Tube MPC for Robust Lateral Control of a Class 8 Tractor-Trailer with Parameter Uncertainty

by

Evan Ellison

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
August 8, 2026

Keywords: Autonomy, ADAS, Model Predictive Control

Approved by

David Bevly, Bill and Lana McNair Professor
Scott Martin, Professor of Mechanical Engineering
Ehsan Taheri, Professor of Aerospace Engineering

Abstract

This thesis presents a Model Predictive Control (MPC) design for motion planning and control of a five axle tractor-trailer vehicle. The targeted use case for this design is for Society of Automotive Engineers (SAE) Level 3-4 features which may include automated highway driving and lane change or obstacle avoidance maneuvers. Autonomous control of commercial tractor trailer vehicles, specifically class 8 trucks, presents unique challenges due to the need for specific safety guarantees and lack of accurate knowledge of all of the model parameters, such as the mass and yaw inertia of the payload in the trailer. These challenges can be handled in part by MPC due to its ability to enforce constraints and find optimal trajectories with respect to an objective. Additionally, many techniques for ensuring constraints are satisfied under uncertainty exist which can prove useful for this application. In this thesis, a commonly used dynamic model for tractor-trailers is first presented. Next, a full prediction model for use in the MPC is developed, which combines the equations for propagating position with respect to the road and a model for the steering actuator with the lateral dynamic model of the vehicle. An MPC design is then introduced by defining the optimal control problem and solving it as a Quadratic Program (QP). The MPC is able to plan and execute a trajectory that ensures constraints related to the vehicle's position and trailer states such as the hitch angle can be met. A higher update rate feedback controller is used to aid the tracking of the latest solution between MPC updates. A constraint tightening technique is also applied which constructs an error tube around the planned trajectory based on the uncertainty of model parameters. An analysis of the total accuracy and performance in different scenarios is presented. The combined online planning and control scheme is validated in simulation, and the MPC performance with and without constraint tightening is compared for several relevant scenarios, and improvements in the number of scenarios that satisfy the lateral position and hitch angle constraints is demonstrated. Finally, the real-world capability of the design is demonstrated on

an autonomy capable Peterbilt 579 with a trailer attached. The experimental testing demonstrates the feasibility of the concept for real-time control through lane keeping tests, resulting in absolute tracking errors with at most a mean of 18.9 cm and a standard deviation of 11.9 cm.

Artificial Intelligence (AI) Use Disclosure Statement

In the preparation of this thesis, the following Artificial Intelligence (AI) tools were used: Anthropic's Claude. These tools were used primarily for error checking and grammatical review. The author acknowledges full responsibility for the intellectual content of this work and has ensured that all AI-assisted sections have been reviewed and revised for accuracy and appropriate academic style. All AI-generated content was reviewed and validated for relevance, appropriateness, and accuracy before incorporation into the final document to maintain scholarly integrity of this research.

Digital Accessibility Disclousre Statement

In the preparation of this thesis, the following digital accessibility tools were used to ensure this document complies with federal requirements: Adobe Acrobat accessibility tools. The author acknowledges full responsibility for the intellectual content of this work and has made a good faith effort to comply with digital accessibility requirements in publishing, wherein the nature of the content does not significantly change in order to do so. Furthermore, all content has been reviewed and revised to meet these requirements prior to final publication.

Acknowledgments

I would first like to thank my family for their support throughout the process of completing my Master's degree. My mother, brothers, and sister have been a source of constant encouragement throughout my educational journey.

I would like to thank my advisor, Dr. Bevly, for giving me the opportunity to work on interesting projects in the GAVLab. I have learned a lot during my time in the lab and it's been a challenging and rewarding experience. I want to thank Dr. Scott Martin for his instruction and guidance in the lab as well. I also owe thanks to Dr. Howard Chen for introducing me to the world of software for autonomy and to the opportunity of graduate school.

I would like to express my gratitude to Dr. Lowell Brown for his feedback on the projects that I have worked on throughout graduate school and during my internship at Daimler Truck. I would also like to thank Cris Paun for his mentorship. I am grateful for all their feedback and advice which has been a source of professional development.

Finally, I want to thank my fellow lab members, especially anyone who I've had the pleasure of working with on data collections, racing events, or demos. I owe a special thanks to Kyle Thompson and Brandon Ravenscraft for their time spent helping with the experimental testing for this thesis.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ACC	Adaptive Cruise Control
ADAS	Advanced Driver Assistance Systems
CAN	Controller Area Network
CBF	Control Barrier Function
CNN	Convolutional Neural Network
LKA	Lane Keep Assist
LTI	Linear Time Invariant
LTV	Linear Time Varying
MPC	Model Predictive Control
NCAT	National Center for Asphalt Technology
NMPC	Nonlinear Model Predictive Control
OCP	Optimal Control Problem
QP	Quadratic Program

RMPC	Robust Model Predictive Control
ROS	Robot Operating System
RTI	Real-Time Iteration
RTK	Real-Time Kinematic
SAE	Society of Automotive Engineers
SGA	Side Guard Assist
SQP	Sequential Quadratic Program

Chapter 1

Introduction

1.1 Background and Motivation

Full and partial autonomous driving features for commercial vehicles are currently an active area of product development and research due to the proposed benefits of these features. The potential benefits of these features include enhanced safety and reduction of driver fatigue, which in turn can increase the amount of time that a trucking fleet is safely operating [1]. The currently available Advanced Driver Assistance Systems (ADAS) features vary in function and in how they are implemented, but are typically classified into six levels of autonomous driving, as defined by the SAE Levels of Automated Driving J3016 standard, and summarized by the graphic shown in Figure 1.1 [2].

SAE Level 1-2 features such as Adaptive Cruise Control (ACC), lane centering, and automatic emergency braking have become common on commercial trucks. In particular, truck manufacturers have included systems that provide limited lateral control of the vehicle. Daimler Truck has equipped some of their vehicles with the Detroit Assurance 5.0 ADAS system. This system includes Lane Keep Assist (LKA), which uses ”micro-steering movements” to assist the driver in maintaining their lane, and a Side Guard Assist (SGA) system which can warn the driver of an impending collision with a vehicle to its side but does not steer to avoid a collision [3]. Volvo Truck’s Pilot Assist system provides lane centering in a similar way to aid the driver [4].

Development of more advanced autonomy features is also an area of active interest. PlusAI has partnered with Bosch to develop what is described as a Level 2++ feature which is intended

to control the vehicle for extended periods of time without driver intervention and can handle scenarios such as traffic jams or merging autonomously [5]. Additionally, companies such as Aurora and Kodiak have begun testing full autonomy systems [6, 7].

Automated control of these vehicles presents a variety of unique challenges compared to passenger cars. These include the need to account for the position of the trailer with respect to the road, objects, and other vehicles. Other challenges include the unknown quantities introduced by the fact that trailers can be exchanged and may have different configurations, cargo can be added or removed, and the tire parameters can vary. These challenges make model-based control techniques more difficult due to the fact that many parameters cannot be assumed to be known a priori. Instead, they must be estimated or bounds on them are assumed. The challenges related to the modeling of the vehicle are exacerbated by the difficulty of directly measuring the hitch angle of the trailer without equipping it with sensors.

The focus of this thesis is on the design of a lateral control system for on-road driving of a commercial tractor-trailer vehicle. In particular, this system is designed to be part of an SAE Level 2-3 ADAS feature which is capable of performing an autonomous avoidance maneuver to avoid objects or vehicles in the road. The method presented could also be used as a subsystem of a feature which is capable of extended periods of autonomous driving on highways. The system presented in this thesis aims to accomplish online planning and control for autonomous lane-keeping and avoidance tasks using a Model Predictive Control (MPC) design combined with a higher update rate feedback controller. This design is able to constrain the states of the vehicle, including the position of the tractor and hitch angle of the trailer, to ensure lateral maneuvers are executed in a way that is safe for avoiding collisions during on-road driving. Additionally, the algorithm aims to ensure these constraints are satisfied even when some of the vehicle parameters may be estimated or are not known exactly.

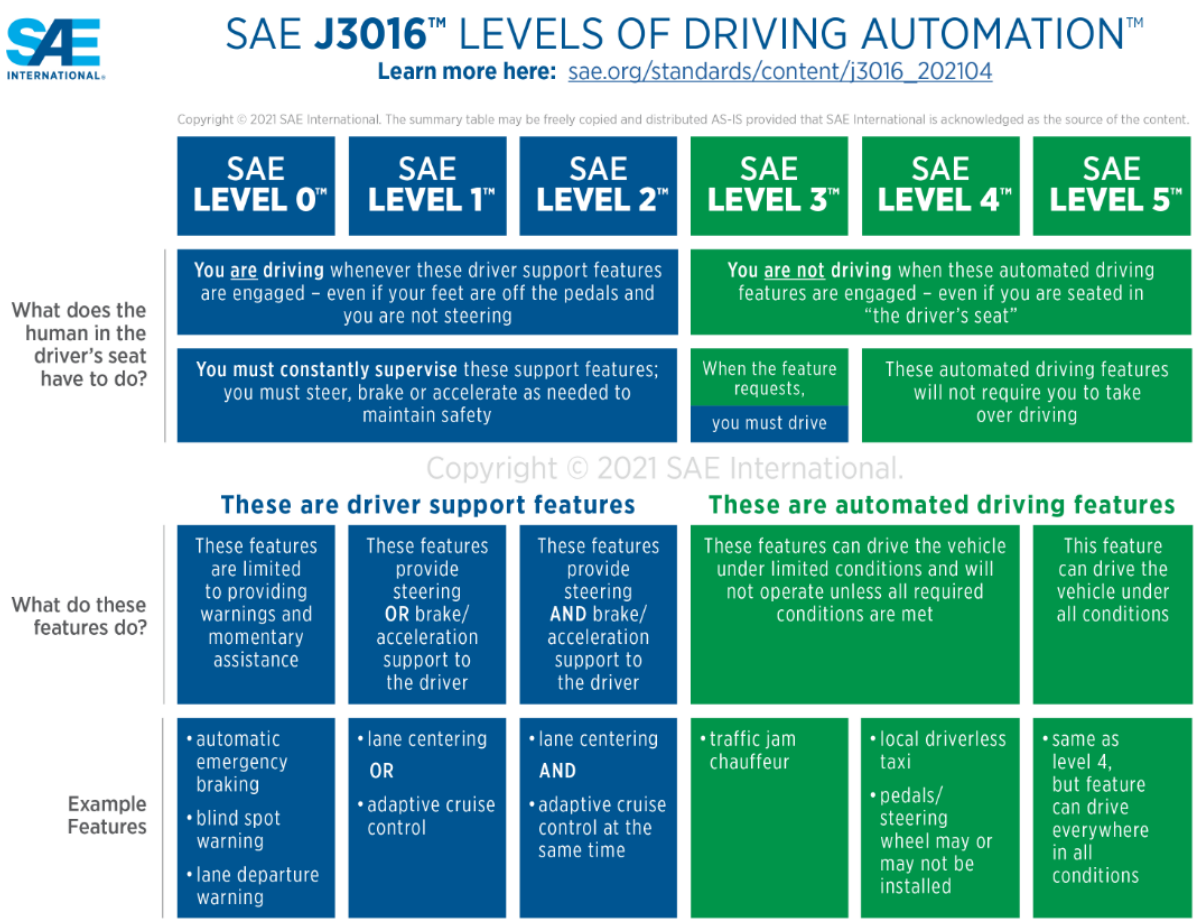

Figure 1.1: SAE J3016 Levels of Driving Autonomy [2]

1.2 Prior Work

While systems for automated control of tractor trailer vehicles exist in various forms in industry, the control techniques used are not always available to the public. However, MPC has recently emerged in the research literature as a popular method for control of ground vehicles. MPC is a form of optimal control that operates on the receding horizon control principle. This means that the control action consists of solving an Optimal Control Problem (OCP) over a finite horizon and then enacting the first predicted optimal control input. At each update, the horizon is shifted forward and the process is repeated [8]. MPC has become a popular method for ground vehicle control in part due to its ability to explicitly place constraints on the states and control variables throughout the prediction horizon.

A variety of MPC implementations using a kinematic model of an articulated vehicle have been proposed in the literature. Earlier work on this topic demonstrated an implementation of MPC using a kinematic model of a tractor trailer for path tracking applications, showing improved lateral error performance compared to LQR [9]. The method proposed in [9] also demonstrated the ability to control the trailer position. In [10], a kinematic model was used in the MPC and a method of adapting the parameters of the controller was implemented to improve performance. The control method was validated in the commercial simulation software TruckSim, and experimental tests were carried out on an autonomous semi-trailer truck at speeds of approximately 18 mph [11]. While an MPC based on a kinematic model of the vehicle can produce reasonable results, as speeds increase and the vehicle is experiencing more slip a dynamic model produces more accurate predictions. An MPC design using a kinematic model would be best suited for low speed driving tasks, and using a dynamic model in the design would be needed for accurate prediction at higher speeds.

Approaches using a dynamic model within an MPC framework have also been proposed. Linear MPC using a dynamic model of a tractor and semi-trailer was demonstrated in [12]. The author also compared MPC using the dynamic model against a design based on a kinematic model. Furthermore, a dynamic model that included roll and non-linear tire models was compared against a model that neglected these effects. The techniques compared in this study are referred to as Nonlinear Model Predictive Control (NMPC) due to the use of nonlinear dynamic models. The authors conducted a comparison in simulation, and found that NMPC performed better than LQR at aggressive driving tasks. They also found that the use of the dynamic model, even without inclusion of roll and nonlinear tire models, showed improvement over the kinematic model. However, the authors cited implementation of Robust Model Predictive Control (RMPC), specifically to handle parameter uncertainty, as a necessary area of exploration for future work on this problem. Furthermore, although the NMPC designs that were presented showed promising real-time capability, the analysis was conducted in simulation only. In [13], a design of an SAE Level 2 lane keeping system was presented which used a dynamic model within an NMPC scheme. While this

method demonstrated the capabilities of NMPC for this application through experimental testing on a truck, it operated as a path tracking controller which followed pre-defined waypoints given by a lead vehicle. The controller minimized squared error to the set of provided waypoints, but was not intended for standalone planning and execution of maneuvers given a set of constraints. Additionally, waypoint following performance at a speed of 25 mph was demonstrated, achieving a lateral RMSE of approximately 27 cm. However, avoidance maneuvers with position constraints on the tractor and trailer were not considered, and the effect of parameter uncertainty was not analyzed.

MPC has shown potential to be a powerful control technique for control of ground vehicles including tractor-trailers. However, the effect of parameter uncertainty is a major concern for any model-based design. A wide range of techniques for managing uncertainty have been proposed that fall under the broad category of RMPC. A basic form of tube-based RMPC, sometimes referred to simply as constraint tightening, assumes a feedback controller is active throughout the horizon and constructs an error bound around the nominal prediction. Tightened constraints are then calculated such that every trajectory that stays within the error bounds will satisfy the original constraints [8, 14]. For a Linear Time Invariant (LTI) system with an additive disturbance, the error bound is constructed offline because the tube will be the exact same size regardless of the trajectory. For cases where the system is not LTI or the uncertainty is not an additive disturbance, constructing an accurate error tube becomes more complicated. Recent work has used the concept of closed loop state sensitivity to model the error around a predicted trajectory caused by error in the parameters. This concept was demonstrated for offline trajectory planning of a quadrotor in [15]. In that work, the sensitivity matrix itself was included in the cost so that a trajectory was chosen that was inherently robust to parameter error. In [16], the concept of closed loop state sensitivity was used to explicitly construct an error tube around the predicted trajectory for use in an RMPC method. The method proposed by the authors was tested in simulation and in real experiments on a quadrotor and showed improvements in constraint satisfaction. The tube propagation method was extended in [17] to account for uncertainty in the initial state estimate and estimates of obstacle locations

as well as the model parameter estimates. One of the most popular methods for handling parameter uncertainty has been Scenario MPC, in which multiple versions of the model are sampled. Constraints are then enforced on every realization of the model. However, this can lead to a large increase in the number of decision variables, increasing computation time [18, 19]. The sensitivity based method, denoted by the authors of [16] as Sensitivity-aware Tube MPC (ST-MPC), aims to account for parameter uncertainty while maintaining computational efficiency.

RMPC has also seen some success in ground vehicle applications. In [20], the authors demonstrated a tube MPC approach for a passenger car. The disturbances were modeled as an additive disturbance and a constant tube radius was used in the design. The authors demonstrated cases of high speed obstacle avoidance in simulation and in experimentation, with the RMPC method showing improved robustness. RMPC methods have also been applied to tractor-trailer vehicles. In [21], the authors applied tube MPC to a small agricultural tractor trailer vehicle and demonstrated trajectory tracking capabilities. An additive disturbance model was used in that work, with tightened constraints computed offline. More recently, a robust MPC method was demonstrated for a combination vehicle with two trailers in simulation by Han et al. [22]. The authors applied a constraint tightening method based on the effect of varying trailer mass on the hitch angle transfer functions. The constraint tightening method ensured that the geometric lane-keeping constraints were satisfied on curved roads, and improved performance was demonstrated compared to controllers which did not account for the uncertainty in trailer masses and other parameters which are affected by trailer masses.

In addition to the control techniques available in the literature, there has been a significant amount of research on state estimation, parameter estimation, and rollover prevention. Contributions in these areas are relevant to this thesis because they inform the design of the control system. Most importantly, feedback of trailer hitch angle and hitch rate is assumed to be available. Despite this assumption, obtaining measurements of the trailer states is not a trivial task because adding sensors to the trailer itself is not viable for commercial trucking. However, recent work has demonstrated the potential for obtaining measurements of trailer angle without the need for adding sensors

to the trailer [23]. This is accomplished through the use of a Convolutional Neural Network (CNN) in combination with rear facing cameras on the cab and was demonstrated in simulation. Estimation of the model parameters is another important component. The authors of [24] demonstrate the ability to estimate the vehicle's mass to within 10 percent. Considering that approximate values of some parameters can be assumed given the mass, and other parameters can be estimated using this quantity, knowledge of the mass provides a valuable starting point for obtaining other model parameters. Finally, rollover is a major concern when automating lateral control of tractor trailer vehicles. State of the art methods for automated rollover prevention using data-driven techniques are described in [25], which could be combined with different control methods. These developments are not alternatives to the work presented in this thesis, but are relevant background material which demonstrate how the control methods presented in this thesis may fit into automated driving features for tractor-trailers.

1.3 Contributions

While NMPC techniques have been demonstrated for control of tractor-trailers, this thesis presents a Linear Time Varying (LTV) MPC design that can be solved as a QP and is experimentally validated on a Class 8 truck with trailer attached. Additionally, a real-time suitable implementation of a robust MPC method to account for parameter uncertainty is presented. The method used to account for parameter uncertainty used in Sensitivity-aware Tube MPC is adapted for this thesis, and provides a computationally efficient way of improving robustness to parameter uncertainty. Specifically, the contributions in this thesis are:

- An LTV MPC design for tractor trailers that uses a dynamic model of the vehicle and can be solved as a quadratic program (QP).
- An implementation of a constraint tightening method based on the sensitivity of the dynamic model to its parameters, and an analysis of the scenarios in which it can improve constraint satisfaction.

- A validation of the proposed methods for obstacle avoidance in simulation, and experimental demonstration of the real-time capability in lane keeping tests on a Class 8 truck with a trailer attached.

1.4 Thesis Outline

There are five remaining chapters in this thesis. Chapter 2 contains the derivation of the lateral dynamic model for a five axle tractor-trailer vehicle, along with a model that uses road curvature to mechanize position of the vehicle with respect to the road. The full prediction model used in the MPC is presented, which also includes a steering actuation model. Chapter 3 includes the model from Chapter 2 into the design of the control strategy. First, the MPC design is introduced. Then, a method of creating error bounds on the MPC prediction based on parameter uncertainty is presented. Finally, the procedure used for real-time implementation is detailed, which incorporates the error bounds on the prediction and includes a higher update rate feedback controller used to maintain tracking of the MPC trajectory between updates. Chapter 4 presents simulation results which validate the approach presented in Chapter 3 with analysis of several relevant scenarios. Chapter 5 presents results from a real-time implementation on a test vehicle to demonstrate the capability of the proposed method. Finally, Chapter 6 includes the conclusions and summary of this work, and provides suggestions for future work.

Chapter 2

Vehicle Modeling

2.1 Lateral Dynamic Model

The lateral dynamic model for tractor-trailers used in this thesis is a single-track five axle model which was derived in [26] and further validated in [27]. Core assumptions associated with the single track nature of the model are that there is a single effective steer angle at the front axle and the cornering stiffnesses of the tires are combined for each axle. The free body diagram is shown in Figure 2.1, where the states are labeled and include tractor yaw ψ, tractor yaw rate $\dot{\psi}$, tractor lateral velocity v_y, hitch rate $\dot{\gamma}$, and hitch angle γ. Masses and yaw inertias of the tractor and trailer are labeled m_1, J_1, and m_2, J_2 respectively. The body frame is fixed to the center of gravity of the tractor and the y axis is positive out of the driver side while angular rates, yaw and hitch angle are counter-clockwise positive.

The model is derived in [26] using the Lagrange method because this provides a straight forward way to account for the relationship between the tractor and trailer. The first step in deriving the equations of motion is to write the Lagrangian, which in this context is defined as the difference between kinetic energy, T, and potential energy, V, and is shown in Equation (2.1).

$$L = T - V \tag{2.1}$$

Equation (2.2) shows the Lagrangian for this model, which is made up of translational and rotational kinetic energies of the tractor and trailer.

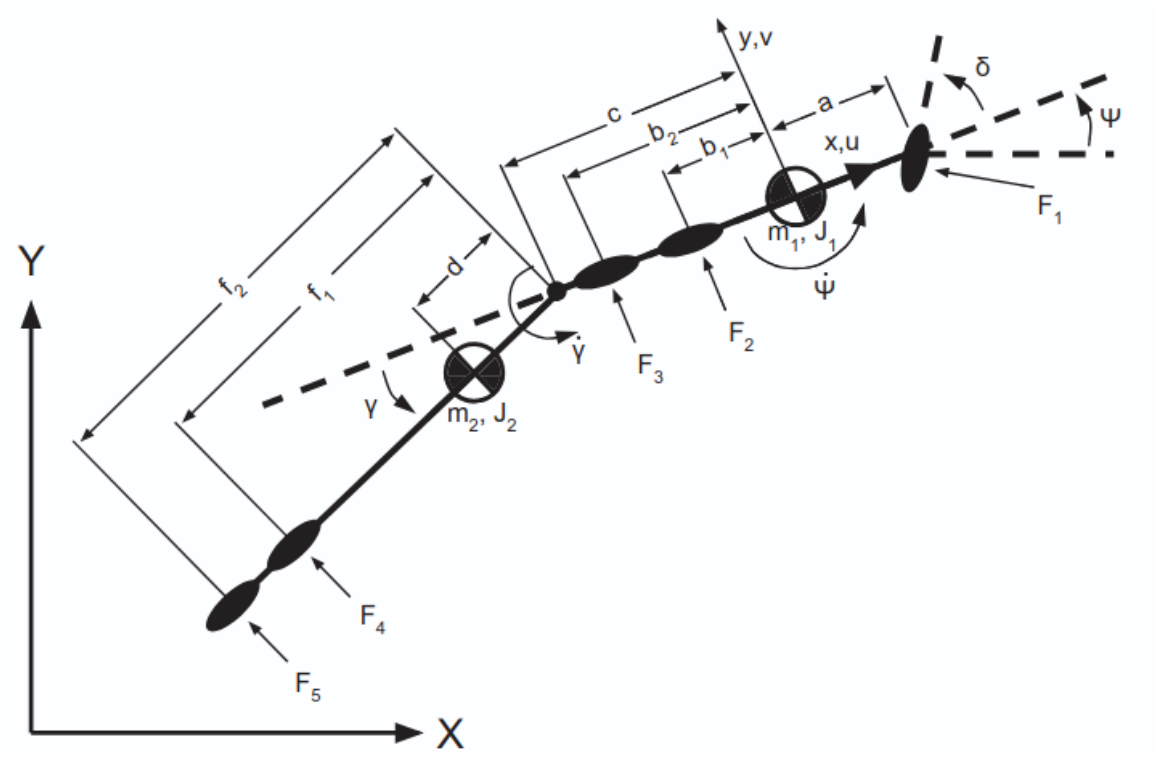

Figure 2.1: 5 Axle Tractor-Trailer Diagram [26]

$$L = \frac{1}{2}m_1(v_x^2 + v_y^2) + \frac{1}{2}J_1\dot{\psi}^2 + \frac{1}{2}m_2(v_{x,trailer}^2 + v_{y,trailer}^2) + \frac{1}{2}J_2\dot{\psi}_{trailer}^2 \tag{2.2}$$

The next step in the derivation is to choose a set of generalized coordinates, $\boldsymbol{q}$, that represent the problem well. For this model these coordinates are the y coordinate in the body frame of the vehicle, yaw, and hitch angle, and are shown in Equation (2.3).

$$\boldsymbol{q} = \begin{bmatrix} y & \psi & \gamma \end{bmatrix}^T \tag{2.3}$$

Relationships between the tractor and trailer velocities can also be defined, which relate the trailer translational and angular velocities to the generalized coordinates. The first relationship is between longitudinal velocities and is given in Equation (2.4).

$$v_{x,trailer} = v_x + dsin(\gamma)(\dot{\psi} + \dot{\gamma}) \tag{2.4}$$

In [26], the assumption is made that $v_{x,trailer} \approx v_x$, which follows from the assumption that the hitch angle, γ, remains small. The second relationship is between the tractor and trailer lateral velocities given by Equation (2.5).

$$v_{y,trailer} = v_y - (c + dcos(\gamma))\dot{\psi} - dcos(\gamma)\dot{\gamma} \tag{2.5}$$

Using the above two relationships, Equation (2.2) can be expressed in terms of only the masses, inertias, and the generalized coordinates. Once in this form, the Lagrange equation, shown in Equation (2.6), can be applied, where Q_q represents the generalized force for each generalized coordinate [28].

