyarira, Sheunesu Brandon: "Truck Fuel Consumption Prediction Using Logistic Regression and Artificial Neural Networks". In: *International Journal of Operations Research and Information Systems* (2023). DOI: 10.4018/ijoris.329240 (cit. on p. 3).

[Sil01] Silva, Paulo Sérgio Pereira da and Filho, Carlos Corrêa: "Relative flatness and flatness of implicit systems". In: *SIAM Journal on Control and Optimization* 39.6 (2001), pp. 1929–1951 (cit. on p. 15).

[Sim13] Simon, Daniel; Löfberg, Johan and Glad, Torkel: "Nonlinear model predictive control using Feedback Linearization and local inner convex constraint approximations". In: *2013 European Control Conference (ECC)*. 2013, pp. 2056–2061. DOI: 10.23919/ECC.2013.6669575 (cit. on pp. 74, 75).

[Slo91] Slotine, Jean-Jacques E. and Li, Weiping: Applied Nonlinear Control. Prentice-Hall, Inc., 1991 (cit. on p. 18).

[Smi17] Smith, Nathan E; Arendt, Christopher D; Cobb, Richard G and Reeger, Jonah A: "Implementing conditional inequality constraints for optimal collision avoidance". In: *Journal of Aeronautics and Aerospace Engineering* (2017) (cit. on p. 126).

[Sor93] Sordalen, O.J.: "Conversion of the kinematics of a car with n trailers into a chained form". In: *[1993] Proceedings IEEE International Conference on Robotics and Automation*. 1993, 382–387 vol.1. DOI: 10.1109/ROBOT.1993.292011 (cit. on p. 41).

[Spo20] Spong, Mark W; Hutchinson, Seth and Vidyasagar, Mathukumalli: Robot modeling and control. John Wiley & Sons, 2020 (cit. on p. 14).

[Ste94] Stengel, Robert F: Optimal control and estimation. Courier Corporation, 1994 (cit. on p. 20).

[Til95] Tilbury, Dawn M.; Sørdalen, O.J.; Bushnell, Linda and Sastry, S. Shankar: "A Multisteering Trailer System: Conversion Into Chained Form Using Dynamic Feedback". In: *Ieee Transactions on Robotics and Automation* (1995). doi: 10.1109/70.478428 (cit. on p. 44).

[Wer14] Werling, Moritz; Reinisch, Philipp; Heidingsfeld, Michael and Gresser, Klaus: "Reversing the general one-trailer system: Asymptotic curvature stabilization and path tracking". In: *IEEE Transactions on intelligent transportation systems* 15.2 (2014), pp. 627–636 (cit. on pp. 29, 43, 44).

[Wu17] Wu, Tong and Hung, John Y.: "Path following for a tractor-trailer system using model predictive control". In: *SoutheastCon 2017*. 2017, pp. 1–5. doi: 10.1109/SECON.2017.7925337 (cit. on pp. 34, 36, 69).

[Zah23] Zahirah Ismadi, Nurul Syamimi: "Drag Reduction of a Simplified Truck Model Using Cab Roof Fairings". In: *Asean Engineering Journal* (2023). doi: 10.11113/aej.v13.19277 (cit. on p. 3).

[Zan18] Zanchetta, M.; Tavernini, Davide; Sorniotti, Aldo; Gruber, Patrick; Lenzo, Basilio; Ferrara, Antonella; Nijs, Wouter De; Sannen, Koen and Smet, Jasper De: "On the Feedback Control of Hitch Angle Through Torque-Vectoring". In: *2018 IEEE 15th International Workshop on Advanced Motion Control (AMC)* (2018). doi: 10.1109/amc.2019.8371150 (cit. on pp. 60, 61).

[Zei11] Zeilinger, Melanie Nicole: "Real-time model predictive control". PhD thesis. ETH Zurich, 2011 (cit. on p. 74).

[Zha24a] Zhang, Songyi; Wang, Runsheng; Jian, Zhiqiang; Zhan, Wei; Zheng, Nanning and Tomizuka, Masayoshi: "Clothoid-Based Reference Path Reconstruction for HD Map Generation". In: *IEEE Transactions on Intelligent Transportation*

Systems 25.1 (2024), pp. 587–601. DOI: 10.1109/TITS.2023.3305198 (cit. on p. 151).