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_q \tag{2.6}$$

The generalized forces represent how external forces and moments are applied to the system, which in this case are due to the forces at the tires. Once it is evaluated, Equation (2.6) produces three equations of motion for v_y, $\dot{\psi}$, and $\dot{\gamma}$.

The equations resulting from Equation (2.6) are not evaluated here because they are most easily evaluated in software with a symbolic math toolbox, although a more detailed derivation is provided in [26]. These equations of motion are augmented with integral states for yaw and hitch angle and result in the full form of the model found in [26] and shown in Equation (2.7), where M

is the mass matrix, K is the stiffness matrix, and F is the input matrix.

$$M\dot{\boldsymbol{x}}_{\boldsymbol{w}} = K\boldsymbol{x}_{\boldsymbol{w}} + F\delta \tag{2.7}$$

$$\dot{\boldsymbol{x}}_{\boldsymbol{w}} = M^{-1}K\boldsymbol{x}_{\boldsymbol{w}} + M^{-1}F\delta \tag{2.8}$$

$$A_w = M^{-1}K, \;\; B_w = M^{-1}F \tag{2.9}$$

The M, K, and F matrices used in this thesis are provided in full in Appendix A. Note that although A_w and B_w depend on the steer angle and hitch angle, in this thesis these matrices are always constructed around $\delta = 0$ and $\gamma = 0$. This results in a linear model which has v_x as a parameter. The states of this model are shown in Equation (2.10) and are denoted with a subscript w to avoid confusion.

$$\boldsymbol{x}_w = \begin{bmatrix} v_y \\ \dot{\psi} \\ \psi \\ \dot{\gamma} \\ \gamma \end{bmatrix} \tag{2.10}$$

2.2 Road Representation

A way of mechanizing the position of the vehicle is also needed in the prediction model. The position can be mechanized in a local cartesian frame by numerically integrating Equations (2.11-2.12), where x_l and y_l are the position in the local frame.

$$\dot{x}_l = v_x cos(\psi) - v_y sin(\psi) \tag{2.11}$$

$$\dot{y}_l = v_x sin(\psi) + v_y cos(\psi) \tag{2.12}$$

However, a curvilinear coordinate frame, commonly referred to as a Frenet frame, can be constructed that follows a reference curve instead [29]. Because the application of this work is for on-road driving, this coordinate system can be constructed along the centerline of a lane and provides a useful way to mechanize position with respect to the road. Equations (2.13)-(2.15) are used to mechanize the position of the vehicle with respect to this curvilinear frame, where s is the distance moved along the reference curve (along track), e is the cross track distance from the reference curve, and ψ_{rel} is the relative yaw from the reference curve [29].

$$\dot{s} = \frac{v_x cos(\psi_{rel}) - v_y sin(\psi_{rel})}{1 - \kappa e} \tag{2.13}$$

$$\dot{e} = v_x sin(\psi_{rel}) + v_y cos(\psi_{rel}) \tag{2.14}$$

$$\dot{\psi}_{rel} = \dot{\psi} - \kappa \dot{s} \tag{2.15}$$

The curvature of the road, κ is also needed, and is a function of the along track distance if the curvature of the road is not constant. The along track coordinate, s, is considered to be positive in the direction of travel, lateral position is positive to the left of the curve, and relative heading is positive counter-clockwise. Road curvature is also considered to be positive for left turns. Figure 2.2 illustrates the curvilinear and cartesian coordinate frames and sign conventions used throughout this thesis.

Use of the curvilinear frame provides several benefits in the MPC design. Most importantly, it allows for simple calculation of lateral distance from the reference curve and allows linear constraints to be placed on the cross-track (lateral) distance state, e. Furthermore, the linearization of these equations is simple because, for on-road driving, relative heading and lateral position remain small. Therefore, linearizing about zero for these variables in Equations (2.13-2.15) is an appropriate assumption. The linearization is discussed in more detail in Section 2.4.

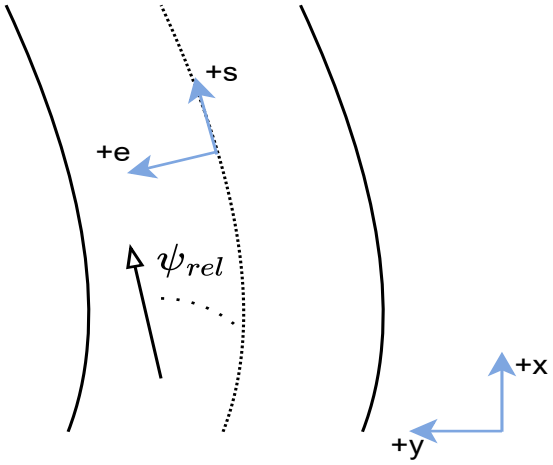

Figure 2.2: Curvilinear and Cartesian Coordinate Frames

2.3 Steering Actuator

Steering actuation on the vehicle used in this thesis is achieved by a motor which receives a commanded steer angle from the user and acts on the steering column of the vehicle. However, in systems such as these, an internal control loop handles the actual tracking of the desired steer angle sent by the user. The action of this internal control loop is hidden from the designer, but some assumptions can be made on how the entire steering actuation system operates. At some point in the actuation, a voltage is supplied to the motor which produces a torque that moves the wheel to the desired position. However, there is a limit on how much torque can be used by the motor and there can also be limits on the steer rate the motor is allowed to achieve. Figure 2.3 shows a block diagram of the steering actuation system, where V is the voltage at the motor and saturation is shown at both of the possible locations. G(s) represents the lumped dynamics from voltage at the motor to angular velocity of the steering column.

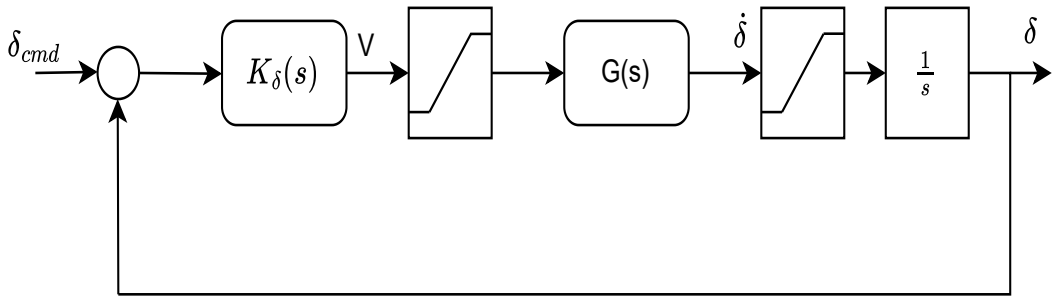

Figure 2.3: Steering Actuation System Block Diagram

The entire steering system model is simplified to a first order model, shown in Equation (2.16), for inclusion in the MPC design. The response of this model is defined by the time constant, τ_δ, but is shown in Equation (2.16) with a_δ which is equal to $\frac{1}{\tau_\delta}$.

$$\dot{\delta} = a_\delta(\delta_{cmd} - \delta) \tag{2.16}$$

Even though a simplified model for the steering control system is used in the design, an understanding of the full actuation is helpful to understand the behavior of the steering actuator and demonstrates the need for hard constraints on steering rate in the MPC.

2.4 Full Prediction Model and Linearization

Combining the position mechanization equations, the lateral dynamic model, and the steering dynamics, the full prediction model can be defined in state space form by Equation (2.17), where

$\dot{\boldsymbol{x}}$ is a function of the states $\boldsymbol{x}$, control input $\boldsymbol{u}$, and the model parameters $\boldsymbol{p}$.

$$\underbrace{\begin{bmatrix} \dot{s} \\ \dot{e} \\ \dot{\psi}_{rel} \\ \dot{v}_x \\ \dot{\boldsymbol{x}}_w \\ \dot{\delta} \end{bmatrix}}_{\dot{\boldsymbol{x}}} = \underbrace{\begin{bmatrix} \frac{1}{1-\kappa e}(v_x cos(\psi_{rel}) - v_y sin(\psi_{rel})) \\ v_x sin(\psi_{rel}) + v_y cos(\psi_{rel}) \\ \dot{\psi} - \kappa \dot{s} \\ 0 \\ A_w \boldsymbol{x_w} + B_w \delta \\ a_\delta(\delta_{cmd} - \delta) \end{bmatrix}}_{\boldsymbol{f(x,u,p)}} \tag{2.17}$$

Note that the dimension of $\dot{\boldsymbol{x}}_w$ from the lateral dynamic model given previously by Equation (2.8), is 5x1, but the lateral dynamic model has been inserted as a block for simplicity. The longitudinal velocity of the vehicle, v_x, was also included as a state to complete the mechanization equations. However, since longitudinal control is not considered in this design, $\dot{v}_x$ is always assumed to be zero, and v_x is instead updated each time the model is linearized.

The model now includes all of the parameters shown in Table 2.1. These include the positions of each axle, trailer hitch and center of gravity locations, total cornering stiffness at each axle, curvature of the road, and the time constant of the steering actuator. A linearized model was created by taking the partial derivatives of each component of $\dot{X}$ with respect to each component of X, resulting in the continuous time A and B matrices of the linearized model given by Equation (2.18) [30].

$$A_c = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}|_{x_0,u_0}, \quad B_c = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}|_{x_0,u_0} \tag{2.18}$$

Table 2.1: Model Parameters

Parameter	Description	Units
$C_{\alpha,1}$	Total cornering stiffness of axle 1.	N/rad
$C_{\alpha,2}$	Total cornering stiffness of axle 2.	N/rad
$C_{\alpha,3}$	Total cornering stiffness of axle 3.	N/rad
$C_{\alpha,4}$	Total cornering stiffness of axle 4.	N/rad
$C_{\alpha,5}$	Total cornering stiffness of axle 5.	N/rad
m_1	Mass of the cab.	kg
m_2	Mass of the trailer.	kg
J_1	Yaw inertia of the cab.	$kg * m^2$
J_2	Yaw inertia of the trailer.	$kg * m^2$
a	Distance from cab's center of gravity (CG) to axle 1.	m
b_1	Distance from cab's CG to axle 2.	m
b_2	Distance from cab's CG to axle 3.	m
c	Distance from cab's CG to hitch point.	m
d	Distance from hitch point to trailer's CG.	m
f_1	Distance from hitch point to axle 4.	m
f_2	Distance from hitch point to axle 5.	m
κ	Road curvature.	m^{-1}
a_δ	Steering actuator parameter.	sec^{-1}

Equation (2.19) shows the structure of the continuous A and B matrices in the full linearized model.

$$\underbrace{\begin{bmatrix} \dot{s} \\ \dot{e} \\ \dot{\psi_{rel}} \\ \dot{v_x} \\ \dot{v_y} \\ \ddot{\psi} \\ \dot{\psi} \\ \ddot{\gamma} \\ \dot{\gamma} \\ \dot{\delta} \end{bmatrix}}_{\dot{\boldsymbol{x}}} = \underbrace{\begin{bmatrix} 0 & A_{12} & A_{13} & A_{14} & A_{15} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & A_{23} & A_{24} & A_{25} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & A_{33} & A_{34} & A_{35} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & & & & & & 0 \\ 0 & 0 & 0 & 0 & & \vdots & & & \vdots & 0 \\ 0 & 0 & 0 & 0 & \dots & \boldsymbol{A_w} & \dots & & \boldsymbol{B_w} & 0 \\ 0 & 0 & 0 & 0 & & \vdots & & & \vdots & 0 \\ 0 & 0 & 0 & 0 & & & & & & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -a_\delta \end{bmatrix}}_{A_c} \underbrace{\begin{bmatrix} s \\ e \\ \psi_{rel} \\ v_x \\ v_y \\ \dot{\psi} \\ \psi \\ \dot{\gamma} \\ \gamma \\ \delta \end{bmatrix}}_{\boldsymbol{x}} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ a_\delta \end{bmatrix}}_{B_c} \underbrace{\begin{bmatrix} \delta_{cmd} \end{bmatrix}}_{\boldsymbol{u}} \quad (2.19)$$

These matrices must be evaluated at reference values of the states and control variables, $\boldsymbol{x_0}$ and $\boldsymbol{u_0}$. The interaction between v_x and the lateral dynamics were not considered in this work, so the partial derivatives of $\dot{\boldsymbol{x}}_w$ with respect to v_x were neglected in the linearization.

Because the lateral dynamic model is already linearized, A_w and B_w can just be inserted into A_c. However, the terms that result from linearizing the mechanization equations also have to be calculated by taking the partial derivative of each equation with respect to each state, and these calculations are shown in Equations (2.20-2.30).

$$A_{12} = \frac{\partial \dot{s}}{\partial e} = \frac{\kappa(v_x cos(\psi_{rel}) - v_y sin(\psi_{rel}))}{(1-\kappa e)^2} \tag{2.20}$$

$$A_{13} = \frac{\partial \dot{s}}{\partial \psi_{rel}} = \frac{-v_x sin(\psi_{rel}) - v_y cos(\psi_{rel})}{1-\kappa e} \tag{2.21}$$

$$A_{14} = \frac{\partial \dot{s}}{\partial v_x} = \frac{cos(\psi_{rel})}{1-\kappa e} \tag{2.22}$$

$$A_{15} = \frac{\partial \dot{s}}{\partial v_y} = \frac{-sin(\psi_{rel})}{1-\kappa e} \tag{2.23}$$

$$A_{23} = \frac{\partial \dot{e}}{\partial \psi_{rel}} = v_x cos(\psi_{rel}) - v_y sin(\psi_{rel}) \tag{2.24}$$

$$A_{24} = \frac{\partial \dot{e}}{\partial v_x} = sin(\psi_{rel}) \tag{2.25}$$

$$A_{25} = \frac{\partial \dot{e}}{\partial v_y} = cos(\psi_{rel}) \tag{2.26}$$

$$A_{33} = \frac{\partial \dot{\psi}_{rel}}{\partial \psi_{rel}} = -\kappa(-v_x sin(\psi_{rel}) - v_y cos(\psi_{rel})) \tag{2.27}$$

$$A_{34} = \frac{\partial \dot{\psi}_{rel}}{\partial v_x} = \frac{\kappa sin(\psi_{rel}}{1-\kappa e} \tag{2.28}$$

$$A_{35} = \frac{\partial \dot{\psi}_{rel}}{\partial v_y} = \frac{-\kappa cos(\psi_{rel})}{1-\kappa e} \tag{2.29}$$

$$A_{36} = \frac{\partial \dot{\psi}_{rel}}{\partial \dot{\psi}} = 1.0 \tag{2.30}$$

2.5 Discretization

Once the continuous time A and B matrices have been constructed, they must be discretized for use in the MPC design. The discrete time A and B matrices are defined by Equations (2.31-(2.32)) where T_s is the sample time [30].

$$A_d = e^{A_c T_s} \tag{2.31}$$

$$B_d = \int_0^{T_s} e^{A\tau} d\tau B_c \tag{2.32}$$

The discrete time matrices can be computed exactly, assuming a zero order hold on the input, by placing A_c and B_c into a larger zero padded matrix, multiplying by the sample time, and calculating the matrix exponential. The discrete time matrices can be extracted from the resulting matrix. This method, known as the Van Loan method, is shown in Equations (2.33-2.34)[31].

$$S = \begin{bmatrix} A_c & B_c \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{2.33}$$

$$e^{ST_s} = \begin{bmatrix} A_d & B_d \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix} \tag{2.34}$$

In the real-time implementation, the matrix exponential was computed using the C++ library Eigen [32]. The discrete time system is defined by Equation (2.35), where f_d denotes the discrete state space equation.

$$\boldsymbol{x_{k+1}} = \underbrace{A_d \boldsymbol{x_k} + B_d \boldsymbol{u_k}}_{f_d(\boldsymbol{x_k}, \boldsymbol{u_k}, \boldsymbol{p})} \tag{2.35}$$

Chapter 3

MPC Design

3.1 Optimal Control Problem

The first step in the design of the MPC is to define the Optimal Control Problem (OCP) that is to be solved. The OCP is made up of the cost and constraints that are applied over a certain horizon, and is shown in Equation (3.1).

$$\min_{\delta \in [\delta_{min}, \delta_{max}]} \int_0^{t_f} [(\boldsymbol{x} - \boldsymbol{x_{ref}})^T Q (\boldsymbol{x} - \boldsymbol{x_{ref}}) + R\delta_{cmd}^2 + w_{\dot{\delta}} \dot{\delta}^2] dt + (\boldsymbol{x(t_f)} - \boldsymbol{x_{ref}})^T Q_{tf} (\boldsymbol{x(t_f)} - \boldsymbol{x_{ref}}) \tag{3.1}$$

subject to:

$$\dot{\boldsymbol{x}} = A_c \boldsymbol{x} + B_c \delta_{cmd}$$

$$\boldsymbol{x(t_0)} = \boldsymbol{x_0}$$

$$e_{min} \leq e \leq e_{max}$$

$$\gamma_{min} \leq \gamma \leq \gamma_{max}$$

$$a_{y,min} \leq a_y \leq a_{y,max}$$

$$\delta_{min} \leq \delta \leq \delta_{max}$$

$$\dot{\delta}_{min} \leq \dot{\delta} \leq \dot{\delta}_{max}$$

For this implementation, the cost consists of a quadratic function of the error between states and reference states and control variables integrated over the horizon, with a terminal cost applied to

the difference between states and reference at the final time in the horizon. A cost is also applied to steer rate, $\dot{\delta}$, to penalize aggressive changes in steering. The constraints include equality constraints for the dynamics and inequality constraints on the state and control variables, as well as a constraint on steer rate. The Q matrix is composed of the cost associated with the state error for each state on its diagonal. The structure of Q is shown in Equation (3.2).

$$Q = \begin{bmatrix} q_1 & & & \\ & q_2 & & \\ & & \ddots & \\ & & & q_{10} \end{bmatrix} \tag{3.2}$$

The lateral acceleration, a_y, is defined as the derivative of body frame velocity with respect to the inertial frame as shown in Equation (3.3).

$$a_y = (\frac{d^2y}{dt^2})_{inertial} \tag{3.3}$$

Therefore, the a_y constraint must be defined using the sum of $\dot{v}_y$ and the yaw rate multiplied by longitudinal velocity, shown in Equation (3.4) [33].

$$a_y = \dot{v}_y + \dot{\psi} v_{x,ref} \tag{3.4}$$

The reference longitudinal velocity is used to compute this equation so that the constraint remains a linear function of the states. Constraining lateral acceleration is necessary so that the MPC will only plan trajectories that don't exceed the handling or rollover limits of the vehicle.

3.2 Reference Trajectory Construction and Model Evaluation

Some additional assumptions are used to evaluate the linearized model used at each time in the MPC horizon. To accurately model the problem while still allowing for the resulting OCP to be solved as a quadratic program, the model was linearized about a reference trajectory at discrete

points in the prediction horizon. This procedure results in a linearized model for each discrete point in the horizon that can be used to set linear dynamics constraints. Because the longitudinal velocity is known a priori, the longitudinal position along the road at the given time in the horizon can be calculated. Additionally, the longitudinal velocities throughout the prediction horizon are assumed to be known a priori, and the longitudinal velocity profile is denoted $v_{x,ref}(t)$. As a result, the longitudinal position s_k is used to look up the curvature of the road. Figure 3.1 illustrates the information that is calculated along the reference line and is assumed to be known at each time in the horizon a priori. Equation (3.5) shows the values that are used to construct the reference states, which are used in the evaluation of the cost.

$$\boldsymbol{x}_{ref}^{(k)} = \begin{bmatrix} s_{ref}^{(k)} \\ e_{ref}^{(k)} \\ \psi_{rel,ref}^{(k)} \\ v_{x,ref}^{(k)} \\ v_{y,ref}^{(k)} \\ \dot{\psi}_{ref}^{(k)} \\ \psi_{ref}^{(k)} \\ \dot{\gamma}_{ref}^{(k)} \\ \gamma_{ref}^{(k)} \\ \delta_{ref}^{(k)} \end{bmatrix} = \begin{bmatrix} \int_0^{t^{(k)}} v_{x,ref}(t) \\ e_{ref} \\ 0 \\ v_{x,ref}(t) \\ 0 \\ v_{x,ref}(t)\kappa(s_{ref}^{(k)}) \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.5}$$

The reference states and control variables at each point in the horizon are now used to evaluate the cost. Additionally, the road curvature at each time is used to create the linearized discrete A and B matrices, A_d and B_d. Note that for some states such as heading, the cost weighting in the Q matrix can also be set to zero, so the reference value has no impact. Furthermore, some states such as hitch angle and hitch rate may have a reference value of zero but can be constrained to some bounds instead. In these cases, the cost on these states is applied to favor these values remaining close to zero.

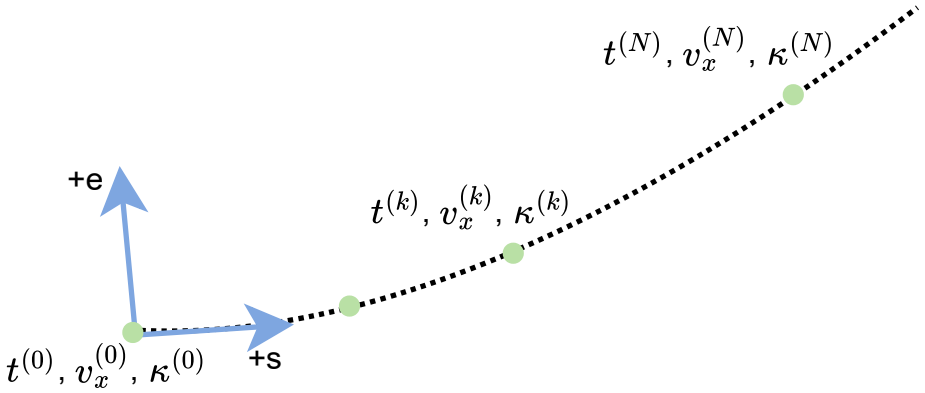

Figure 3.1: A Priori Information Along the Reference Curve

3.3 Formulation as a Quadratic Program

The OCP presented by Equation (3.1) is solved using direct methods in this work. This means the original OCP, which is defined in continuous time, is first discretized and then solved using numerical optimization methods [34]. The decision variables of this new optimization problem are the states and control variables at each discrete time in the horizon. The dynamics constraints are applied as linear equality constraints relating the states and control at each time step to the states at the next time. The discrete version of the problem is given by Equation (3.6), where J is the cost, and $g(z)$ is the function containing all the constraints.

$$J(z) = \sum_{k=0}^{N} [(\boldsymbol{x^{(k)}} - \boldsymbol{x_{ref}^{(k)}})^T Q (\boldsymbol{x_k} - \boldsymbol{x_{ref}^{(k)}}) + R {\delta_{cmd}^{(k)}}^2 + w_{\dot{\delta}} \dot{\delta}^{(k)^2}] \quad +$$

$$(\boldsymbol{x^{(N)}} - \boldsymbol{x_{ref}^{(N)}})^T Q_{tf} (\boldsymbol{x^{(N)}} - \boldsymbol{x_{ref}^{(N)}})$$

Subject to:

$$l \leq g(z) \leq u \tag{3.6}$$

The vector of decision variables, shown in (3.7) is denoted by z and is made up of the states and control variables at each time step labeled with a superscript from 0 to N.

$$z = \begin{bmatrix} \boldsymbol{x}^{(0)^T} & \delta_{cmd}^{(0)} & \dots & \boldsymbol{x}^{(k)^T} & \delta_{cmd}^{(k)} & \dots & \boldsymbol{x}^{(N)^T} & \delta_{cmd}^{(N)} \end{bmatrix}^T \tag{3.7}$$

When the objective function is a quadratic function of the decision variables, and all equality and inequality constraints are linear functions of the decision variables, the optimization problem can be solved as a Quadratic Program (QP) [35]. A QP is a type of constrained optimization problem that is quadratic in the decision variables and has linear equality and inequality constraints. Solving the OCP as a QP provides computational benefits because there are algorithms for solving QP's that are computationally efficient, making them suitable for real-time implementation. In general, a QP takes the form defined in Equation (3.8),

$$\begin{gathered} \min z^T H z + b^T z \\ \text{subject to:} \\ l \leq Gz \leq u \end{gathered} \tag{3.8}$$

where H is the hessian of the cost, b is the gradient of the cost, and G is the gradient of the constraints function [36].

To form the QP, first the gradient and hessian of the cost are constructed. The gradient is first taken for a single time step, k, which is shown in Equation (3.9).

$$b_k = \frac{\partial J_k}{\partial z_k} = \begin{bmatrix} \frac{\partial J_k}{\partial s} \\ \frac{\partial J_k}{\partial e} \\ \frac{\partial J_k}{\partial \psi_{rel}} \\ \frac{\partial J_k}{\partial v_x} \\ \frac{\partial J_k}{\partial v_y} \\ \frac{\partial J_k}{\partial \dot{\psi}} \\ \frac{\partial J_k}{\partial \psi} \\ \frac{\partial J_k}{\partial \dot{\gamma}} \\ \frac{\partial J_k}{\partial \gamma} \\ \frac{\partial J_k}{\partial \delta} \\ \frac{\partial J_k}{\partial \delta_{cmd}} \end{bmatrix} = \begin{bmatrix} 2q_1(s - s_{ref}) \\ 2q_2(e - e_{ref}) \\ 2q_3(\psi_{rel} - \psi_{rel,ref}) \\ 2q_4(v_x - v_{x,ref}) \\ 2q_5(v_y - v_{y,ref}) \\ 2q_6(\dot{\psi} - \dot{\psi}_{ref}) \\ 2q_7(\psi - \psi_{ref}) \\ 2q_8(\dot{\gamma} - \dot{\gamma}_{ref}) \\ 2q_9(\gamma - \gamma_{ref}) \\ 2q_{10}(\delta - \delta_{ref}) - 2w_{\dot{\delta}} a_\delta^2(\delta_{cmd} - \delta) \\ 2q_{11}\delta_{cmd} + 2w_{\dot{\delta}} a_\delta^2(\delta_{cmd} - \delta) \end{bmatrix} \tag{3.9}$$

The gradient of J at each stage is then concatenated to create b, which is the gradient of J with respect to all of the decision variables, as shown below in Equation (3.10).

$$b = \frac{\partial J}{\partial z} = \begin{bmatrix} b_0 \\ \vdots \\ b_k \\ \vdots \\ b_N \end{bmatrix} \tag{3.10}$$

The hessian, H, of the cost is also created by taking the derivative of each row in the gradient with respect to each of the decision variables. The hessian for one timestep is shown in Equation

(3.11).

$$H_k = \begin{bmatrix} 2q_1 & & & & & & & & & & \\ & 2q_2 & & & & & & & & & \\ & & 2q_3 & & & & & & & & \\ & & & \ddots & & & & & & & \\ & & & & \ddots & & & & & & \\ & & & & & \ddots & & & & & \\ & & & & & & 2q_9 & & & & \\ & & & & & & & 2q_{10} + 2w_{\dot{\delta}}a_\delta^2 & -2w_{\dot{\delta}}a_\delta^2 & \\ & & & & & & & -2w_{\dot{\delta}}a_\delta^2 & 2q_{11} + 2w_{\dot{\delta}}a_\delta^2 & \end{bmatrix} \tag{3.11}$$

The full hessian is created by concatenating the hessian from each time step as shown in Equation (3.12).

$$H = \begin{bmatrix} H_0 & & & & & \\ & H_1 & & & & \\ & & H_2 & & & \\ & & & \ddots & & \\ & & & & H_{N-1} & \\ & & & & & H_N \end{bmatrix} \tag{3.12}$$

The dynamics constraints are also converted to their discrete version, and the constraints are rearranged to form the G matrix along with the upper and lower bounds required for the QP. To create G, it is useful to split the constraints into equality and inequality constraints, and then concatenate them together as shown in Equation (3.13).

$$G_{eq} = \begin{bmatrix} \boldsymbol{I}_{n_x} & & & & & & \\ \boldsymbol{A}_d^{(0)} & \boldsymbol{B}_d^{(0)} & -\boldsymbol{I}_{n_x} & & & & \\ & & \boldsymbol{A}_d^{(1)} & \boldsymbol{B}_d^{(1)} & -\boldsymbol{I}_{n_x} & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \boldsymbol{A}_d^{(N)} & \boldsymbol{B}_d^{(N)} & -\boldsymbol{I}_{n_x} \end{bmatrix} \tag{3.13}$$

The first 10 rows of G_{eq} applies the initial condition constraint $X(k = 0) = X_0$. Each block afterwards enforces the dynamics constraint, $A_{d,k}\boldsymbol{x_k}, +B\delta_{cmd} = \boldsymbol{x}_{k+1}$. The matrix of inequality constraints, G_{ineq}, has the structure shown in Equation (3.14).

$$G_{\text{ineq}} = \begin{bmatrix} G_{\text{ineq}}^{(0)} & & & \\ & G_{\text{ineq}}^{(1)} & & \\ & & \ddots & \\ & & & G_{\text{ineq}}^{(N)} \end{bmatrix} \tag{3.14}$$

The block structure for time step k is shown in Equation (3.15), where I_{n_x} and I_{n_u} are identity matrices of size $n_x \times n_x$ and $n_u \times n_u$, respectively.

$$G_{ineq}^{(k)} = \begin{bmatrix} G_{ineq,x}^{(k)} \\ G_{ineq,u}^{(k)} \\ G_{ineq,\dot{\delta}}^{(k)} \\ G_{ineq,a_y}^{(k)} \end{bmatrix} = \begin{bmatrix} -I_{n_x} & \mathbf{0}_{n_x \times n_u} \\ I_{n_x} & \mathbf{0}_{n_x \times n_u} \\ \mathbf{0}_{n_u \times n_u} & -I_{n_u} \\ \mathbf{0}_{n_u \times n_u} & I_{n_u} \\ -\frac{\partial \dot{\delta}^{(k)}}{\partial x^{(k)}} & -\frac{\partial \dot{\delta}^{(k)}}{\partial u^{(k)}} \\ \frac{\partial \dot{\delta}^{(k)}}{\partial x^{(k)}} & \frac{\partial \dot{\delta}^{(k)}}{\partial u^{(k)}} \\ -\frac{\partial a_y^{(k)}}{\partial x^{(k)}} & -\frac{\partial a_y^{(k)}}{\partial u^{(k)}} \\ \frac{\partial a_y^{(k)}}{\partial x^{(k)}} & \frac{\partial a_y^{(k)}}{\partial u^{(k)}} \end{bmatrix} \tag{3.15}$$

Because $\dot{\delta}$ is an element of $\dot{x}$, the partial of $\dot{\delta}$ with respect to x is obtained by extracting the row of A_c that corresponds to $\dot{\delta}$. Similarly, to obtain the partial of $\dot{\delta}$ with respect to the input, u, the corresponding row of B_c is extracted. Equations (3.16-3.17) show the partial derivatives of $\dot{\delta}$ with respect to the state and control variables, where $A_c^{(k)}(10,:)$ denotes the 10th row of matrix A_c at time step k, and $B_c^{(k)}(10,:)$ indicates the 10th row of $B_c^{(k)}$ at time step k.

$$\frac{\partial \dot{\delta}^{(k)}}{\partial x^{(k)}} = A_c^{(k)}(10,:) \tag{3.16}$$

$$\frac{\partial \dot{\delta}^{(k)}}{\partial u^{(k)}} = B_c^{(k)}(10,:) \tag{3.17}$$

The partial derivatives of a_y are also needed. However, a_y is a function of both $\dot{v}_y$ and $\dot{\psi}$, so its partial with respect to the states is slightly more complex and is shown in Equation (3.18).

$$\frac{\partial a_y^{(k)}}{\partial x^{(k)}} = \begin{bmatrix} A_c^{(k)}(5,1) & \dots & A_c^{(k)}(5,5) & A_c^{(k)}(5,6) + v_{x,ref}^{(k)} & A_c^{(k)}(5,7) & \dots & A_c^{(k)}(5,10) \end{bmatrix} \tag{3.18}$$

Because the equation for a_y does not directly contain the input, u, the partial with respect to u is zero, as shown in Equation (3.19).

$$\frac{\partial a_y^{(k)}}{\partial u^{(k)}} = 0 \tag{3.19}$$

The complete constraint matrix, G, is obtained by concatenating G_{eq} and G_{ineq}, and is given in Equation (3.20).

$$G = \begin{bmatrix} G_{eq} \\ G_{ineq} \end{bmatrix} \tag{3.20}$$

The lower and upper bounds on the decision variables of the QP are then defined to match the structure of the constraint matrices. First, the equality bounds are created by setting the lower and upper bound of the states at $k = 0$ to force the initial states to be equal to $\boldsymbol{x}_0$. All other values in the equality bounds are set to zero so that the dynamics constraints are enforced at each time step. The bounds on the equality constraints, denoted l for lower and u for upper, are given in Equation (3.21).

$$l_{eq} = \begin{bmatrix} \boldsymbol{x}_0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}, \quad u_{eq} = \begin{bmatrix} \boldsymbol{x}_0 \\ 0 \\ 0 \\ \vdots \end{bmatrix} \tag{3.21}$$

The inequality bounds, given in Equation (3.22), are constructed to match the structure of G_{ineq} with the desired maximum and minimum values at each time step.

$$l_{ineq} = \begin{bmatrix} \boldsymbol{x}_{min}^{(0)} \\ \boldsymbol{u}_{min}^{(0)} \\ \dot{\delta}_{min}^{(0)} \\ a_{y,min}^{(0)} \\ \vdots \\ \boldsymbol{x}_{min}^{(k)} \\ \boldsymbol{u}_{min}^{(k)} \\ \dot{\delta}_{min}^{(k)} \\ a_{y,min}^{(k)} \\ \vdots \\ \boldsymbol{x}_{min}^{(N)} \\ \boldsymbol{u}_{min}^{(N)} \\ \dot{\delta}_{min}^{(N)} \\ a_{y,min}^{(N)} \end{bmatrix}, \quad u_{ineq} = \begin{bmatrix} \boldsymbol{x}_{max}^{(0)} \\ \boldsymbol{u}_{max}^{(0)} \\ \dot{\delta}_{max}^{(0)} \\ a_{y,max}^{(0)} \\ \vdots \\ \boldsymbol{x}_{max}^{(k)} \\ \boldsymbol{u}_{max}^{(k)} \\ \dot{\delta}_{max}^{(k)} \\ a_{y,max}^{(k)} \\ \vdots \\ \boldsymbol{x}_{max}^{(N)} \\ \boldsymbol{u}_{max}^{(N)} \\ \dot{\delta}_{max}^{(N)} \\ a_{y,max}^{(N)} \end{bmatrix} \tag{3.22}$$

The equality and inequality bounds, shown in Equation (3.23), are also concatenated, matching the structure of G.

$$l = \begin{bmatrix} l_{eq} \\ l_{ineq} \end{bmatrix}, \quad u = \begin{bmatrix} u_{eq} \\ u_{ineq} \end{bmatrix} \tag{3.23}$$

The optimal solution to the QP defined by Equation (3.8) is referred to as z^*, and the optimal control and state variables in this solution are u^* and x^* respectively.

3.4 Calculation of Error Bounds from Parameter Sensitivity

A constraint tightening method is also applied to improve the robustness of the design. To account for the uncertainty of the vehicle parameters, rather than creating error bounds based on an additive disturbance, a sensitivity based approach used for motion planning of a quadrotor in [15] and for closed loop MPC on a quadrotor in [16] was pursued instead. This method uses the closed loop sensitivity of the model with respect to its parameters to propagate approximate error bounds on the prediction. Ideally, the error bounds capture the spread of possible trajectories of the vehicle with different combinations of parameters. Figure 3.2 illustrates the error bounds around a nominal trajectory, where ρ represents the radius of the tube in a certain direction [16].

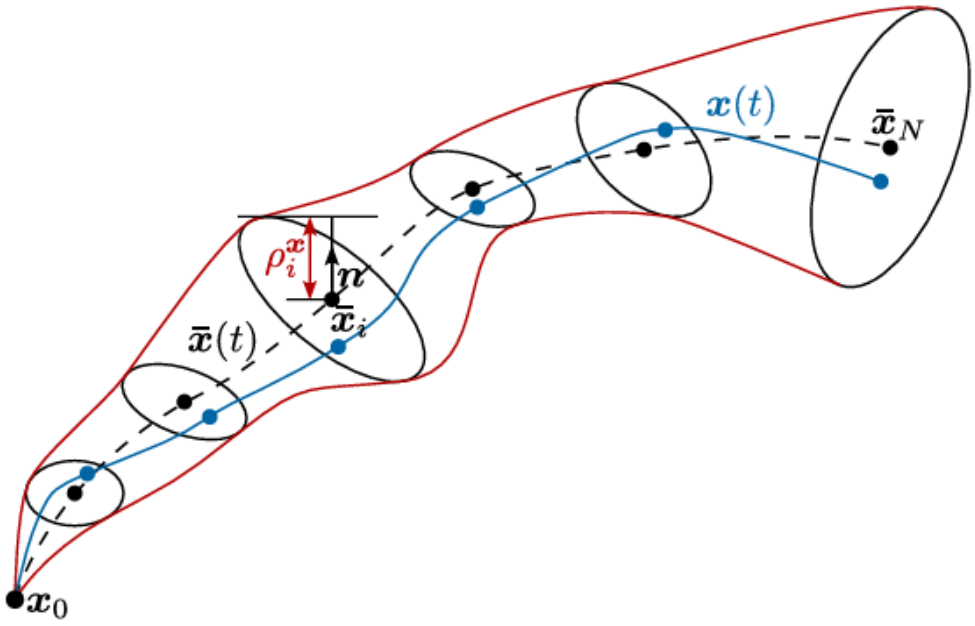

Figure 3.2: Visualization of Error Tube [16]

The sensitivity based approach has the advantage that the size of the error tube scales with the magnitude of the bounds on the model parameters and the feedback gains that are applied. The size of the tube can also scale during different parts of the horizon where the parameters will have more effect. For example, when driving on a straight road where steer angles are small, a tube created based on an additive disturbance would give the same error bounds as it would during

an aggressive lane change maneuver. However, a tube created based on parameter sensitivity is able to predict very small errors during straight driving, when parameter error is not affecting the prediction, but larger errors during a lateral maneuver.

To calculate the error bounds around a nominal trajectory using the sensitivity based approach, first the closed loop state sensitivity, Π, is needed throughout the horizon. The closed loop state sensitivity is defined as the partial derivative of the states with respect to the parameters, and is shown in Equation (3.24).

$$\Pi = \frac{\partial x(t)}{\partial p}|_{p_0} \tag{3.24}$$

This quantity represents the effect that variation in parameters has on the prediction of the states. The discrete time A and B matrices and two other quantities are needed to calculate Π throughout the horizon. The Jacobian of the control law with respect to the states is needed and is referred to as F.

$$F = \frac{\partial u(t)}{\partial x} \tag{3.25}$$

A constant gain feedback controller is assumed to be applied to track the trajectory produced by the MPC, which follows the equation shown in Equation (3.26).

$$u = -Kx_{err} \tag{3.26}$$

Therefore, F is just the negative feedback gain matrix, K, from this control law, as shown in Equation (3.27).

$$F = -K \tag{3.27}$$

The Jacobian of the parameters with respect to the discrete state space system, M, also needs to be computed for use in the calculation of Π, and the definition is shown in Equation (3.28).

$$M_k = \frac{\partial f_d}{\partial p}|_{\bar{x_k},\bar{u_k},\bar{p}} \tag{3.28}$$

With a large number of parameters and states, M can become difficult to derive analytically. As in prior work that used this method, M was calculated using the Autodiff library in C++, which automatically calculates this Jacobian given the discrete state space equations [37].

Using the update equation defined below in Equation (3.29), Π can be calculated for each time in the horizon.

$$\Pi_{k+1} = (A_k + B_k F_k)\Pi_k + M_k \tag{3.29}$$

Note that the sensitivity of the states to the parameters at $k = 0$ is the zero matrix, because x_0 is the current known state of the vehicle. In other words, there is an initial condition that $\Pi_0 = \mathbf{0}$, which is needed to use the update equation given by Equation (3.29).

Once Π is obtained at each point in the horizon, the deviation of the parameters can be mapped into an error in the prediction. For this method, the bounds on the parameter deviations are assumed to be ellipsoidal, and are defined by the matrix shown in Equation (3.30).

$$W = diag(\Delta p_{max}^2) \tag{3.30}$$

The matrix W is constructed by placing the squared maximum parameter deviations on the diagonal. This matrix could also be interpreted as the covariance of the parameter estimates if the parameter estimates are Gaussian.

Finally, Equation (3.31) maps the parameter bounds into ellipsoidal bounds on the states, where P_x is the matrix that defines the ellipsoid.

$$P_x = \Pi W \Pi^T \tag{3.31}$$

The radius of the bounds that correspond to each state can be extracted by taking the square root of the diagonal element corresponding to each state. This is shown in Equation (3.32), where $P_x(i,i)$ denotes the i'th diagonal entry of the matrix and $\rho_{x,i}$ is the radius of the bound on the i'th state.

$$\rho_{x,i} = \sqrt{P_x(i,i)} \tag{3.32}$$

For tightening of the input constraints, first the state sensitivity is mapped into an input sensitivity matrix, Θ, as shown in Equation (3.33).

$$\Theta = -K\Pi \tag{3.33}$$

In a similar way to how Π maps variation in parameters to deviation in states, Θ maps deviation in parameters into the deviation in control input that may result based on the control law. The input ellipsoid matrix and radius are then calculated in the same way as in Equation (3.31), and this process is completed using Equations (3.34-3.35).

$$P_u = \Theta W \Theta^T \tag{3.34}$$

$$\rho_{u,j} = \sqrt{P_u(j,j)} \tag{3.35}$$

In Equation (3.35), $\rho_{u,j}$ is the bound on the j'th input.

When constraint tightening is applied, the original constraints on states and inputs are modified by subtracting the radius on the error bounds. The intuition is that by tightening the constraints based on the error bounds, the MPC is forced to find a solution that will satisfy the original constraints for any trajectory within the error bounds. As an example, the modified constraints for the lateral position are shown in Equation (3.36).

$$e_{min} + \rho_{x,2} \leq e \leq e_{max} - \rho_{x,2} \tag{3.36}$$

In addition to tightening the state and input constraints, it may also be necessary to tighten constraints on other values that are functions of the states and inputs, such as steer rate. This can be accomplished using the relationship shown in Equation (3.37) to calculate the sensitivity of the function to the model parameters.

$$\Gamma = \frac{\partial \gamma}{\partial x}\Pi + \frac{\partial \gamma}{\partial u}\Theta \tag{3.37}$$

Here, similar to Π and Θ, if γ is a function of the states and control, Γ maps deviation in the parameters to deviation in the value of the function. As an example, Equation (3.37) was applied to the steer rate constraint. To summarize the process, the necessary partials of the steer rate equation are taken, and then inserted into Equation (3.37). The necessary partials are shown in Equations (3.38-3.39).

$$\frac{\partial \dot{\delta}}{\partial x} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & -a_\delta \end{bmatrix} \tag{3.38}$$

$$\frac{\partial \dot{\delta}}{\partial u} = a_\delta \tag{3.39}$$

Then, the resulting steer rate sensitivity is calculated using Equation 3.40.

$$\Gamma_{\dot{\delta}} = -a_\delta \Pi + a_\delta \Theta \tag{3.40}$$

Once again, the ellipsoidal error bounds on the function are created as shown in Equations (3.41-3.42).

$$P_{\dot{\delta}} = \Gamma_{\dot{\delta}} W \Gamma_{\dot{\delta}}^T \tag{3.41}$$

$$\rho_{\dot{\delta}} = \sqrt{P_{\dot{\delta}}} \tag{3.42}$$

In this case, $P_{\dot{\delta}}$ is a scalar, since the steer rate equation is only one dimensional. The radius of the error bound which is used to tighten the constraint is $\rho_{\dot{\delta}}$. Finally, the tightened steer rate constraint is calculated as shown in Equation (3.43).

$$\dot{\delta}_{min} + \rho_{\dot{\delta}} \leq \dot{\delta} \leq \dot{\delta}_{max} - \rho_{\dot{\delta}} \tag{3.43}$$

This process can be repeated for any constraint which is a combination of the state and control variables. In this thesis, this method was applied to calculate how much to modify each constraint at each time step in the horizon.

3.5 Implementation Procedure

At each update of the MPC, the QP is solved and produces a trajectory based on the initial states of the vehicle and the constraints throughout the horizon. If the constraint tightening method is active, then this first QP solution is used to calculate the error bounds around the nominal prediction, and the QP is solved again with these tightened bounds. These constraints may include lateral position constraints that require the tractor to be at a certain lateral position with respect to the lane centerline at a specified time in the horizon, They may also include constraints on the hitch angle of the trailer which can be related to constraints on the position of the entire vehicle. Throughout the maneuver, the trajectory produced by the solution of the QP is tracked by a feedback controller. The form of the feedback gains that was selected to be used in this thesis is shown in equation (3.44) and was chosen during tuning in the experimental testing.

$$K = \begin{bmatrix} 0 & k_e & k_{\psi_{rel}} & 0 & k_{v_y} & k_{\dot{\psi}} & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.44}$$

The entire procedure is summarized below in Algorithm 1. The feedback controller linearly interpolates the state trajectory solution produced by the QP to obtain $x^*(t)$. The control solution is interpolated based on a zero order hold assumption, to obtain $u^*(t)$. The control solution, $u^*(t)$ is combined with the feedback law to produce the total control command, u, at each update of the feedback controller. This step is shown in line 7 of Algorithm 1. Note that the feedback control law can be calculated at a faster rate than the QP solution. It may also be possible to reduce the entire algorithm to one QP solve by using the previous solution to calculate the tightened bounds. However, this was not tested in this thesis.

Algorithm 1 Closed Loop MPC Procedure

while Running **do**
  x^*, u^* ← SolveQP(x_0, ConstraintBounds)
  if UseConstraintTightening **then**
    ConstraintBounds ← CalculateTightenedBounds(x^*, u^*, ConstraintBounds)
    x^*, u^* ← SolveQP(x_0, ConstraintBounds)
  end if
  u = u*(t) + K(x-x*(t))
end while

Figure 3.3 shows a flowchart of the system as it may be implemented within an autonomy system. However, in some simulations in this thesis the MPC replanning rate is executed at a different rate than 10 Hz which is noted in each section of the thesis where this occurs.

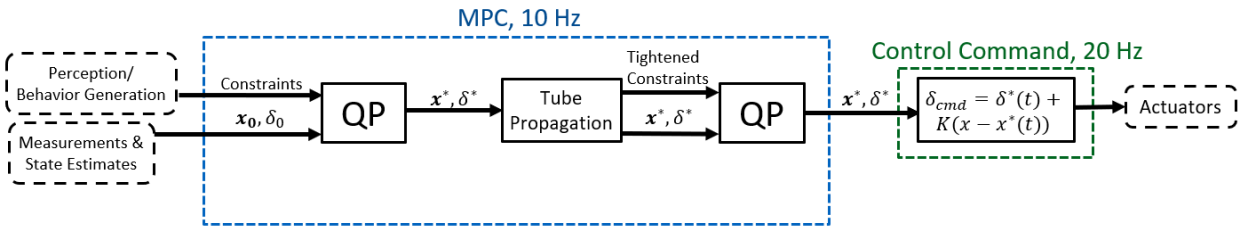

Figure 3.3: System Flowchart

Although alternative methods may be able to produce a path that satisfies position constraints on the tractor and trailer based on geometry alone and assuming perfect path tracking, the procedure given in this section aims to ensure that all constraints will be satisfied even when the vehicle cannot track the path perfectly due to parameter uncertainty.

Chapter 4

Constraint Satisfaction Simulation Results

4.1 State Constraints

4.1.1 Open Loop Validation

First, to validate both the MPC design and the sensitivity-based error tube propagation, a 500 run Monte Carlo validation was performed on the open loop MPC prediction. This was done to validate the error tube propagation by comparing the spread of the tracked trajectories to the bounds predicted by the sensitivity based tube. In this test, the MPC solution was only calculated once at the beginning of the maneuver, but the solution was tracked by the feedback controller. This was done to match the assumptions used by the constraint tightening method.

The simulation environment was developed in MATLAB. For all simulation work, the automatic differentiation library CasADi with its interface to the nonlinear programming solver IPOPT was used to solve the optimization problem [38, 39]. The nonlinear lateral dynamic model given previously by Equation (2.8) was used as the vehicle model in this validation. The nominal vehicle parameters used are shown in Table 4.1. The parameter deviations were sampled from a Gaussian distribution at the beginning of each run with the standard deviation also shown in Table 4.1. In the tube propagation, the maximum parameter deviations were considered to be the 5σ value.