[Zha24b] ZHAO, Tianrui: "A Simple Curvature-Based Backward Path-Tracking Control for a Mobile Robot With N Trailers". In: *Actuators* (2024). DOI: 10.3390/act13070237 (cit. on p. 31).

[Zip15] ZIPS, Patrik; BÖCK, Martin and KUGI, Andreas: "An Optimisation-Based Path Planner for Truck-Trailer Systems with Driving Direction Changes". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. Seattle, WA, USA, May 2015, pp. 630–636. DOI: 10.1109/ICRA.2015.7139245 (cit. on p. 125).

Own publications

[1] Kumar, Mohit; Hildebrandt, Arne-Christoph; Strauss, Peter; Kraus, Sven; Stiller, Christoph and Zimmermann, Andreas: "An Optimal Lateral Trajectory Stabilization of Vehicle using Differential Dynamic Programming". In: *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2020, pp. 623–630.

[2] Kumar, Mohit; Hildebrandt, Arne-Christoph; Strauss, Peter; Kraus, Sven; Stiller, Christoph and Zimmermann, Andreas: "Lateral Trajectory Stabilization of an Articulated Truck during Reverse Driving Maneuvers". In: *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2020, pp. 744–751.

[3] Kumar, Mohit; Haas, Andreas; Strauss, Peter; Kraus, Sven; Taş, Ömer Şahin and Stiller, Christoph: "Conception and experimental validation of a model predictive control (MPC) for lateral control of a truck-trailer". In: *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2022, pp. 1550–1557.

[4] Kumar, Mohit; Strauss, Peter; Kraus, Sven; Taş, Ömer Şahin and Stiller, Christoph: "Sharpness Continuous Path optimization and Sparsification for Automated Vehicles". In: *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2022, pp. 1473–1479.

Supervised student theses

[1] Mittal, Ayush: “Lateral Trajectory Stabilization of a Truck-Semitrailer during High Dynamic Maneuvers”. Masters Thesis. RWTH Aachen University, 2020.

[2] Haas, Andreas: “Lateral Trajectory Optimziation and Control of a Truck & Trailer System”. Masters Thesis. Karlsruhe Institute of Technology, 2021.

[3] Kiani, Keymand: “Predictive lateral trajectory optimization and control of a truck for automated logistic hubs”. Masters Thesis. RWTH Aachen University, 2021.

List of Figures

List of Tables

A. Appendix

A.1. Cascade Control PPC-LC

This section elaborates the developed cascade control for trajectory tracking control of a truck-trailer. The cascade control uses bi-level stabilization: a high-level control computes the desired hitch angle and a low-level control stabilizes the system at the desired hitch angle.

This section presents the high-level control for the trailer. The trailer is assumed as a *Virtual Vehicle* with hitch angle θ as virtual steering input, depicted in fig. A.1.

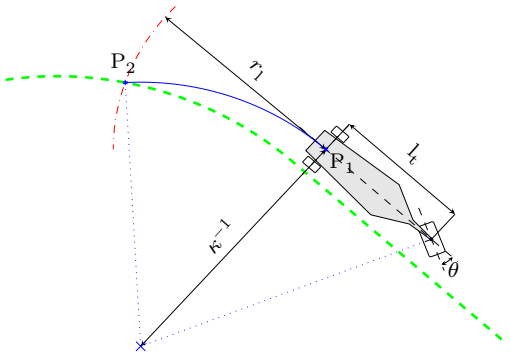


Figure A.1.: Geometric Illustration of Pure Pursuit Control for Desired Hitch Angle Calculation in Trailer Tracking

Due to its straightforward application, pure pursuit control is adopted for high-level reference tracking. The desired steering angle is computed by assuming that the vehicle moves along an arc of constant curvature. This arc extends from the current position to the look-ahead point, defined as the intersection of the reference path with the look-ahead circle. The desired curvature κ can be derived from the geometry shown in Fig. A.1 as follows:

$$\kappa = \frac{2y}{r_1^2}. \tag{A.1}$$

where r_1 is the radius of the look-ahead circle and y is the coordinate of of point P_2 in vehicle coordinate frame originated P_1.
Assuming the on axle hitching as depicted in fig. A.1 and using the standard 1-trailer system as discussed in section 3.3.1, the desired hitch angle can be computed as:

$$\theta = \arctan \mathrm{l_t} \kappa. \tag{A.2}$$

The hitch angle is stabilized using the Simiplified Lyapunov Control (SLC) explained in chapter 4.