Table 4.1: Parameters and Deviation for Monte Carlo Analysis

Parameter	Nominal Value	1 σ Deviation	Units
$C_{\alpha,1}$	2.58e5	1e5	N/rad
$C_{\alpha,2}$	1.68e5	1e5	N/rad
$C_{\alpha,3}$	1.68e5	1e5	N/rad
$C_{\alpha,4}$	1.17e5	1e5	N/rad
$C_{\alpha,5}$	1.17e5	1e5	N/rad
m_1	6,493	0	kg
m_2	8,196	1000	kg
J_1	19,665	0	$kg \times m^2$
J_2	204,104	10000	$kg \times m^2$
a	1.384	0	m
b_1	3.616	0	m
b_2	4.886	0	m
c	4.251	0	m
d	7.0	0.5	m
f_1	12.308	0.2	m
f_2	13.596	0.2	m
a_δ	10.0	0.5	sec^{-1}
κ	0	$5e$-5	m^{-1}

The weighting for Q shown below in Equation (4.1) was chosen for the simulations based on experimental hand tuning.

$$Q = diag(\begin{bmatrix} 0 & 100 & 1e4 & 0 & 100 & 0.01 & 0 & 100 & 0 & 0 \end{bmatrix}) \tag{4.1}$$

Additional weights and parameters for the MPC and feedback controller, including the discretization size dt, and the length of the time horizon t_f, are given in Table 4.2, where $|\dot{\delta}|_{max}$ is specified in degrees per second at the ground wheel.

To simulate what may be required for an avoidance maneuver, constraints on the states were applied for this test, where a box shaped region was chosen to be avoided during a period of time in the horizon. This maneuver was chosen to demonstrate a scenario in which constraint tightening is likely to improve constraint satisfaction. This is because for an avoidance maneuver that is not a full lane change, the reference conflicts with the constraints. The desired lateral position is zero meters from the lane centerline, but the constraint forces the MPC to plan a trajectory that moves

Table 4.2: Controller Parameters for MC Analysis

Parameter	Value	Units
k_e	0.1	-
k_ψ	1.0	-
k_{v_y}	0.0	-
$k_{\dot{\psi}}$	0.0	-
$w_{\dot{\delta}}$	1.5e5	-
R	1e3	-
$\lvert\dot{\delta}\rvert_{max}$	45	deg/sec
dt	0.1	sec
t_f	5.0	sec

away from the reference. In these situations, the MPC is likely to plan a trajectory that is close to the constraint boundary.

Speed was maintained at 70 mph throughout the maneuver. Heading of the vehicle was aligned with respect to the path at the beginning of the maneuver to ensure feasibility of the QP at the initial point. In addition to the lateral position constraints, a maximum hitch angle of two degrees was applied to demonstrate the capability of positional constraints on the trailer. Although the precise hitch angle needed to avoid the trailer entering the obstacle region was not calculated for this maneuver, two degrees was chosen as this produces approximately 0.5 meters of lateral offset at the rear of a standard 53 foot trailer.

Figure 4.1 shows the lateral position over time for nominal MPC where the constraint tightening method was not used. The red shaded area shows the obstacle region, and the solid orange line is the nominal predicted trajectory. Figure 4.2 shows the hitch angle for this case, where the vertical green dashed lines show the beginning and end of the time that the constraint is active, and the red dashed line is the constraint boundary. In each plot a solid orange line shows the initial optimal solution produced by the MPC. It can be seen that, when regular MPC is used, the solution lies as close as possible to the constraint in some instances. In the open loop case, without constraint tightening, the hitch angle constraint was violated in 99.6 percent of runs while the lateral position constraint was violated in 8.0 percent of runs.

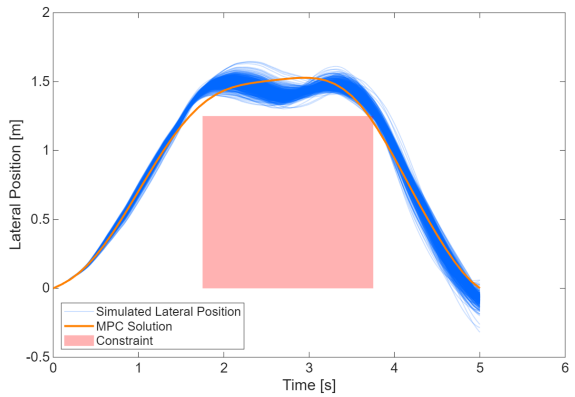

Figure 4.1: Lateral Position, Regular Constraints

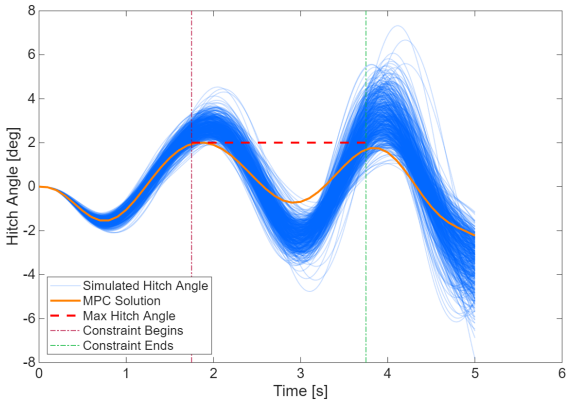

Figure 4.2: Hitch Angle, Regular Constraints

Next, tightened constraints were applied, and the 500 run Monte Carlo was performed. Figure 4.3 shows the lateral position for the constraint tightened case, where Figure 4.4 shows the hitch angle for the constraint tightened case. In the constraint tightened case, the lateral position constraint was not violated in any runs, while the hitch angle constraint was violated in 0.4 percent of cases.

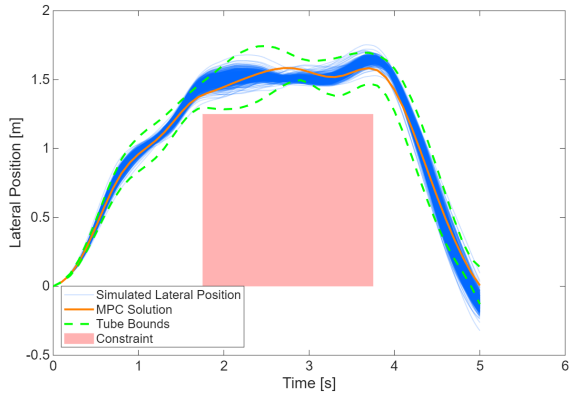

Figure 4.3: Lateral Position, Tightened

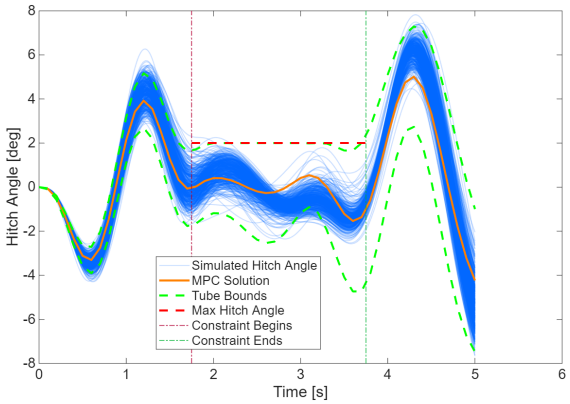

Figure 4.4: Hitch Angle, Tightened

When the MPC is executed only once and the feedback controller alone is active throughout the trajectory, the assumptions made by the tube propagation are matched and the coverage of the error tubes compared to the real spread of trajectories can be analyzed. Figures 4.5-4.6 show the comparison of the Monte Carlo statistics to the MPC's nominal prediction and tube boundaries for the lateral position state, while Figures 4.7-4.8 show the comparison of Monte Carlo statistics

against the MPC's prediction and tube boundaries for the hitch angle state. Table 4.3 summarizes

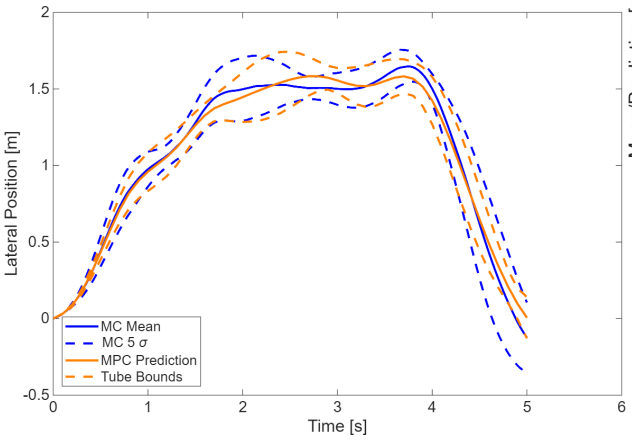

Figure 4.5: Tube Comparison, Lateral Position

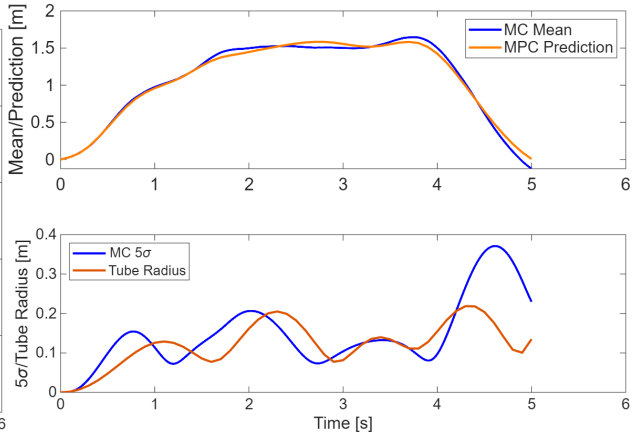

Figure 4.6: Monte Carlo Statistics vs Prediction, Lateral Position

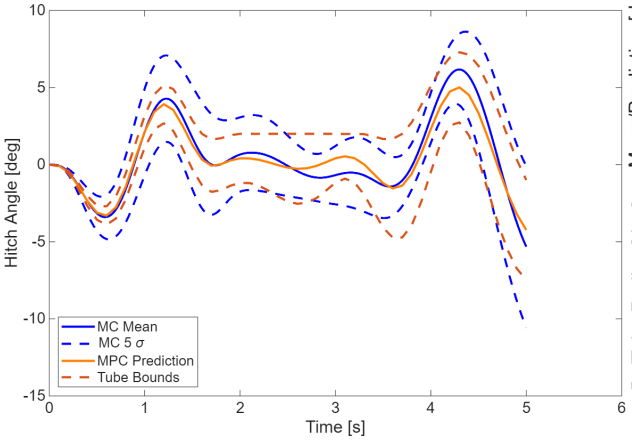

Figure 4.7: Tube Comparison, Hitch Angle

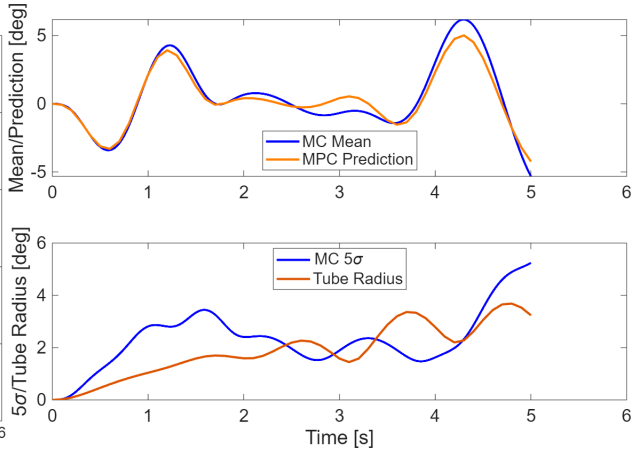

Figure 4.8: Monte Carlo Statistics vs Prediction, Hitch Angle

the differences in the Monte Carlo results and the predictions from the MPC and tube propagation in terms of maximum absolute error and RMSE. The tube radius over and under estimates the uncertainty in different parts of the trajectory. However, it provides enough of a buffer to improve constraint satisfaction. This difference is partially explained by the fact that the tube propagation method used in this thesis is a first-order approximation of how parameter error affects the system. Higher order methods could be explored in future work as a way to improve the accuracy of the tube propagation.

State	Comparison	Maximum Absolute Error	RMSE	Units
Lateral Position	MC Mean vs Prediction	12.89	5.14	cm
Lateral Position	MC 5σ vs Tube Radius	22.29	7.47	cm
Hitch Angle	MC Mean vs Prediction	6.0839	0.0514	deg
Hitch Angle	MC 5σ vs Tube Radius	5.1855	2.5539	deg

Table 4.3: Summary of Monte Carlo Results

4.1.2 Closed Loop Validation

The 500 run Monte Carlo analysis is now performed with the MPC running in closed loop, and two different update rates were compared. It was found that with higher update rates of the MPC, constraint tightening becomes somewhat less of a necessity as the entire control scheme becomes inherently more robust. However, even at rates of 2 Hz the robust method is necessary, and at 10 Hz some improvement in constraint satisfaction is still seen. Note that the vehicle is controlled at 20 Hz by the feedback controller in all cases, so the MPC update rate mentioned here is the rate at which re-planning occurs. In closed loop, the initial predicted trajectory and error tubes are replaced with updated plans as the MPC calculates new solutions. These are therefore omitted in the rest of the figures.

Figure 4.9 shows the lateral position and Figure 4.10 shows the hitch angle for the 2 Hz case of regular MPC. In this case, the lateral position constraint was violated in 99.0 percent of the runs, although this is in part due to discretization error, and the magnitude of the constraint violation is very small and for a short duration. Additionally, the hitch angle constraint was violated in 65.4 percent of cases.

Figure 4.11 shows the lateral position and Figure 4.12 shows the hitch angle for the 2 Hz case of constraint tightened MPC. In this case, the lateral position constraint was never violated in any runs while the hitch angle constraint was violated in 0.2 percent of runs.

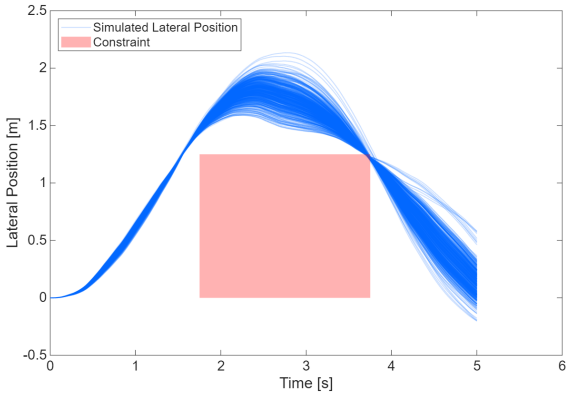

Figure 4.9: Lateral Position, 2 Hz, Regular Constraints

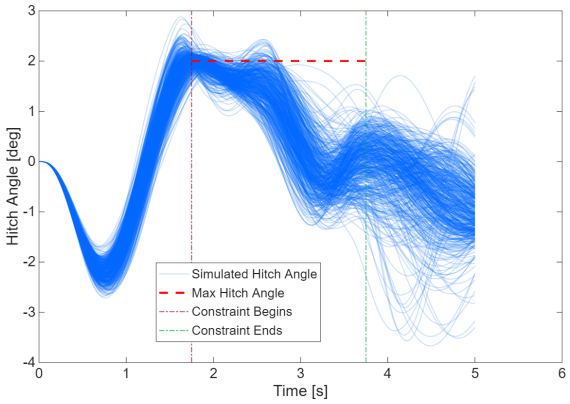

Figure 4.10: Hitch Angle, 2 Hz, Regular Constraints

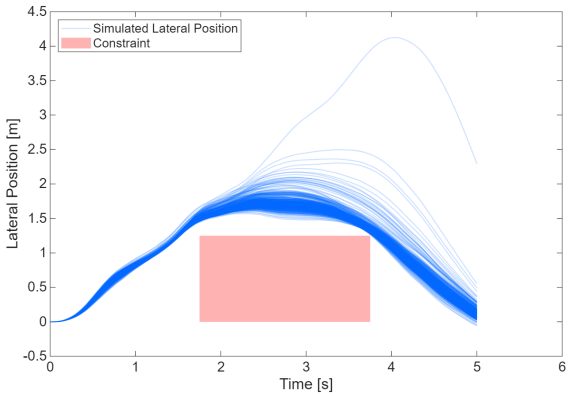

Figure 4.11: Lateral Position, 2 Hz, Tightened

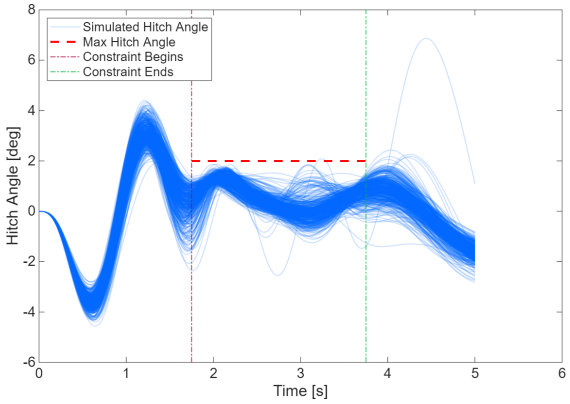

Figure 4.12: Hitch Angle, 2 Hz, Tightened

The Monte Carlo runs were repeated with the MPC running at 10 Hz. Figure 4.13 shows the lateral position and Figure 4.14 shows the hitch angle for the 10 Hz case of without constraint tightening. The lateral position constraint was violated in 95.8 percent of runs, but once again this can be attributed partially to discretization error due to the fact that the constraint is only violated when the time until the constraint is deactivated is less than the discretization used in the MPC. The hitch angle constraint was violated in 46.6 percent of cases. Note that there is significantly less hitch angle violation compared to the 2 Hz update rate.

Figure 4.15 shows the lateral position and Figure 4.16 shows the hitch angle for the 10 Hz case with constraint tightening. Unsurprisingly, this case demonstrates the best performance with zero violations of the lateral error or hitch angle constraints.

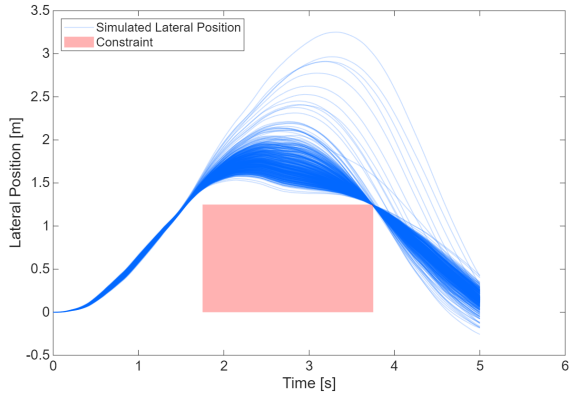

Figure 4.13: Lateral Position, 10 Hz, Regular Constraints

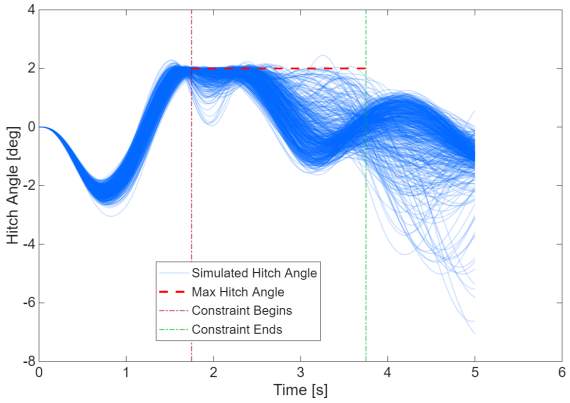

Figure 4.14: Hitch Angle, 10 Hz, Regular Constraints

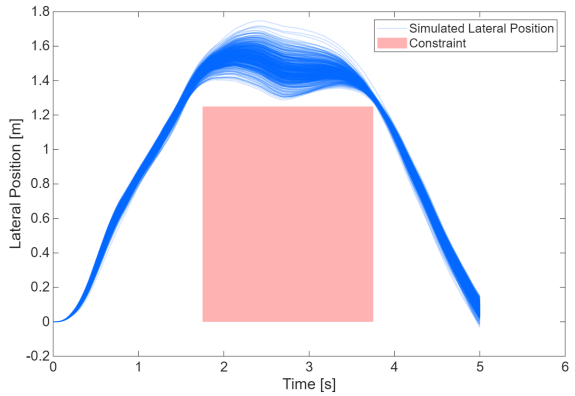

Figure 4.15: Lateral Position, 10 Hz, Tightened

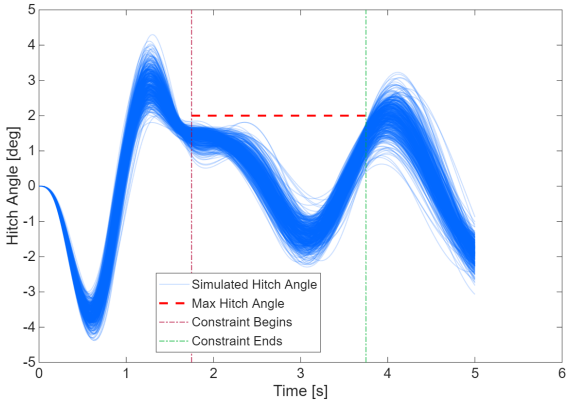

Figure 4.16: Hitch Angle, 10 Hz, Tightened

The results show that improved robustness can be achieved either by increasing the update rate of the MPC or by applying constraint tightening. However, in cases where the update rate of the controller is limited by computing power, applying constraint tightening is an efficient way to ensure robustness. Furthermore, even when the MPC is run at 10 Hz, constraint tightening eliminated constraint violations. It is possible that at a high enough update rate the regular MPC is robust enough without additional constraint tightening.

4.1.3 High Fidelity Simulation

An additional simulation environment was developed in MATLAB that interfaced with a vehicle dynamics model provided by TruckSim. TruckSim is a professional simulation software that

provides a high fidelity vehicle dynamics model and allows the user to specify parameters of the vehicle and environment [11]. The MATLAB environment was created to modify configuration files of the TruckSim software to vary the vehicle parameters between runs, so that Monte Carlo analysis could be performed. The parameters were varied as closely as possible to what was shown in the previous section. However, the friction parameter of the road, μ, was varied with a 1σ of 0.05 around a mean value of 0.8 instead of varying tire cornering stiffness as done in the previous simulations.

Additionally, zero mean Gaussian noise was added to the states that were fed into the MPC and the feedback controller throughout all simulation runs. Note that measurements of heading relative to a local frame and along track position are not directly needed, and the current steer angle is assumed to be measured accurately. The standard deviation of the measurement noise added to each state is shown in Table 4.4.

Table 4.4: Measurement Noise in Simulation

Measurement	$1\ \sigma$	Units
Lateral Position (e)	5	cm
Relative Heading (ψ_{rel})	0.2	Degrees
Lateral Velocity (v_y)	0.01	m/s
Yaw Rate ($\dot{\psi}$)	0.001	deg/s
Hitch Rate ($\dot{\gamma}$)	0.002	deg/s
Hitch Angle (γ)	0.4	Degrees

A 100 run analysis was first completed without constraint tightening active, and the MPC running at 2 Hz. The lateral position constraint was violated in 53 percent of runs, while the hitch angle constraint was violated in 12 percent of the runs. Figure 4.17 shows the lateral position over time for the regular MPC case, while Figure 4.18 shows the hitch angle. Figures 4.19 and 4.20 show the steer angle and lateral acceleration respectively throughout the maneuver for the regular MPC case.

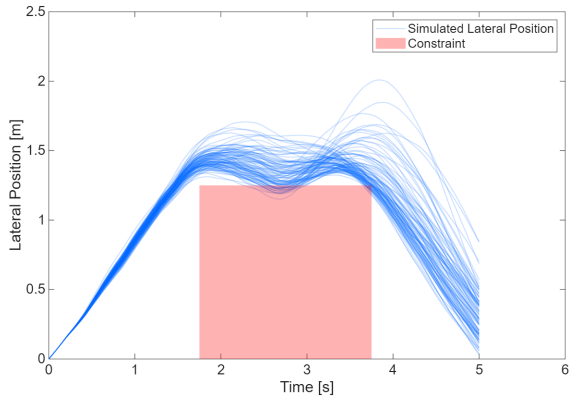

Figure 4.17: Lateral Position, Regular Constraints, TruckSim

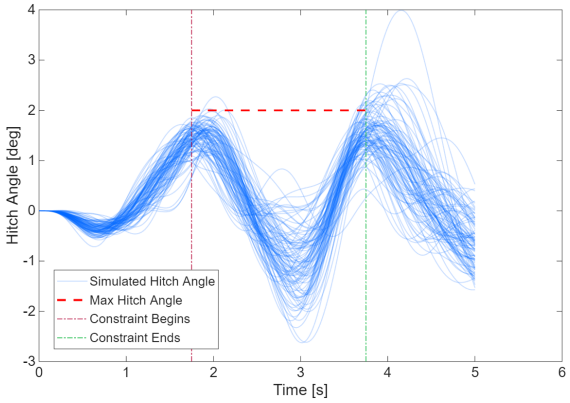

Figure 4.18: Hitch Angle, Regular Constraints, TruckSim

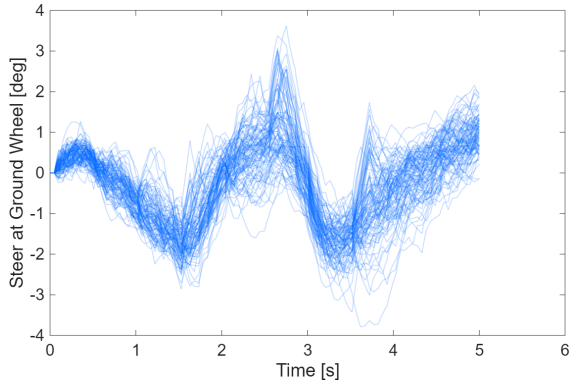

Figure 4.19: Steer Angle, Regular Constraints, TruckSim

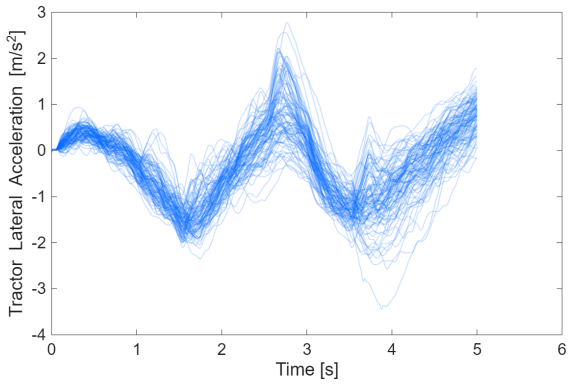

Figure 4.20: Lateral Acceleration, Regular Constraints, TruckSim

The simulation was repeated with constraint tightening active. With constraint tightening active, the lateral position constraint was violated in zero runs, while the hitch angle was violated in 2 percent of the runs (2 runs). Figure 4.21 shows the lateral position for the constraint tightened case, and 4.22 shows the hitch angle. Figures 4.23 and 4.24 again show the spread of steer angles and lateral accelerations respectively throughout the maneuver for the constraint tightened case.

The constraint tightened cases show a major improvement in robustness compared to regular MPC. However, they also produce noticeably more aggressive changes in steer angle and lateral accelerations. With tighter constraints applied, more aggressive changes in control input are required to ensure the tighter constraints are always met. This also demonstrates a tradeoff between conservativeness with respect to constraints and other performance metrics such as ride comfort.

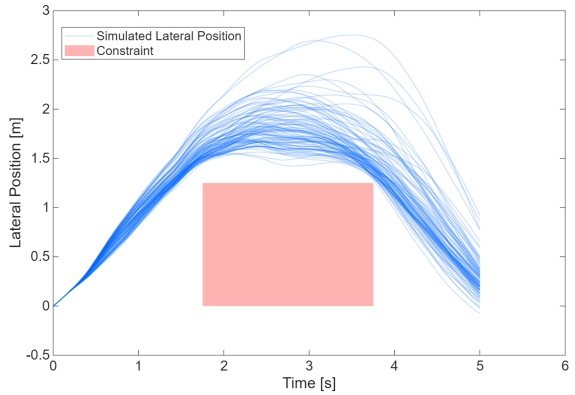

Figure 4.21: Lateral Position, Tightened Constraints, TruckSim

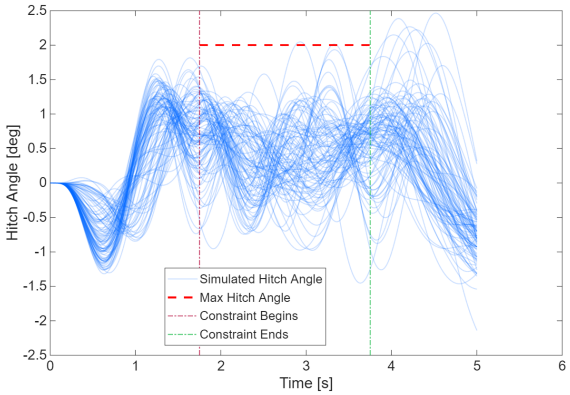

Figure 4.22: Hitch Angle, Tightened Constraints, TruckSim

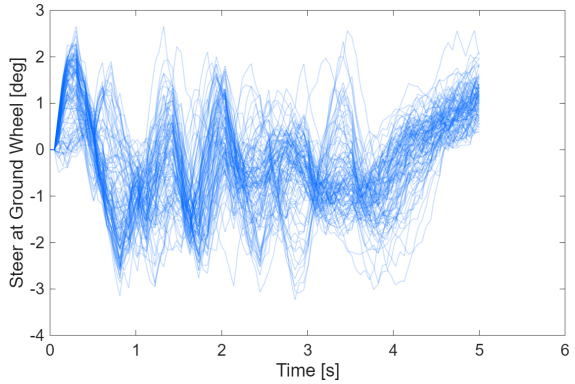

Figure 4.23: Steer Angle, Tightened Constraints, TruckSim

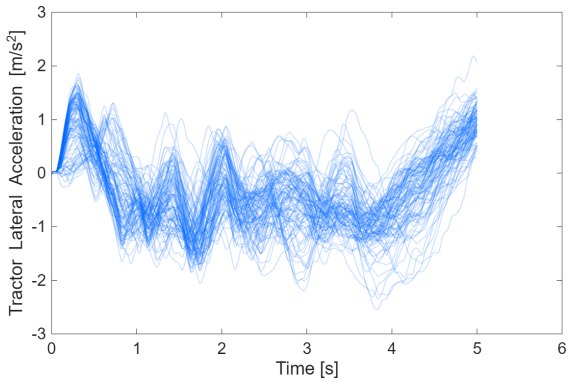

Figure 4.24: Lateral Acceleration, Tightened Constraints, TruckSim

Although use of constraint tightening prevents the vehicle from violating hard constraints, this can come at a cost of aggressive control action. This also demonstrates the need for error tubes that are not overly conservative. The smallest possible tube that covers the spread of possible trajectories is desired, so that performance is not sacrificed. For this reason, the sensitivity based tube propagation can fit this task well. Additionally, while the measurement noise was not explicitly accounted for, in practice the constraint tightening method still works well. It should also be noted, that the constraint tightening method demonstrated cannot show an improvement in all cases. In cases where the original problem is infeasible or very close to being infeasible, constraint tightening will only serve to make the problem more infeasible. This method instead shows benefit

in situations where the optimal solution would otherwise lie close to the constraint boundary, but a safety buffer is desired.

The error tube then serves as a way to choose the necessary safety buffer based on physically meaningful quantities that represent the uncertainty on the prediction. This process differs from just adding a constant size safety buffer to each constraint, or from increasing the size of the obstacle region, however. This method instead adds a buffer to constraints separately at each time in the horizon based on how much the prediction will be affected by the parameter uncertainty. For example, the constraints will be tightened more during aggressive lateral maneuvers where the accuracy of the prediction is more affected by model parameters such as the trailer's yaw moment of inertia. Furthermore, the constraints should be tightened most only during points in the horizon where the accuracy of the prediction is most affected by parameter error. The advantage of this method is that conservatism should be added to the constraints only where it is needed.

4.2 Trailer Position Constraints

The previous sections demonstrated constraints on the tractor's lateral position and the trailer hitch angle. However, with some assumptions, the hitch angle constraint can be changed to a trailer position constraint instead to more accurately model an obstacle avoidance scenario. Figure 4.25 illustrates the avoidance scenario that the trailer position constraint intends to capture.

The lateral position of the midpoint of the trailer bumper with respect to the reference line on a straight road can be described by Equation (4.2).

$$e_{tr,bumper} = e - c\sin(\psi_{rel}) - L\sin(\psi_{rel} + \gamma) \tag{4.2}$$

Recall that e is the lateral position of the tractor's center of mass, c is the distance between tractor CG and hitch point, and L is the length of the trailer from hitch to rear bumper. A linearized version of this constraint is applied, in which a small angle assumption is applied to the hitch angle, so that

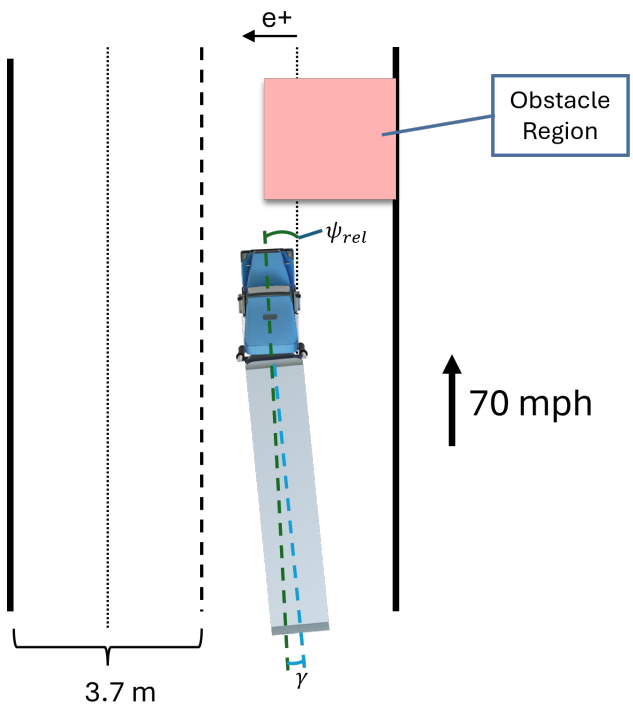

Figure 4.25: Obstacle Avoidance Scenario

$sin(\gamma) \approx \gamma$. The linearized relationship is given by Equation (4.3).

$$e_{tr,bumper} = e - (c + L)\psi_{rel} - L\gamma \tag{4.3}$$

For this simulation study, the linear constraints on hitch angle were replaced with a constraint on the midpoint of the trailer bumper as defined by Equation (4.3).

4.2.1 Open Loop Validation

The tube propagation was again validated by a 500 run Monte Carlo simulation in which the feedback controller tracks the first MPC prediction. The nonlinear vehicle dynamic model given by Equation (2.8) was again used for the simulation. The trailer position constraint is applied starting at 2.1 seconds into the horizon and ending at 3.65 seconds. The lateral trailer position and tractor position during this time are constrained to be greater than or equal to 1.25 meters with respect to the road reference line. These values are chosen because they provide a feasible scenario while still

requiring some significant input to move the vehicle safely out of the way. Furthermore, a greater than constraint forces the solution to lie to the left of the obstacle. Alternatively, a constraint forcing the path to lie to the right of the obstacle could be chosen. This is a choice that is assumed to be made before or during the planning step. While not explicitly handled in this thesis, this decision is a simple binary choice, and could be handled by solving the QP with either constraint active and using the solution with lower cost. Solving the QP an additional time would also add to the computation time however.

Note that the constraints are applied to the midpoint of the trailer bumper and to the center of mass of the tractor respectively. To fully avoid the obstacle in this work, half the trailer width is considered to be included in the constraint value. All other simulation parameters and controller settings are identical to those in Sections 4.1.1 and 4.1.2. The additional parameter in Equations (4.2) and (4.3) is the trailer length, L, which was chosen to be the same as the hitch point to rear axle distance, f_2, for the purpose of this simulation.

Figures 4.26 and 4.27 show the tractor CG position and trailer bumper position respectively without constraint tightening applied. The orange line shows the MPC prediction, and the blue lines show the result from each simulation run. The linearized relationship for trailer position, Equation (4.3), is used for the constraint in the MPC, while the simulation result for trailer bumper position was calculated using the nonlinear relationship given by Equation (4.2). Here, the tractor position constraint was violated in 100 percent of cases, and the trailer constraint was violated in 87.4 percent of cases. The large number of violations indicates that the deviation from the predicted trajectory is not only caused by parameter uncertainty.

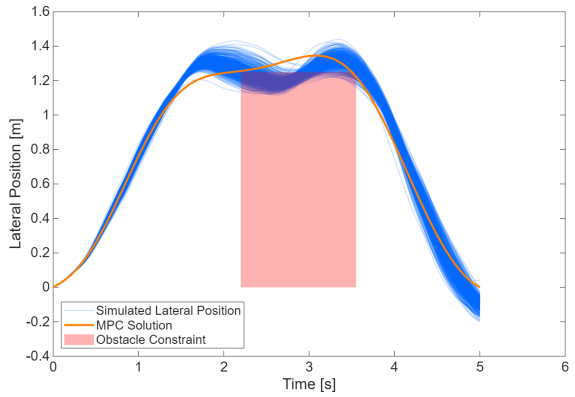

Figure 4.26: Tractor Lateral Position, Regular Constraints

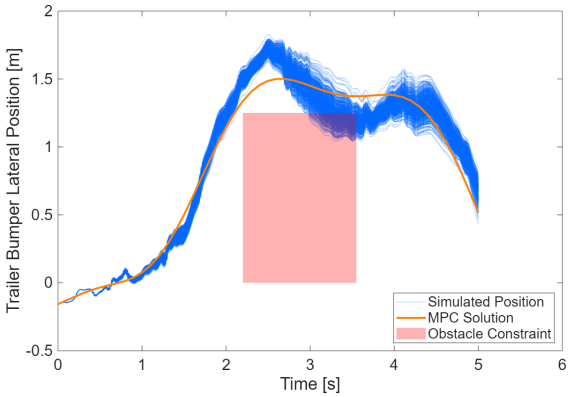

Figure 4.27: Trailer Bumper Position, Regular Constraints

Figures 4.28 and 4.29 show the tractor CG position and trailer bumper position respectively with tightened constraints applied. The tractor position constraint was violated in zero cases and the trailer position constraint was violated in 0.6% of the runs.

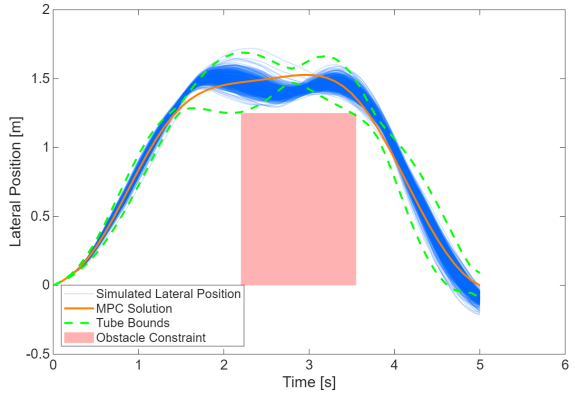

Figure 4.28: Tractor Lateral Position, Tightened Constraints

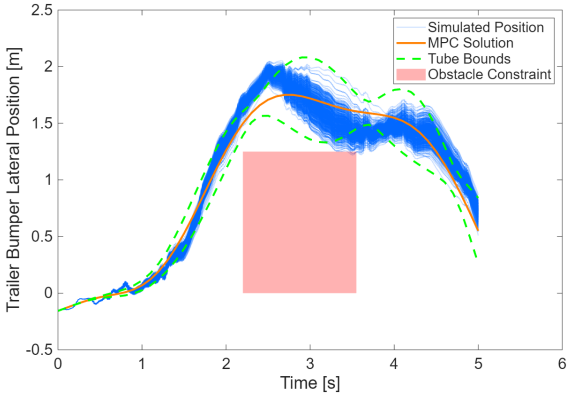

Figure 4.29: Trailer Bumper Position, Tightened Constraints

4.2.2 Closed Loop Validation

The analysis was repeated with the MPC executing at 10 Hz. First, the simulation was performed without constraint tightening active. In this case, the tractor constraint was violated in 96.7 percent of runs. However, as in Section 4.1, these violations only occurred between the last sample the constraint was enforced and the next sample at which the MPC replanned. Because of the discretization used in the MPC, the constraint was satisfied at every point in the horizon, but

was violated in the simulation only in between MPC executions. Disregarding these violations, which were only due to the MPC discretization, the tractor position constraint was satisfied in all runs. The trailer position constraint was violated in only a single simulation run. As expected, the feedback introduced by updating the MPC at 10 Hz improves robustness significantly.

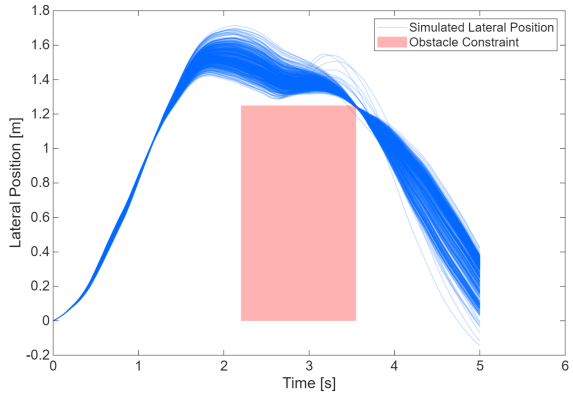

Figure 4.30: Lateral Position, Regular Constraints, 10 Hz

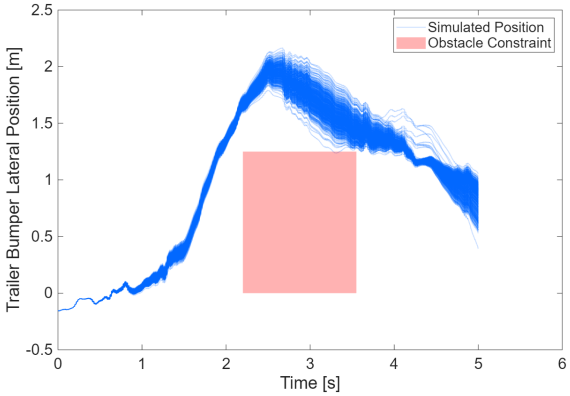

Figure 4.31: Hitch Angle, Regular Constraints, 10 Hz

The simulation was run again with the constraint tightening method active. In this case, the tractor position constraint was violated in a single run and the trailer position was violated in zero runs. Although increasing the MPC feedback improves robustness significantly, using the tightened constraints adds a reasonable safety buffer to the initial planned trajectory that improves constraint satisfaction even when sources of error other than parameter uncertainty, such as may cause the constraints to be violated.

Figure 4.34 shows a visualization of the tractor and trailer during a successful maneuver, with zoomed regions showing areas where the vehicle is close to the obstacle which were created using the ZoomPlot package in MATLAB[40]. The approximate bounding regions of the tractor and trailer are represented by the green and blue rectangles whose dimensions are set to approximately match the length and width of a Peterbilt 579 and 53 foot van trailer. The spatial constraint that corresponds to the time based constraints on tractor and trailer position is marked by the red region, which occupies almost half of the lane's width for a short duration. This region was calculated

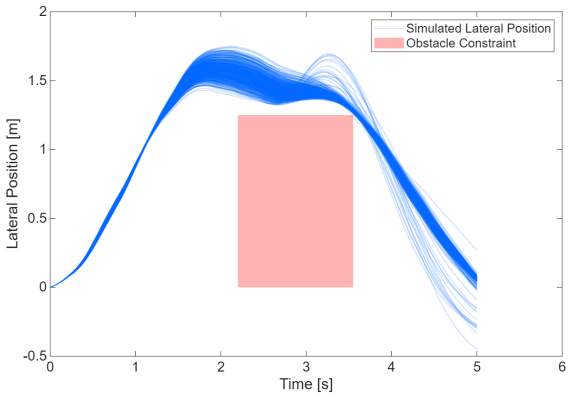

Figure 4.32: Lateral Position, Tightened Constraints, 10 Hz

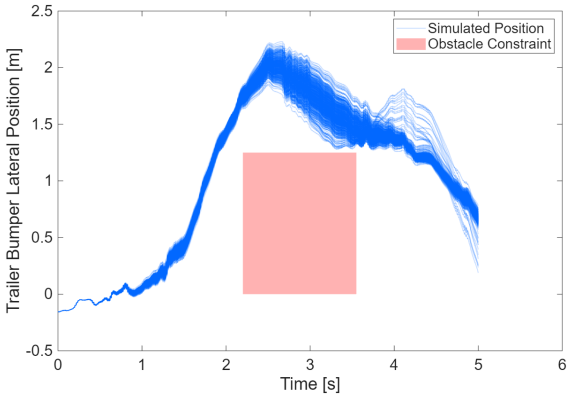

Figure 4.33: Trailer Position, Tightened Constraints, 10 Hz

using the relationships shown in Equations (4.4-4.5).

$$x_{obs,min} = x_0 + t_1 v \tag{4.4}$$

$$x_{obs,max} = x_0 + t_2 v - (c + L) \tag{4.5}$$

In these equations, t_1 and t_2 mark the times between which both constraints are active, c is the distance from the tractor CG to the hitch point, and L is the trailer length. Note that this is an approximate conversion from the time based constraints to their spatial equivalents. Additionally, the time based constraints are a conservative approximation of the obstacle region as the tractor constraint remains active longer than necessary. However, the vehicle can be constrained to avoid a box shaped spatial region using the time based constraints that are demonstrated in this section. The trailer constraints are also placed at the midpoint of the vehicle's bumper. Therefore, the width of the vehicle is accounted for when calculating the constraint value using the maximum van trailer width of 8.5 ft ($\approx$ 2.6 m).

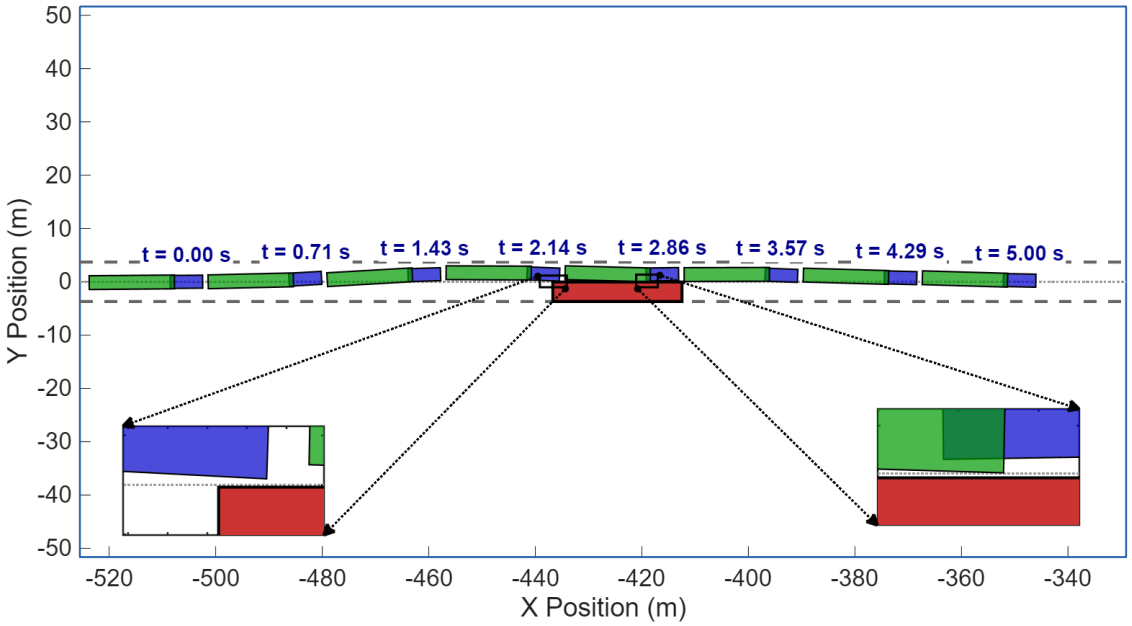

Figure 4.34: Obstacle Avoidance Visualization

4.2.3 Comparison to Nonlinear Constraint

To validate the accuracy of the linearized trailer position constraint, the trailer position was calculated throughout the MPC's planned trajectory using both the nonlinear relationship from Equation (4.2) and the linearized relationship from Equation (4.3). Figure 4.35 shows the direct comparison of the nonlinear and linearized trailer rear bumper position calculations, which are nearly identical.

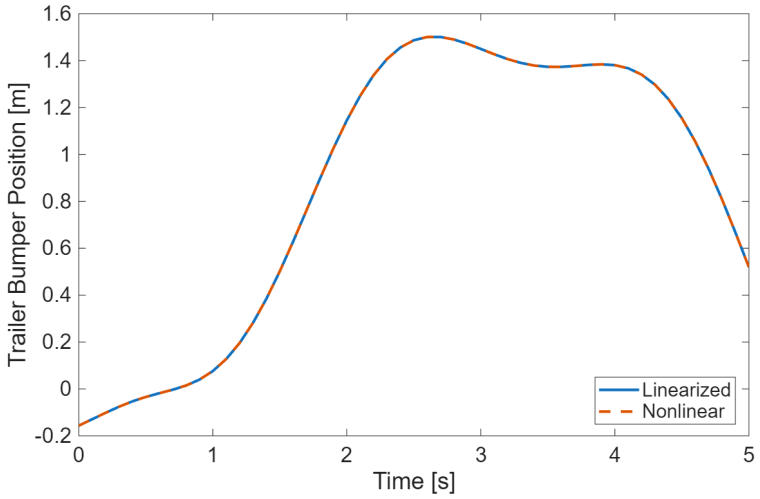

Figure 4.35: Linearized vs Nonlinear Trailer Position

Figure 4.36 shows the error on the linearized trailer bumper position, which is less than 0.5 millimeters throughout the horizon.

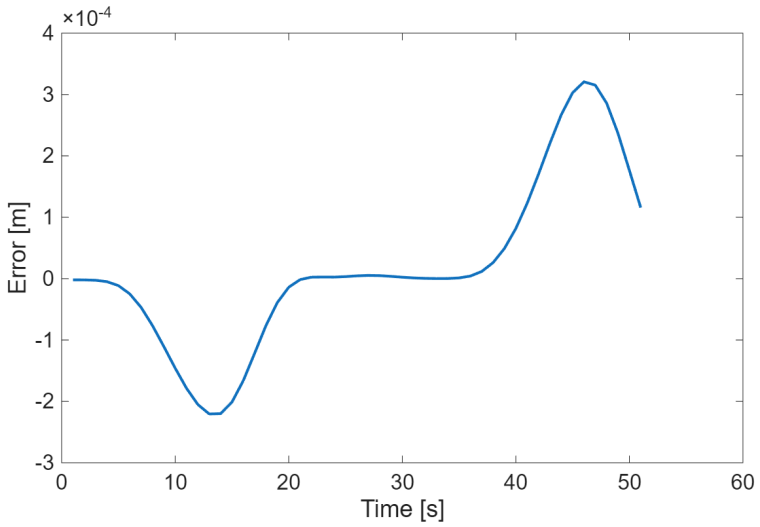

Figure 4.36: Trailer Position Approximation Error

Due to the accuracy of the linearization of $\sin$ at the values of heading and hitch angle in this range, the error on the linearized constraint is negligible.

Chapter 5

Lane Keeping Experimental Results

5.1 Experimental Setup

Experimental validation of the system was performed on a GPS and Vehicle Dynamics Lab (GAVLab) test vehicle. This experimentation was performed to demonstrate the capability of the MPC design for automated control of a class 8 tractor trailer vehicle in an environment that is similar to on-road driving. The vehicle used in this experimentation was a Peterbilt 579 with a 53 foot van trailer attached referred to as A2, and is shown in Figure 5.1.

Figure 5.1: Peterbilt 579 (A2)

Both the cab and trailer were outfitted with a Honeywell eTalin, shown in Figure 5.2. The eTalin is a drop-in navigation solution that can provide a precise estimate of attitude and body-frame velocities of a vehicle [41]. The eTalin was used as the source of heading estimates with respect to the North for both the tractor and the trailer. The trailer and tractor heading estimates from the eTalin were differenced to obtain the hitch angle. The eTalin was also used as the source of yaw rate for both the tractor and trailer, and these were differenced to obtain hitch rate. Although the eTalin can provide a global position from GPS as well, in this work a Novatel FlexPak6, shown in Figure 5.3, was used with corrections from a Real-Time Kinematic (RTK) base station to provide precise positioning [42].

Figure 5.2: Honeywell eTalin [41]

Figure 5.3: Novatel FlexPak6 [42]

The vehicle is also equipped with a ZF steering motor at the steering column, which receives a commanded steer angle over the vehicle's Controller Area Network (CAN) bus. An onLogic computer, running Ubuntu 18 and Robot Operating System (ROS), is used to receive all of the information from the sensors and the vehicle's CAN bus, run the autonomy software, and send actuation commands over CAN [43]. The vehicle is also equipped with a CAN gateway device used to enable or disable actuation and to interface with a physical emergency stop button. The architecture of the system is shown in Figure 5.4.

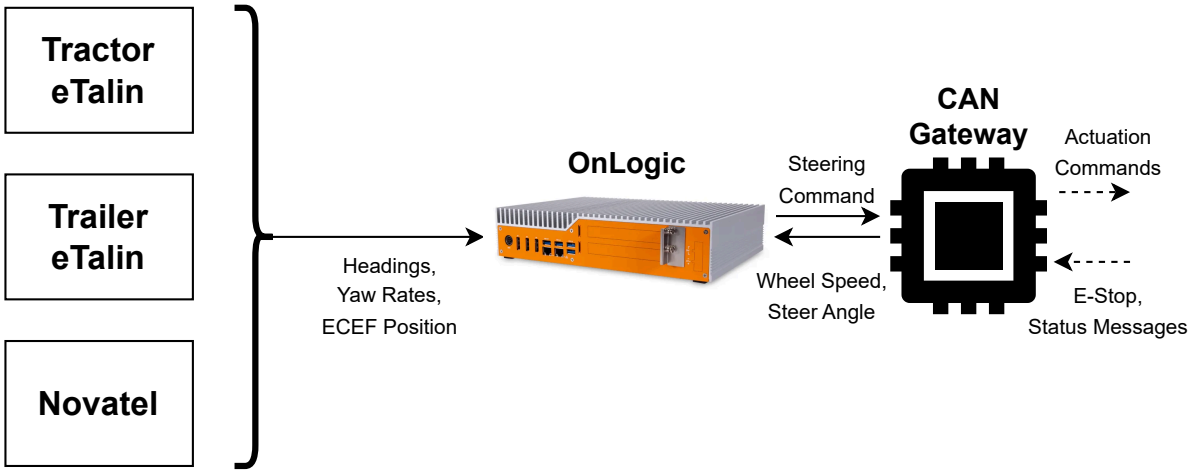

Figure 5.4: Test Vehicle Configuration

5.2 Real-Time Implementation

The MPC design was implemented in C++, and the Operator Splitting QP (OSQP) solver was used to solve the QP [36]. The osqp-cpp wrapper library was used to interface with the OSQP solver in C++ [44]. OSQP was chosen based on its efficiency compared to other available solvers, and its capability for real-time or embedded applications [36]. For the real-time implementation, rather than using automatic differentiation software such as CasADi, the analytical gradients and hessian of the cost and constraints derived in Chapter 3 were used to form the QP which was then passed to the solver. Automatic differentiation was not used for the real-time implementation, since faster run-times can be achieved by using the analytically derived gradients and hessian.

The C++ class used to construct the QP and call the solver was integrated into a ROS node, which was used to receive the sensor feedback and send actuation commands. The ROS node consists of two threads, one of which was used to calculate the MPC solution at approximately 10 Hz, and the other was used to interpolate the latest available MPC solution and calculate the additional feedback command at 20 Hz, as described in Algorithm 1. The C++ Lanelet2 library was used to manage the map containing the lane centerlines and boundaries. This library was used to load a previously surveyed map of the lane centerlines and boundaries in the OpenStreetmap (OSM) format, and was also used to calculate curvatures of the lane online [45]. Upon receiving

a global position and heading solution, the ROS node calculated the relative position and relative heading of the vehicle with respect to the desired lane centerline selected by the user.

The weights and gains for the MPC and feedback controller were chosen through a process of experimental tuning. First, the feedback gains were set to zero while the performance of the MPC alone was tuned. The relative heading weight was set much higher than the lateral position weight, which had the effect of adding damping and reducing steering oscillation. Other weights were slowly increased until satisfactory performance was achieved, but it is possible that better performance could be achieved with a different combination of weights. Equation (5.1) shows the tuning of the Q matrix during experimental testing.

$$Q = diag\left(\begin{bmatrix} 0 & 5e2 & 1e4 & 0 & 1.0 & 10.0 & 0 & 10.0 & 0 & 0 \end{bmatrix}\right) \tag{5.1}$$

Equation (5.2) shows the tuning of R that was used in these tests.

$$R = 1e4 \tag{5.2}$$

The parameters of the truck and trailer used are given in Table 5.1. Trailer axle locations were measured, and a previously identified parameter for the steering actuator, a_δ, was used. However, all other parameters were duplicated from the TruckSim model of a Peterbilt 579. Additionally, the following discretization, horizon length, and feedback gains that were used in the experimental testing are shown in Table 5.2.

Table 5.1: Test Vehicle Parameters

Parameter	Nominal Value	Units
$C_{\alpha,1}$	2.58e5	N/rad
$C_{\alpha,2}$	1.68e5	N/rad
$C_{\alpha,3}$	1.68e5	N/rad
$C_{\alpha,4}$	1.16e5	N/rad
$C_{\alpha,5}$	1.16e5	N/rad
m_1	6,493	kg
m_2	3,196	kg
J_1	19,665	$kg \times m^2$
J_2	179,000	$kg \times m^2$
a	1.384	m
b_1	3.616	m
b_2	4.886	m
c	4.251	m
d	7.0	m
f_1	10.7	m
f_2	11.4	m
a_δ	12.0	sec^{-1}
Steering Ratio	18.2	-

Table 5.2: Controller Parameters for Experimental Testing

Parameter	Value	Units
k_e	0.02	-
$k_{\psi,rel}$	0.2	-
k_{v_y}	0.05	-
$k_{\dot{\psi}}$	0.05	-
$w_{\dot{\delta}}$	1e4	-
$\lvert\dot{\delta}\rvert_{max}$	400	deg/sec
dt	0.1	sec
t_f	3.5	sec

5.3 Results

Two tests of the real-time implementation are shown in this section, with differing settings for the lateral position constraints. The procedure given in Algorithm 1 was run throughout both tests. Although the constraint tightening technique has little or no noticeable impact on the performance of the MPC when run in closed loop in normal lane keeping operation, the constraint tightening was kept active throughout both tests to demonstrate the capability of solving both QP's per update of the MPC during real-time operation. Both tests were run at Auburn University's 1.7 mile long National Center for Asphalt Technology (NCAT) test track near Opelika, Alabama. This is an oval track with lane markings that are similar in width to what is found on interstates and highways. Additionally, this track has an approximately 8 degree bank angle in the turns, and the effect of this was not explicitly accounted for in the control design.

The position of the vehicle over the entire first test at NCAT is shown in Figure 5.5, covering approximately two laps of the track. Note that the geoplot imagery provided from MATLAB is not high enough fidelity to align perfectly with the GPS track of the vehicle. However, the vehicle maintained tracking within the inner lane of the track throughout each run.

In the first test that is shown, the lateral position constraint in the controller was set to ± 1.0 meters of the centerline of the inner lane. The speed throughout both runs was controlled by the safety driver, while the steering was controlled by the autonomy system. The driver targeted 35 mph for these tests and speed varied slightly between 30-38 mph. However, an additional test is shown in Appendix B, with relaxed constraints and a top speed of 45 mph, demonstrating the capability towards higher speed control. The results of lateral position with respect to the lane centerline, and the absolute value of the lateral error for Test 1 is shown Figure 5.6. Throughout this test the mean of the absolute value of the lateral error was 18.95 cm, with a standard deviation of 11.91 cm.

The commanded and actual steer angle at the hand wheel throughout this test are shown in Figure 5.7. While the steer command had some amount of oscillation through the turns, there was

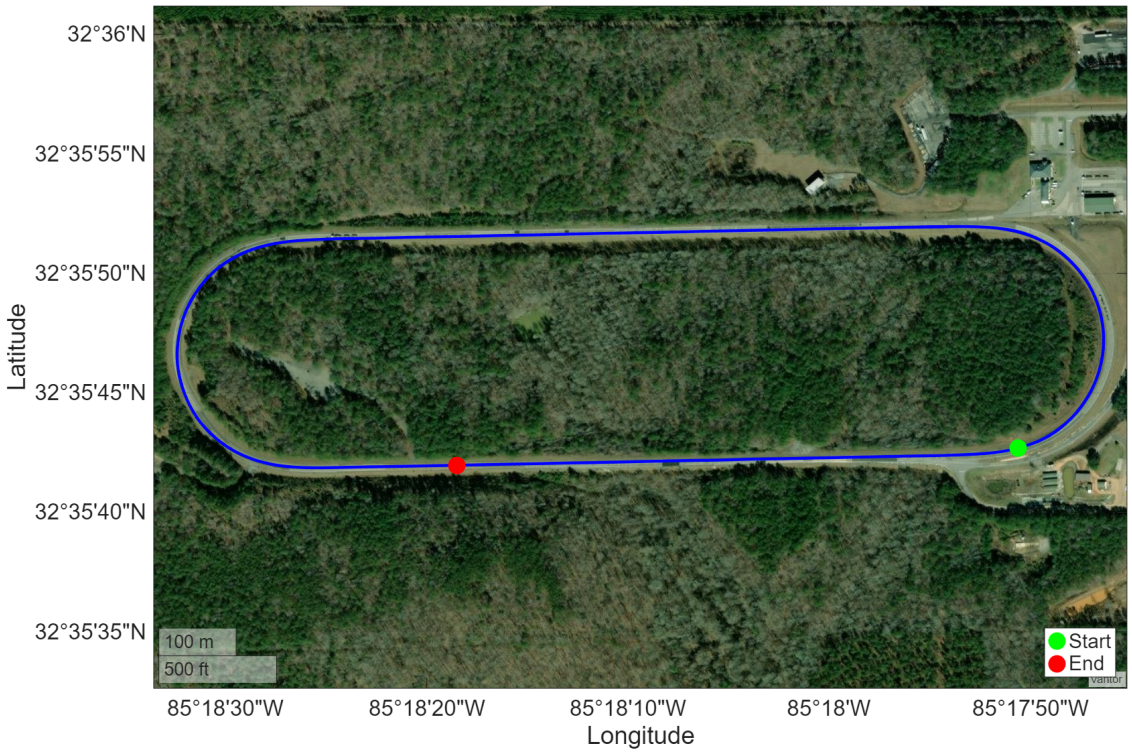

Figure 5.5: NCAT Test Track and Vehicle Position During Test 1

not enough jerk felt in the vehicle to cause great discomfort to the driver or passengers, and the controller remained in operation without intervention throughout all tests shown.

Figure 5.8 shows the hitch angle and hitch rate throughout the run. From the plot it can be determined that the hitch angle has a slight bias of approximately negative 1.0 degrees which becomes apparent on the straight sections of the track. This bias is due to the combination of the slight biases in the tractor and trailer heading estimates by the eTalin in each part of the vehicle, and was not accounted for in this experiment. However, the hitch angle itself was not weighted highly in the MPC, and therefore the bias appears to have had little impact on the performance.

Figure 5.9 shows the relative heading with respect to the lane centerline and Figure 5.10 shows the yaw rate and lateral velocity of the tractor.

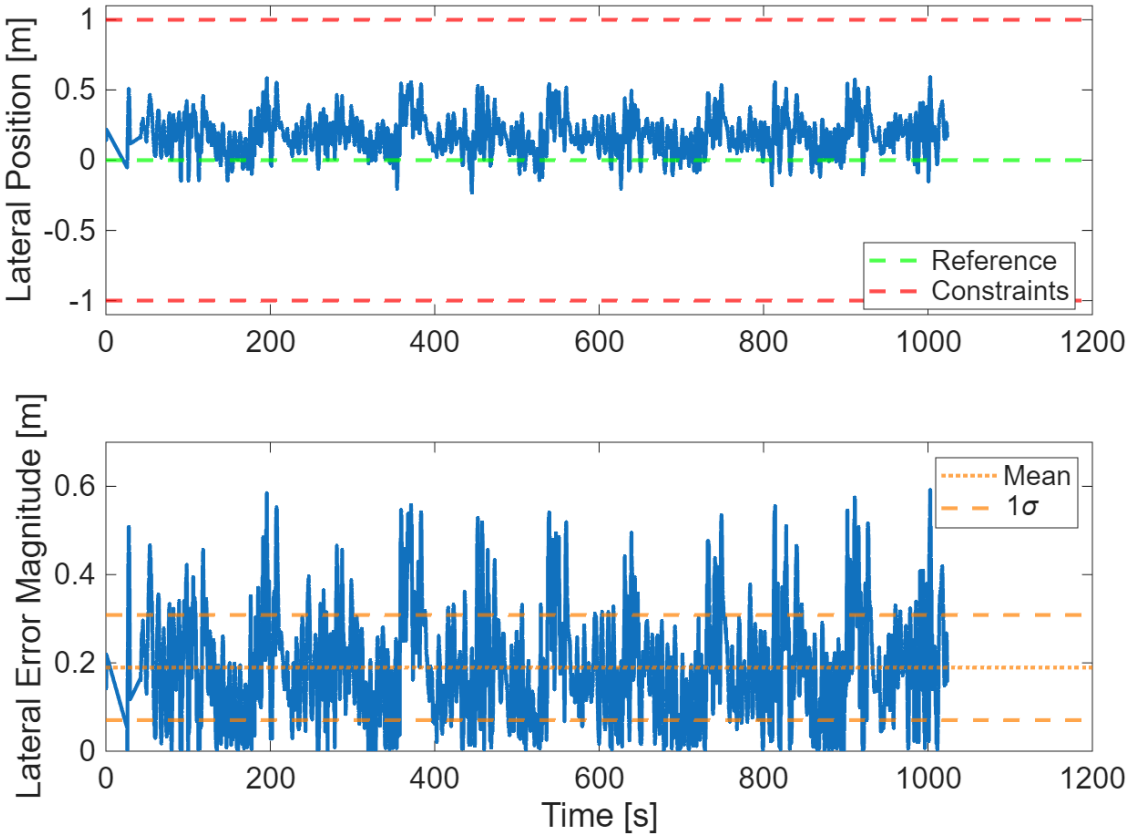

Figure 5.6: Test 1, Lateral Position & Error

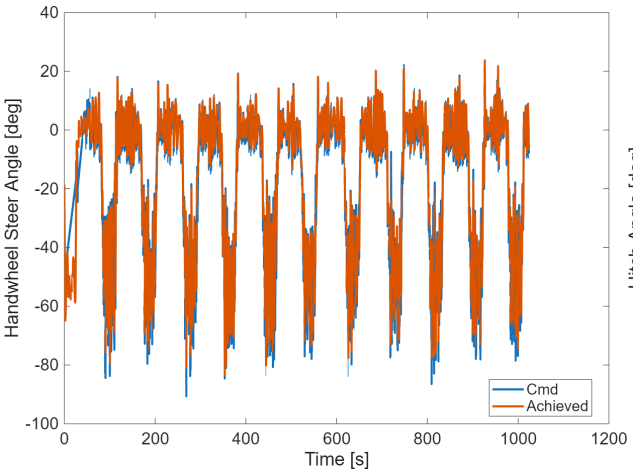

Figure 5.7: Test 1, Steer Angle at Hand Wheel

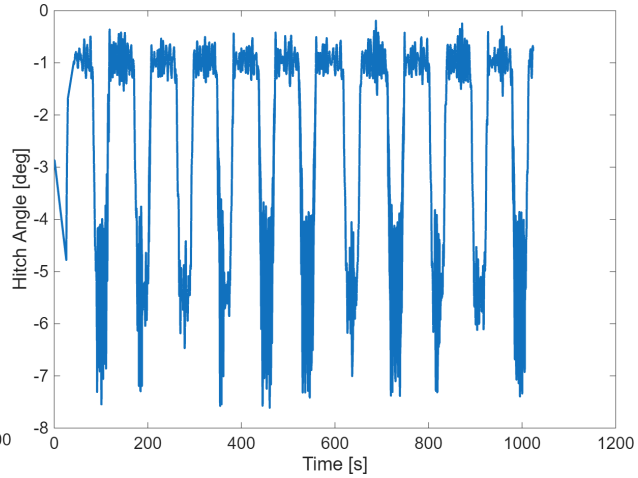

Figure 5.8: Test 1, Hitch Angle

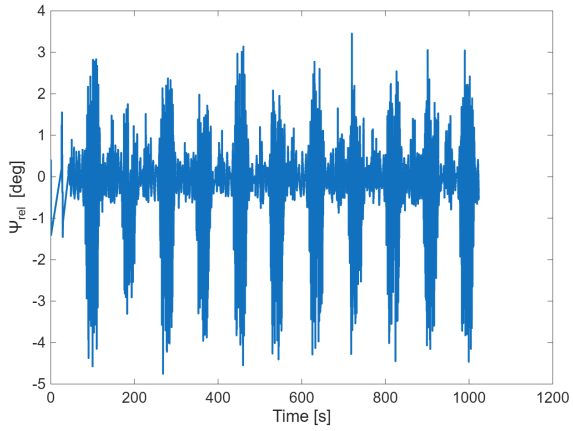

Figure 5.9: Test 1, Relative Heading

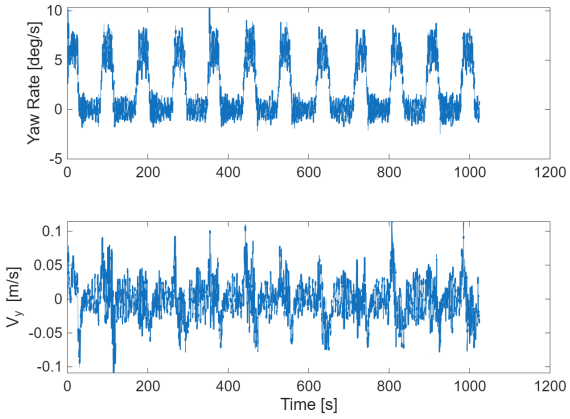

Figure 5.10: Test 1, Yaw Rate & Lateral Velocity

A second test was conducted with a smaller lateral position constraint of plus or minus 0.5 meters from the centerline of the inner lane. In this case, the mean of the absolute value of position error was 15.46 cm with a standard deviation of 10.06 cm, and the plots of lateral position and error are shown in Figure 5.11. The lateral position constraint was violated briefly in one instance at around 250 seconds into the test, and was likely caused by an external disturbance such as rough patches of road or potholes, unmodeled effects from road bank, or other similar disturbances. The steer angle and hitch angle are shown in Figures 5.12 and 5.13 respectively, and both had similar performance to test 1.

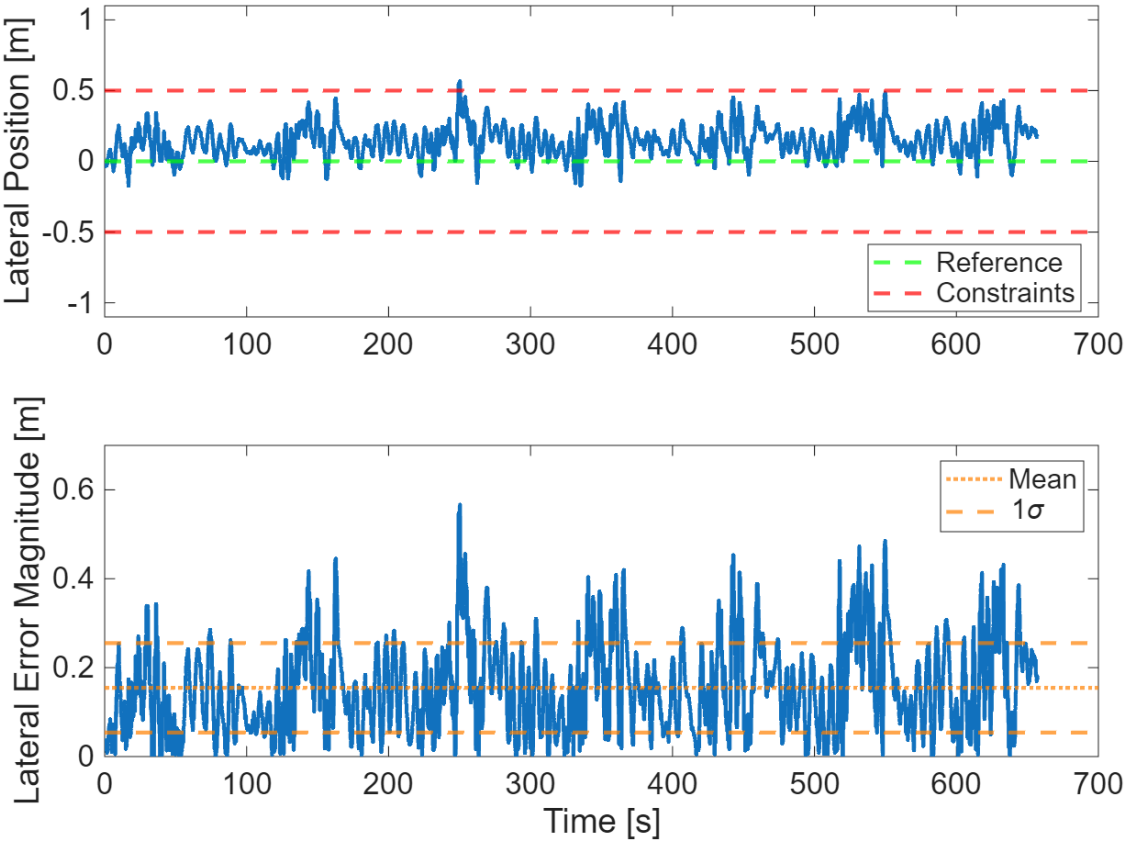

Figure 5.11: Test 2, Lateral Position & Error

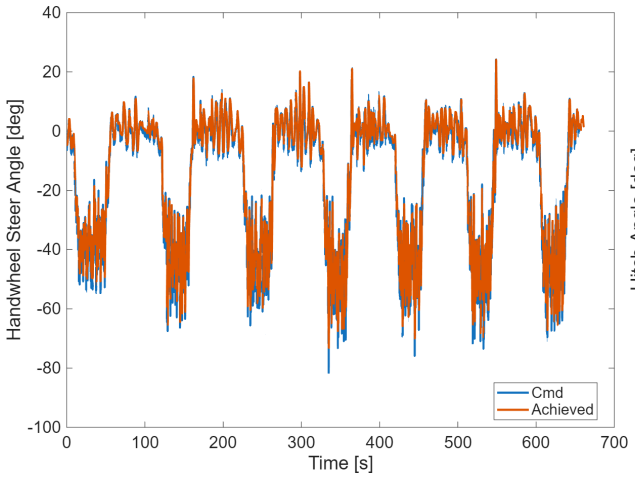

Figure 5.12: Test 2, Steer Angle at Hand Wheel

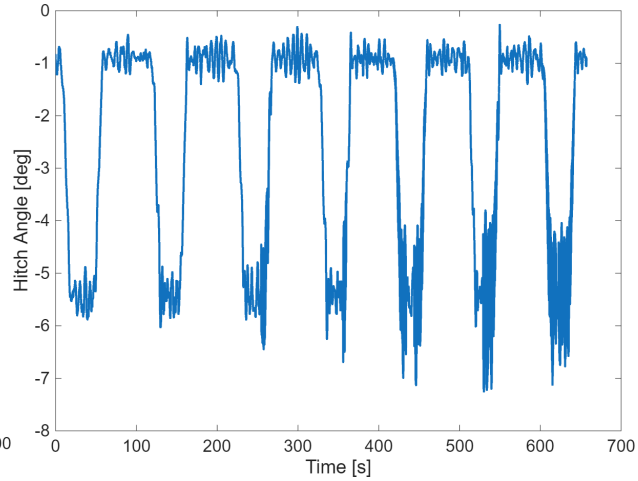

Figure 5.13: Test 2, Hitch Angle

Relative heading for this test is shown in Figure 5.14. Once again, the range is similar to the first test, and is larger in the turns than the straight sections of track. Yaw rate and lateral velocity of the tractor are shown in Figure 5.15.

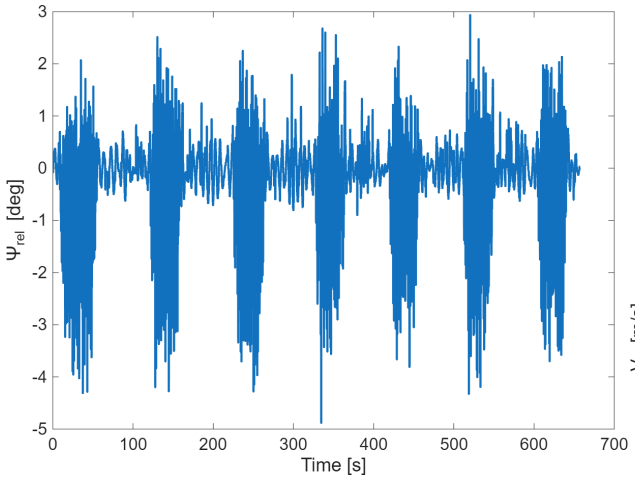

Figure 5.14: Test 2, Relative Heading

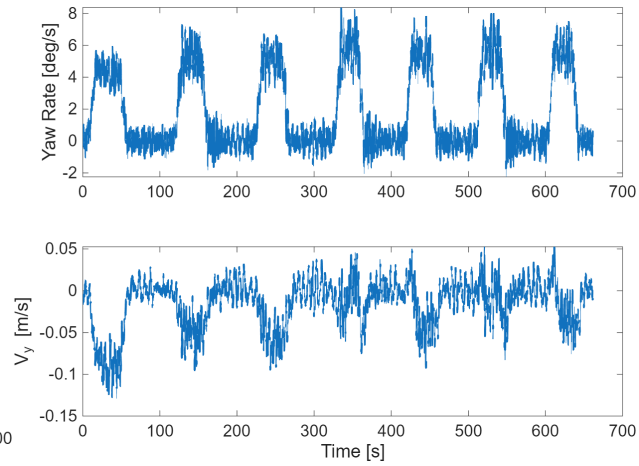

Figure 5.15: Test 2, Yaw Rate & Lateral Velocity

For greater detail, the lateral position and relative heading for the first 200 seconds of this test are shown in Figures 5.16 and 5.17 respectively. The steer angle at the hand wheel for the first 200 seconds is shown in 5.18. This time frame covers roughly one lap around the NCAT track.

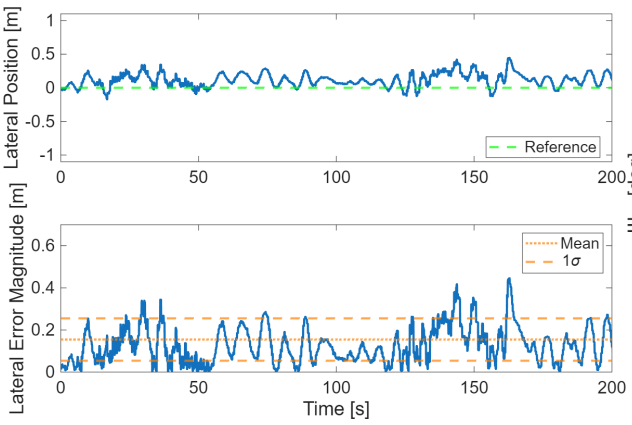

Figure 5.16: Test 2, Lateral Position 0-200 Seconds

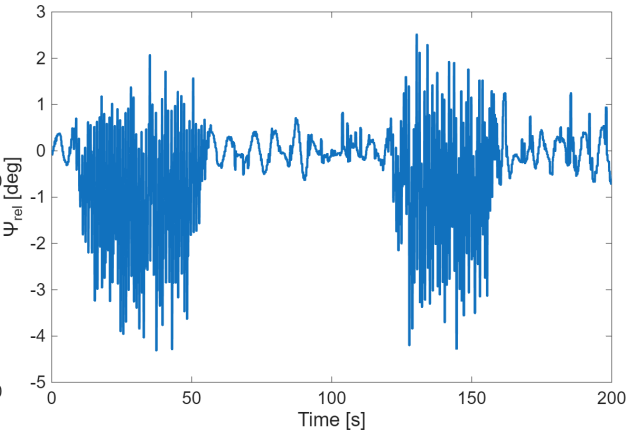

Figure 5.17: Test 2, Relative Heading 0-200 Seconds

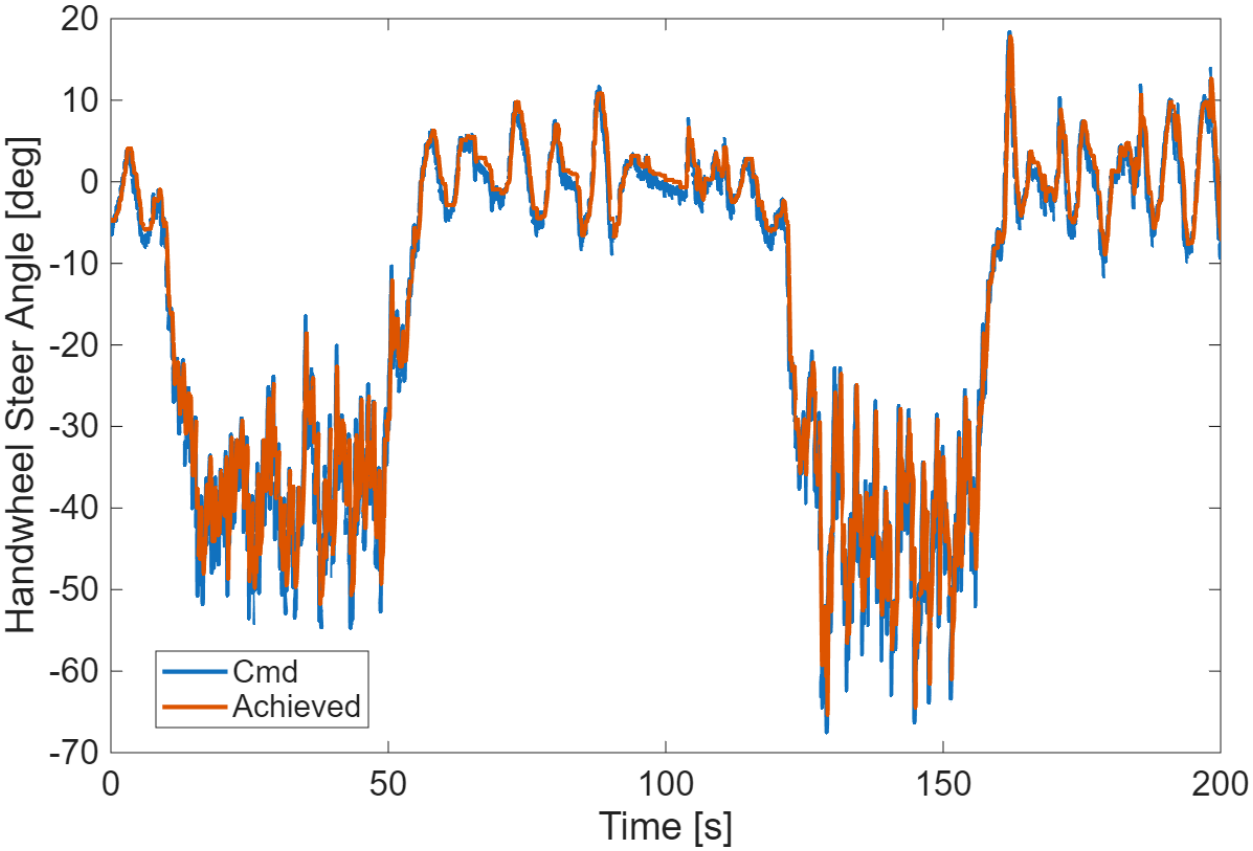

Figure 5.18: Test 2, Steer Angle 0-200 Seconds

Relative heading oscillation with a magnitude of roughly 0.5 deg is seen in the straight sections of the track. On the turns, this magnitude is somewhat increased because the heading is calculated from the tangent of the lane centerline. Although the error remained small and bounded throughout all of the testing, the oscillatory behavior may cause issues as speed is increased. Table 5.3 summarizes the lateral error throughout the experimental tests.

Test #	Mean Absolute Error	1σ
1	18.95 cm	11.91 cm
2	15.46 cm	10.06 cm

Table 5.3: Lateral Error Summary

In both tests, the truck was maintained within the lane lines throughout without any intervention by the driver. While there is room for improving the smoothness of the steering input in the turns specifically, there was not enough jerk or oscillation felt in the vehicle to cause major discomfort. Slight oscillatory behavior around the centerline of the road could also be improved and may allow for operation at higher speeds if resolved. However, these tests demonstrated the feasibility of the overall control method.

5.4 Computation Time

An analysis of the computation time was conducted in post-process on similar hardware to the OnLogic computer that was used in the testing. The inputs from the test were replayed from a ROS bag and the time to compute the solution was recorded using the C++ library Chrono [46]. The computer used for this analysis used an AMD Ryzen 5 8645hs processor with 6 CPU cores and a 4.2 GHz clock speed, and 32.0 GB RAM. The inputs from the entire first test were replayed and the computation time recorded. The histogram of the computation times is shown in Figure 5.19. Note that the constraint tightening method was active, so two QP solves are present in each total computation time that was recorded. Table 5.4 summarizes the computation time statistics.

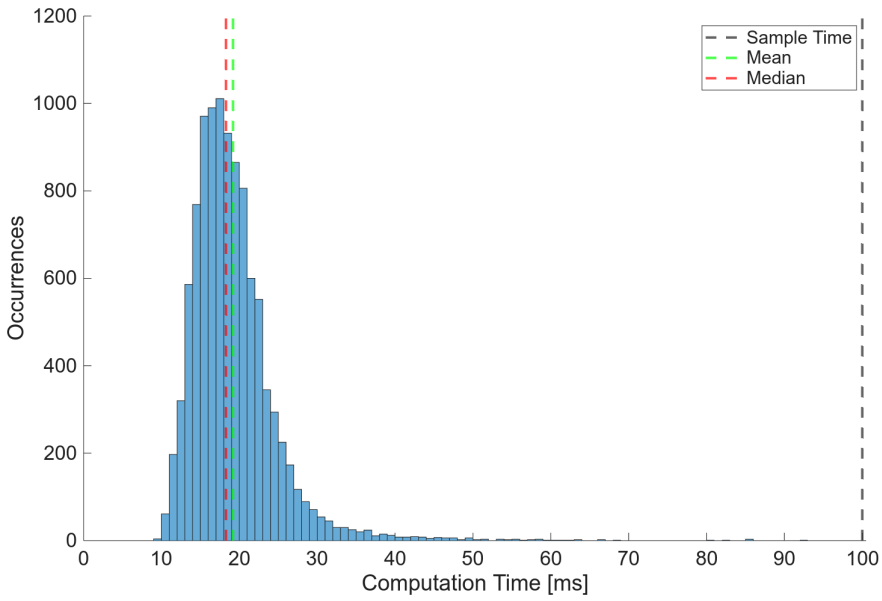

Figure 5.19: Histogram of Computation Times, Test 1

Mean	19.187 ms
Median	18.285 ms
1σ	5.597 ms
Max	92.314 ms

Table 5.4: Computation Time Statistics

Even in the worst case, the computation time was below the chosen sample rate of 10 Hz. However, if the algorithm is run on lower quality hardware, the sample rate may be exceeded in the worst cases. The worst cases are caused by scenarios where the solver's internal algorithm for solving the QP takes a large number of iterations or when the initial guess is far from the solution [36]. Providing intelligent initial guesses to the solver could be explored as a way to mitigate the number of solve times that are far above the average. The MPC horizon length or sample rate could also be modified to accommodate increased computation times.

5.5 Error Source Analysis

An analysis was performed in simulation to determine the possible causes of lateral position bias and oscillations seen in the experimental data. This lateral bias was in the range of 15-19 cm as summarized previously in Table 5.3, and was consistent throughout the tests. The relative heading

oscillated with a maximum magnitude of roughly 0.5 degrees on the straight sections of the track. The same simulation that was used in Chapter 4 was used to perform this analysis, which used the CasADi library in MATLAB to interface with the IPOPT solver, and used the nonlinear tractor-trailer vehicle model from Chapter 2 [38, 39]. The MPC was run at a 10 Hz sample rate, with the feedback controller at 20 Hz to match the sample rates used in the live tests. Each simulation was started with a negative 0.5 meter lateral offset from the centerline so that the transient can be seen. Baseline performance was established by simulating tracking with perfect knowledge of the state estimates. The effect of the lever arm between the CG of the tractor and the GPS antenna, and the effect of biased heading estimates, were then analyzed.

5.5.1 Baseline

First, a simulation was performed on a straight road without bias or noise added to the state feedback. This test established the performance under ideal conditions, and the results of lateral position, relative heading, and commanded steer input at the ground wheel are shown in Figures 5.20, 5.21, and 5.22 respectively. As expected, when given perfect state estimates and no other sources of error the lateral position converges to zero, tracking the centerline without bias.

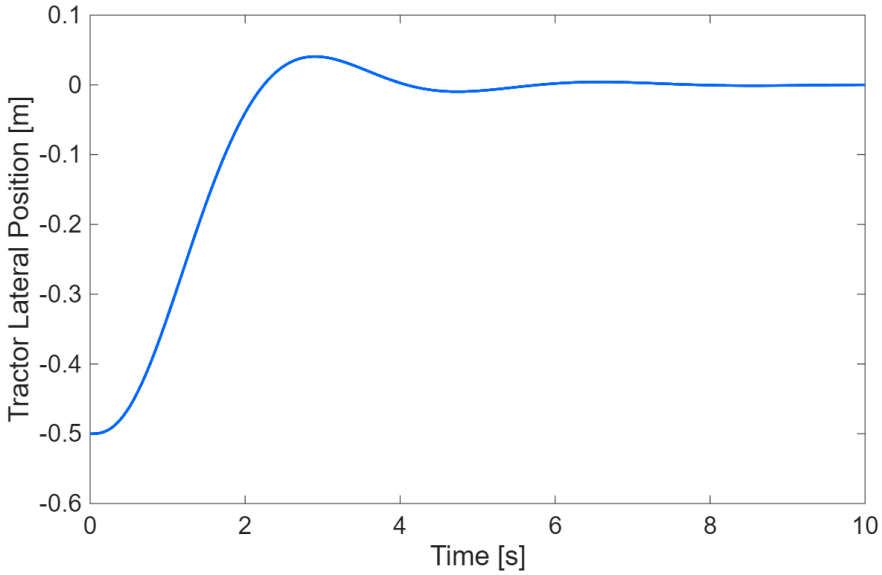

Figure 5.20: Tractor Lateral Position, Baseline

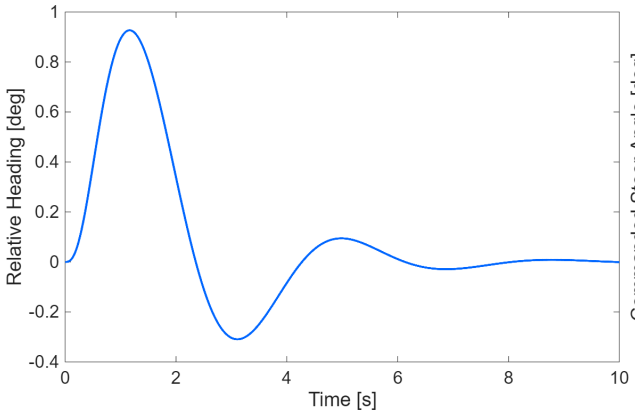

Figure 5.21: Relative Heading, Baseline

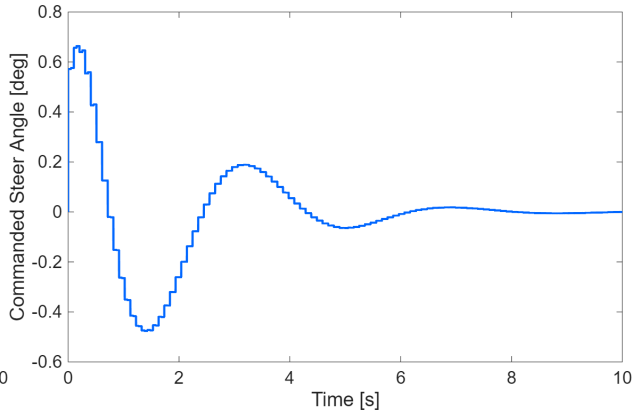

Figure 5.22: Steer Input, Baseline

5.5.2 Antenna Lever Arm

A simulation was performed to show the effect of an unaccounted lever arm between the tractor CG and the GPS antenna. Because the antenna on the test vehicle is mounted in line with the longitudinal axis of the tractor, the most likely error in the lever arm would be in the longitudinal direction rather than the lateral direction. Any error in the lateral component of the lever arm is likely to be on the order of a few centimeters. A 1.0 meter longitudinal offset in the vehicle's body frame was calculated from the CG position in simulation using Equations (5.3-5.4), before cross track position was calculated. In Equations (5.3-5.4), l_x is the longitudinal offset of the antenna position from the CG in the vehicle's body frame, while x_{cg} and y_{cg} represent the tractor's CG position in the simulation frame, and ψ is the heading in the simulation frame. The resulting antenna position in the simulation frame is represented by x_{ant} and y_{ant}.

$$x_{ant} = x_{cg} + l_x cos(\psi) \tag{5.3}$$

$$y_{ant} = y_{cg} + l_x sin(\psi) \tag{5.4}$$

The true lateral position result, calculated using the true tractor CG position alongside the lateral position calculated using the simulated antenna position, is show in Figure 5.23. Even with the large longitudinal offset of one meter that was applied, the difference between CG and antenna

lateral positions is negligible. The relative heading and steer inputs are shown in Figures 5.24 and 5.25.

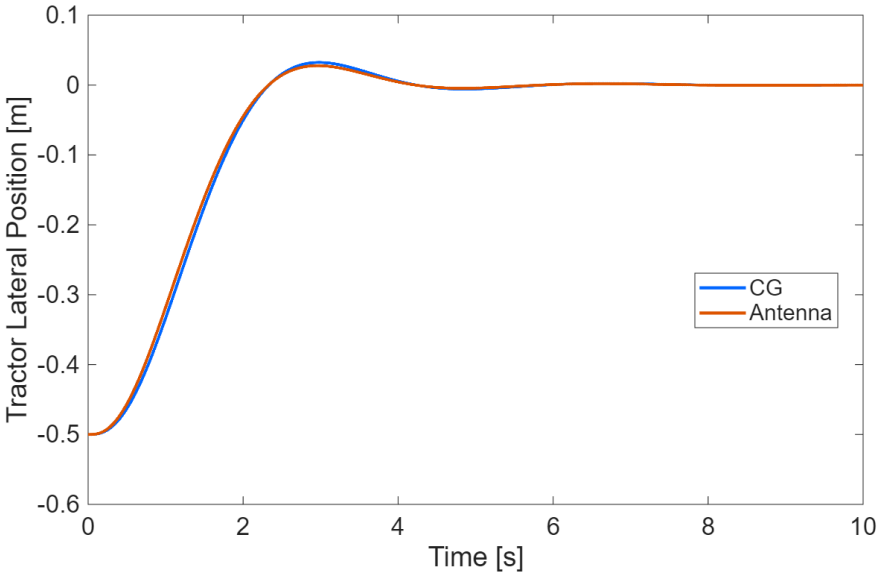

Figure 5.23: Tractor Lateral Position, Antenna Lever Arm

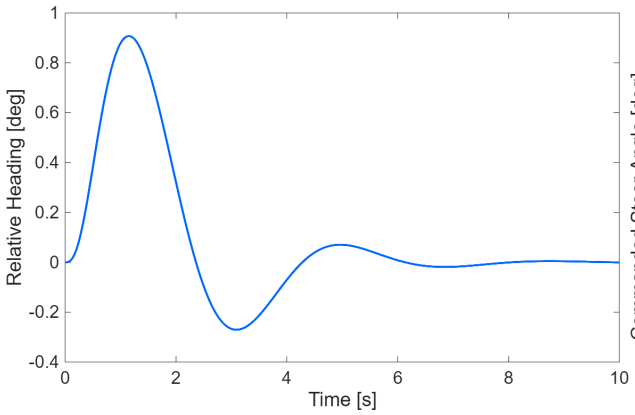

Figure 5.24: Relative Heading, Antenna Lever Arm

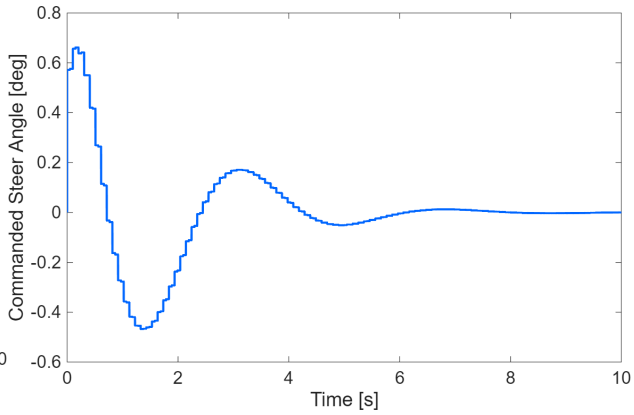

Figure 5.25: Steer Input, Antenna Lever Arm

5.5.3 Heading Bias

The simulation was repeated with a minus 1.0 degree bias added to the feedback of the tractor heading instead of an antenna lever arm. Figure 5.26 shows the lateral position during this simulation, and Figures 5.27 and 5.28 show the true and biased relative headings and steer command at the ground wheel.

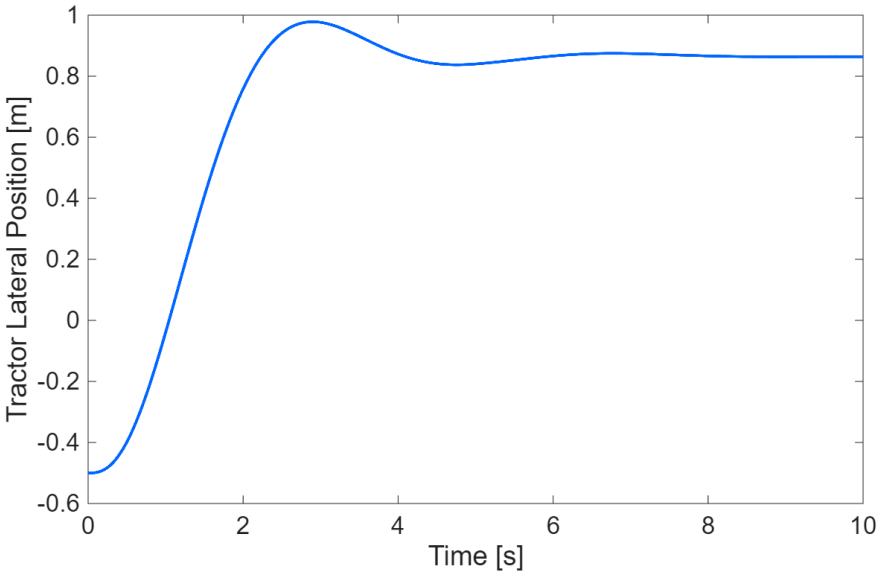

Figure 5.26: Tractor Lateral Position, Biased Heading Feedback

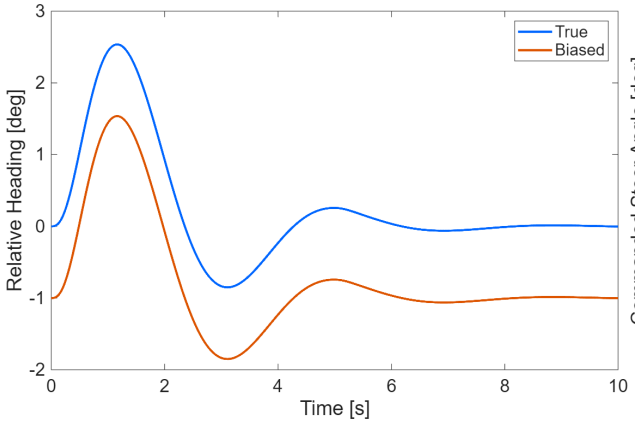

Figure 5.27: True Relative Heading, Biased Heading Feedback

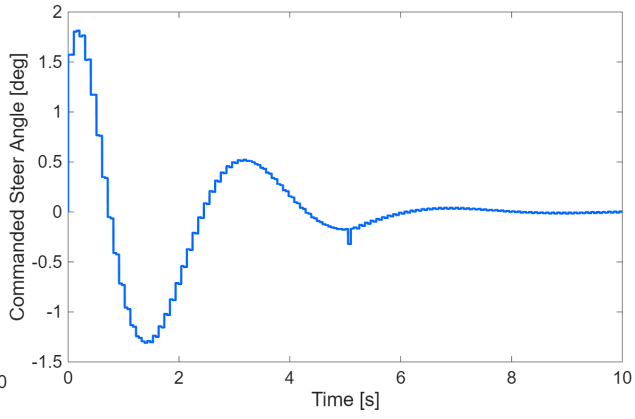

Figure 5.28: Steer Input, Biased Heading Feedback

Applying a bias to the heading feedback produces a bias in lateral position that is similar to what was seen in the experimental data, although not identical in magnitude. When presented with

a biased source of heading, the MPC eventually settles onto a constant zero steering command even while lateral position is non-zero. This is because the MPC predicts that the lateral position will move towards zero because of the incorrect heading, so holding a steer angle of zero is the optimal solution. Heading bias from the navigation solution was also seen in the experimental data, indicating that this is the likely cause of the lateral position bias that was seen in the experimental testing.

5.5.4 Heading Bias and Lateral Position Measurement Noise

A simulation was also performed with a negative 0.5 degree heading bias and measurement noise applied to lateral position with a 5 cm standard deviation. The resulting lateral position is shown in Figure 5.29, the relative heading in Figure 5.30, and the steer input in Figure 5.31.

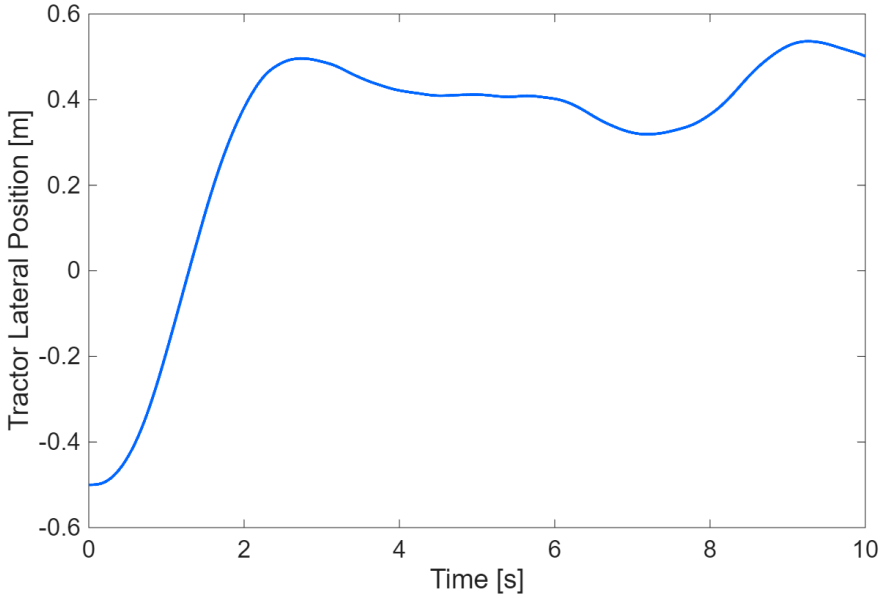

Figure 5.29: Tractor Lateral Position, Biased and Noisy Feedback

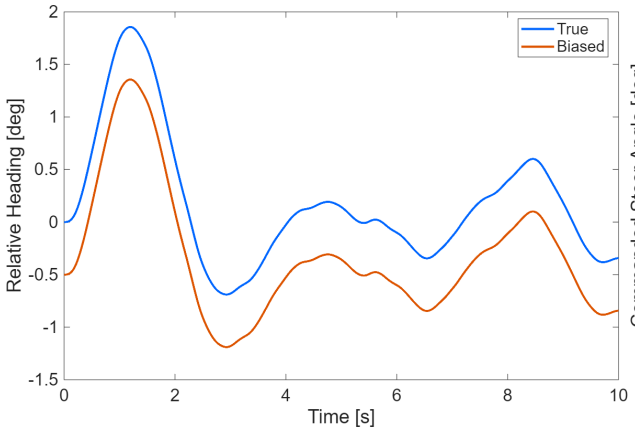

Figure 5.30: Relative Heading, Biased and Noisy Feedback

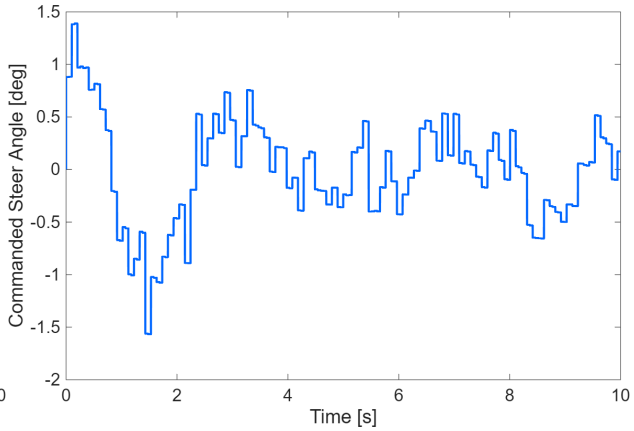

Figure 5.31: Steer Input, Biased and Noisy Feedback

When a heading bias is combined with noisy lateral position measurements, a result similar to what was seen in the experimental data is produced, in which heading and lateral position both showed oscillation and lateral position also had a mean error. This result indicates that improving the heading estimate would likely improve the mean error in lateral position seen in the experimental data, and is an avenue of future work.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

While autonomy in the world of commercial trucking is becoming more prevalent, methods for automated control of these vehicles in the context of SAE Level 2-3 features is still an active area of research and development. Practical methods for motion planning and control are especially relevant given the unique requirements for class 8 tractor-trailers, and the computational and sensing restrictions on the vehicles that are currently being produced.

This thesis presented an Linear Time-Varying Model Predictive Control (LTV-MPC) design that is capable of being solved as a Quadratic Program (QP), which is useful for real-time or embedded environments. When combined with a constraint tightening method, the control design is capable of executing avoidance maneuvers while respecting position constraints even with error on the model parameters. The method for creating an error tube and tightening constraints, while somewhat overly conservative, provided satisfactory coverage of the spread of possible trajectories in most scenarios. The simulation study consisted of avoidance maneuvers at high speeds while varying parameters of the trailer payload and tires. This validated the design and demonstrated the strengths and limitations of using the sensitivity based error tube. Furthermore, the control system was also validated through real-time experimental testing. Although the constraint tightening method does not have a noticeable impact during normal driving conditions, these tests demonstrated the capability of the system for real-time operation. This study demonstrated the performance of the MPC as a lane keeping controller when run at 10 Hz with full feedback of the

trailer states. The complete design was implemented in C++ and tested on a Peterbilt 579 with a trailer attached. The system was capable of extended periods of lateral control without the driver intervening to steer, with the worst tests having mean absolute lateral errors of 18.95 cm, with a standard deviation of 11.91 cm.

6.2 Future Work

This thesis leads to several additional areas of future work. First, alternative methods of Robust MPC could be explored for this application and compared to the method presented in this work. The tube construction could be expanded to include more types of uncertainty following the method shown in recent work [17]. There are also a wide variety of methods for improving robustness, and it is possible that there are methods which would enhance safety and performance in a wider range of scenarios. Furthermore, extreme scenarios such as jackknifing, very low friction surface conditions, and large external disturbances from road conditions or cross-wind should be analyzed, which would provide more valuable areas of research for the application of RMPC methods to tractor trailer control. Based on applications in prior literature, RMPC methods stand out when the vehicle is pushed to limits that standard MPC cannot recover from [20].

In addition to different types of RMPC, alternative approaches could be explored for ensuring constraints are satisfied. Control Barrier Functions (CBFs) are one alternative method that have been proposed. These are intended to be used in combination with a controller, and modify the control input by the minimal necessary amount to prevent the system from violating constraints [47]. CBFs have demonstrated potential for automotive control due to their ability to maintain robustness while being computationally inexpensive.

Aside from satisfying constraints, uncertainty from parameter estimates, unmodeled effects, and external disturbances have an impact on the control performance even when constraints are not active. Instead of only modifying constraints, methods of adapting for these effects could be used. Disturbance observers are one technique that could be employed. These aim to model the difference between the prediction model and the real vehicle's performance [48]. The effect of the

disturbance can then be accounted for by the controller. Online adaptation of the model parameters could also be implemented to improve performance during operation.

The existing design could likely achieve better performance through an automated tuning process. The weights in this study were selected by a hand tuning process based on knowledge of the effect of each weight on the closed loop performance. The constraints should prevent dangerous behavior regardless of the weights that are chosen. However, in practice the weights have an impact on the performance and comfort for drivers and passengers. Bayesian Optimization is one method that has been implemented for this task, with previous work demonstrating it as a method for automated tuning of a vehicle platooning controller [49].

The model used in the design of the LTV MPC was linearized so that the optimization problem can be solved as a QP, which has computational advantages. However, Real-Time Iteration (RTI) schemes can allow for more accurate NMPC formulations to be solved online while still only requiring one QP solution per update of the controller [50]. RTI works by treating each update of the controller as one step of a Sequential Quadratic Program (SQP), which is a technique for solving optimization problems that can have nonlinear constraints. Implementation of MPC within this framework may prove to be a valuable direction for future work on this application. This would provide an efficient way to solve optimal control problems online and on low cost hardware without restricting the problem to quadratic cost and linear constraints.

In this thesis, longitudinal control was done manually by the driver in experimental tests, and held constant in simulation studies. Therefore, including longitudinal control is an important next step in this work. This could be accomplished by adding an additional control variable to the model, and would be expected to improve feasibility in object avoidance scenarios. This is because the controller would have a greater ability to avoid obstacles by commanding braking and steering. Due to imperfect braking control this introduces another source of uncertainty, and may require different techniques for maintaining robustness.

References

[1] Ani Kelkar, Kersten Heineke, Martin Kellner, and Timo Möller. The business case for autonomous truck fleets, September 2024. Accessed: 27-Dec-2025.

[2] SAE International. Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. Standard J3016_202104, SAE International, Warrendale, PA, 2021. Revised April 2021.

[3] Detroit. Detroit Assurance Optional Features. https://www.demanddetroit.com/assurance/optional-features/, 2025. Demand Detroit. Accessed: 2025-12-26.

[4] Volvo Trucks North America. Safety Drives Everything in the All-New Volvo VNL. https://www.volvotrucks.us/news-and-stories/press-releases/2024/march/safety-drives-everything-in-the-all-new-volvo-vnl/, mar 2024. Volvo Trucks North America. Accessed: 2025-12-26.

[5] Plus and Bosch. Bosch and Plus Collaborate on Assisted Driving Solution for Commercial Vehicles, 05 2023.

[6] Aurora Innovation, Inc. Aurora Begins Commercial Driverless Trucking in Texas, Ushering in a New Era of Freight, 05 2025.

[7] Kodiak AI. Kodiak Driver Earns Top Safety Score in Nauto Evaluation, 10 2025.

[8] James B. Rawlings, David Q. Mayne, and Moritz M. Diehl. *Model Predictive Control: Theory, Design, and Implementation*. Nob Hill Publishing, LLC, Santa Barbara, CA, USA, 2009. First edition (original printing).

[9] T. Wu and J. Y. Hung. Path following for a tractor-trailer system using model predictive control. In *2017 IEEE SoutheastCon*, pages 1–5, Concord, NC, USA, March 2017. IEEE.

[10] Jiajia Chen, Shaodong Jiang, Zheng Zhou, Mengyu Zhang, Xiaoke Ming, and Ning Guo. Lateral semi-trailer truck control using a parameter self-learning mpc method in urban environment. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 238(5):964–976, 2024.

[11] Mechanical Simulation Corporation. *TruckSim*. Mechanical Simulation Corporation, Ann Arbor, MI, 2011. Version 8.

[12] Carl Hynén Ulfsjöö and Theodor Westny. Modeling and lateral control of tractor-trailer vehicles during aggressive maneuvers. Master's thesis, Linköping University, 2020. Supervisor: Oskar Ljungqvist, Examiner: Daniel Axehill.

[13] Jacob W. Ward, J. Daniel Pierce, Lowell Brown, and David M. Bevly. Design and implementation of an sae level-2 lane keeping system for class 8 trucks using nonlinear model predictive control. In *2023 IEEE Conference on Control Technology and Applications (CCTA)*, pages 841–846, Bridgetown, Barbados, August 2023. IEEE.

[14] Spencer M. Richards. Robust and adaptive MPC. Lecture Notes, AA 203: Optimal and Learning-Based Control, Autonomous Systems Laboratory, Stanford University, May 2023. Last updated May 17, 2023.

[15] Ali Srour, Salvatore Marcellini, Tommaso Belvedere, Marco Cognetti, Antonio Franchi, and Paolo Robuffo Giordano. Experimental validation of sensitivity-aware trajectory planning for a quadrotor uav under parametric uncertainty. In *2024 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 572–578, 2024.

[16] Tommaso Belvedere, Marco Cognetti, Giuseppe Oriolo, and Paolo Robuffo Giordano. Sensitivity-aware model predictive control for robots with parametric uncertainty. *IEEE Transactions on Robotics*, 41:3039–3058, 2025.

[17] J. Zhu, T. Simeon, and M. Cognetti. Robust sensitivity-aware chance-constrained MPC for efficient handling of multiple uncertainty sources. *IEEE Robotics and Automation Letters*, 2025.

[18] Michael Maiworm, Tobias Bäthge, and Rolf Findeisen. Scenario-based model predictive control: Recursive feasibility and stability. *IFAC-PapersOnLine*, 48(8):50–56, 2015.

[19] F. Fiedler, B. Karg, L. Lüken, D. Brandner, M. Heinlein, F. Brabender, and S. Lucia. do-mpc: Towards fair nonlinear and robust model predictive control. *Control Engineering Practice*, 140:105676, 2023. Software and theory documentation at `https://www.do-mpc.com/`.

[20] Yiqi Gao, Andrew Gray, H. Eric Tseng, and Francesco Borrelli. A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles. *Vehicle System Dynamics*, 52(6):802–823, 2014.

[21] Erkan Kayacan, Erdal Kayacan, Herman Ramon, and Wouter Saeys. Robust tube-based decentralized nonlinear model predictive control of an autonomous tractor-trailer system. *IEEE/ASME Transactions on Mechatronics*, 23(1):197–205, January 2015.

[22] Sangwon Han, Kyusang Yoon, Geonyeong Park, and Kunsoo Huh. Robust lane keeping control for tractor with multi-unit trailer under parametric uncertainty. *IEEE Transactions on Intelligent Vehicles*, 9(1):2333–2347, January 2024.

[23] Tahn Thawainin, Tyler Flegel, and David Bevly. Surveying CNNs for estimating trailer articulation angle using monocular cameras. In *2025 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2025.

[24] Amin Habibnejad Korayem, Amir Khajepour, and Baris Fidan. Trailer mass estimation using system model-based and machine learning approaches. *IEEE Transactions on Vehicular Technology*, 69(11):12536–12546, 2020.

[25] Jacob Ward, Nan Li, David Bevly, and Lowell Brown. Tractor-trailer vehicle rollover avoidance using chance-constrained reference governor and data-driven ultra-local model. *IFAC-PapersOnLine*, 58(28):1025–1030, 2024. The 4th Modeling, Estimation, and Control Conference – 2024.

[26] Sage Wolfe. *Heavy Truck Modeling and Estimation for Vehicle-to-Vehicle Collision Avoidance Systems*. Doctoral dissertation, The Ohio State University, 2014. OhioLINK Electronic Theses and Dissertations Center.

[27] Zachary Brock, James Nelson, and Ross L. Hatton. A comparison of lateral dynamic models for tractor-trailer systems. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 2052–2059, 2019.

[28] Mehrzad Tabatabaian. *Engineering Systems Dynamics, Modelling, Simulation, and Design*. BCcampus Open Publishing, 2021. Open Educational Resource.

[29] Zhang Tantan, Yueshuo Sun, Yazhou Wang, Bai Li, Yonglin Tian, and Fei-Yue Wang. A Survey of Vehicle Dynamics Modeling Methods for Autonomous Racing: Theoretical Models, Physical/Virtual Platforms, and Perspectives, March 2024.

[30] João P. Hespanha. *Linear Systems Theory*. Princeton University Press, Princeton, NJ, 2nd edition, 2018.

[31] Charles Van Loan. Computing integrals involving the matrix exponential. *IEEE Transactions on Automatic Control*, AC-23(3):395–404, June 1978.

[32] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

[33] Rajesh Rajamani. *Vehicle Dynamics and Control*. Springer, Boston, MA, 2nd edition, 2012. Series: Springer Series in Mechanical Engineering.

[34] John T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. Cambridge University Press, 2 edition, 2010.

[35] Liuping Wang. *Model Predictive Control System Design and Implementation Using MATLAB*. Springer-Verlag, London, 2009.

[36] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. OSQP: an operator splitting solver for quadratic programs. *Mathematical Programming Computation*, 12(4):637–672, 2020.

[37] Allan M. M. Leal. autodiff, a modern, fast and expressive C++ library for automatic differentiation. `https://autodiff.github.io`, 2018.

[38] Joel A. E. Andersson, Joris Gillis, Greg Horn, James B. Rawlings, and Moritz Diehl. CasADi: A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019.

[39] Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

[40] Kepeng Qiu. Zoomplot. `https://github.com/iqiukp/ZoomPlot-MATLAB/releases/tag/v1.5.1`, 2026. GitHub. Retrieved April 11, 2026.

[41] Honeywell Aerospace. etalin ins/gnss navigator with embedded gps receiver, 2020. Brochure N61-2528-000-000 I 09/20.

[42] NovAtel Inc. *FlexPak6 GNSS Receiver Enclosure*. NovAtel Inc., Calgary, Canada, 2016. Version 11, D15802 October 2016.

[43] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Ng. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009. Available at: `http://www.willowgarage.com/sites/default/files/icraoss09-ROS.pdf`.

[44] Google. osqp-cpp: A C++ interface for the OSQP quadratic programming solver, 2025. Accessed: May 19, 2026.

[45] Fabian Poggenhans, Jan-Hendrik Pauls, Johannes Janosovits, Stefan Orf, Maximilian Naumann, Florian Kuhnt, and Matthias Mayr. Lanelet2: A high-definition map framework for the future of automated driving. In *Proc. IEEE Intell. Trans. Syst. Conf.*, pages 1672–1679, Hawaii, USA, November 2018.

[46] ISO/IEC. *ISO International Standard ISO/IEC 14882:2014(E) – Programming Language C++*. International Organization for Standardization (ISO), Geneva, Switzerland, 2014. [Standard].

[47] Aaron D. Ames, Xiangru Xu, Jessy W. Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2017.

[48] Gabriele Pannocchia. Offset-free tracking mpc: A tutorial review and comparison of different formulations. In *2015 European Control Conference (ECC)*, pages 527–532, 2015.

[49] Nianhua Zhang, Shima Nazari, Jicheng Chen, Xiangyang Xu, and Hui Zhang. Bayesian optimization for automatic tuning of an energy aware 2-d vehicle platoon. *IEEE Transactions on Intelligent Vehicles*, 9(11):7380–7391, 2024.

[50] Sébastien Gros, Mario Zanon, Rien Quirynen, Alberto Bemporad, and Moritz Diehl. From linear to nonlinear mpc: bridging the gap via the real-time iteration. *International Journal of Control*, 93(1):62–80, 2020.

Appendices

Appendix A

Lateral Dynamics Matrices

The values of the mass, stiffness and input matrices used in this thesis are from Revision 4 of the model presented in [26]. This revision includes a small hitch angle assumption ($cos(\gamma) \approx 1$). The mass matrix is shown in equation (A.1). The variable naming has been adapted to match the notation used throughout this thesis.

$$M = \begin{bmatrix} m_1 + m_2 & -m_2(c+d) & v_x(m_1+m_2) & -m_2 d & 0 \\ -m_2(c+d) & J_1 + J_2 + m_2(c+d)^2 & -m_2 v_x(c+d) & J_2 + m_2 d^2 + m_2 cd & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -m_2 d & J_2 + m_2 d^2 + m_2 cd & -m_2 v_x d & J_2 + m_2 d^2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.1}$$

The stiffness matrix is given by equation (A.2), where the entries of the matrix are shown in equations (A.3)-(A.14).

$$K = \begin{bmatrix} k_{11} & k_{12} & 0 & k_{14} & k_{15} \\ k_{21} & k_{22} & 0 & k_{24} & k_{25} \\ 0 & 1 & 0 & 0 & 0 \\ k_{41} & k_{42} & 0 & k_{44} & k_{45} \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{A.2}$$

$$k_{11} = \frac{1}{v_x}[-C_{\alpha,1} - C_{\alpha,2} - C_{\alpha,3} - cos(\gamma)C_{\alpha,4} - cos(\gamma)C_{\alpha,5}] \tag{A.3}$$

$$k_{12} = \frac{1}{v_x}[-C_{\alpha,1} + C_{\alpha,2}b_1 + C_{\alpha,3}b_2 +$$

$$cos(\gamma)C_{\alpha,4}(c + f_1cos(\gamma)) + cos(\gamma)C_{\alpha,5}(c + f_2cos(\gamma)) \tag{A.4}$$

$$k_{14} = \frac{1}{v_x}[cos^2(\gamma)f_1C_{\alpha,4} + cos^2(\gamma)f_2C_{\alpha,5}] \tag{A.5}$$

$$k_{15} = cos(\gamma)C_{\alpha,4} + cos(\gamma)C_{\alpha,5} \tag{A.6}$$

$$k_{21} = \frac{1}{v_x}[-aC_{\alpha,1} + b_1C_{\alpha,2} + b_2C_{\alpha,3} + (f_1 + ccos(\gamma))C_{\alpha,4} + (f_2 + ccos(\gamma))C_{\alpha,5}] \tag{A.7}$$

$$k_{22} = \frac{1}{v_x}[-a^2C_{\alpha,1} - b_1^2C_{\alpha,2} - b_2^2C_{\alpha,3} -$$

$$(f_1 + ccos(\gamma))C_{\alpha,4}(c + f_1cos(\gamma)) - (f_2 + ccos(\gamma))C_{\alpha,5}(c + f_2cos(\gamma))] \tag{A.8}$$

$$k_{24} = \frac{1}{v_x}[-(f_1 + ccos(\gamma))C_{\alpha,4}f_1cos(\gamma) - (f_2 + ccos(\gamma))C_{\alpha,5}f_2cos(\gamma)] \tag{A.9}$$

$$k_{25} = -(f_1 + ccos(\gamma))C_{\alpha,4} - (f_2 + ccos(\gamma))C_{\alpha,5} \tag{A.10}$$

$$k_{41} = \frac{1}{v_x}[f_1C_{\alpha,4} + f_2C_{\alpha,5}] \tag{A.11}$$

$$k_{42} = \frac{1}{v_x}[-f_1C_{\alpha,4}(c + f_1cos(\gamma)) - f_2C_{\alpha,5}(c + f_2cos(\gamma))] \tag{A.12}$$

$$k_{44} = \frac{1}{v_x}[-f_1^2C_{\alpha,4}cos(\gamma) - f_2^2C_{\alpha,5}cos(\gamma)] \tag{A.13}$$

$$k_{45} = -f_1C_{\alpha,4} - f_2C_{\alpha,5} \tag{A.14}$$

The input matrix is shown in equation (A.15).

$$F = \begin{bmatrix} cos(\delta)C_{\alpha,1} \\ acos(\delta)C_{\alpha,1} \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{A.15}$$

Appendix B

Additional Experimental Results

This appendix shows results from an additional experimental test with higher speed and relaxed lateral position constraints. This test was intended to stress some of the assumptions of the design by applying larger longitudinal accelerations, and taking somewhat higher speeds through the turns. This test ranged in speed from approximately 25 mph to a brief maximum speed of 45 mph on a straight section of the track. However, a longer period of time was spent at roughly 37-40 mph. Figure B.1 shows the wheel speed throughout the duration of the test.

Again the goal was to track the centerline of the lane without leaving the lane boundaries. Constraints on lateral position and hitch angle were relaxed so that they were effectively inactive. The lateral error results are shown in Figure B.2. The vehicle had a mean absolute lateral error of 20.27 cm throughout the test, with a 1σ of 13.54 cm. The maximum lateral error was approximately 0.7 m which was quickly recovered from. Relative heading with respect to the road centerline and hitch angle are shown in Figures B.4 and B.5 respectively. The steer angles are given in figure B.3, and the yaw rate and lateral velocity are shown in Figure B.6.

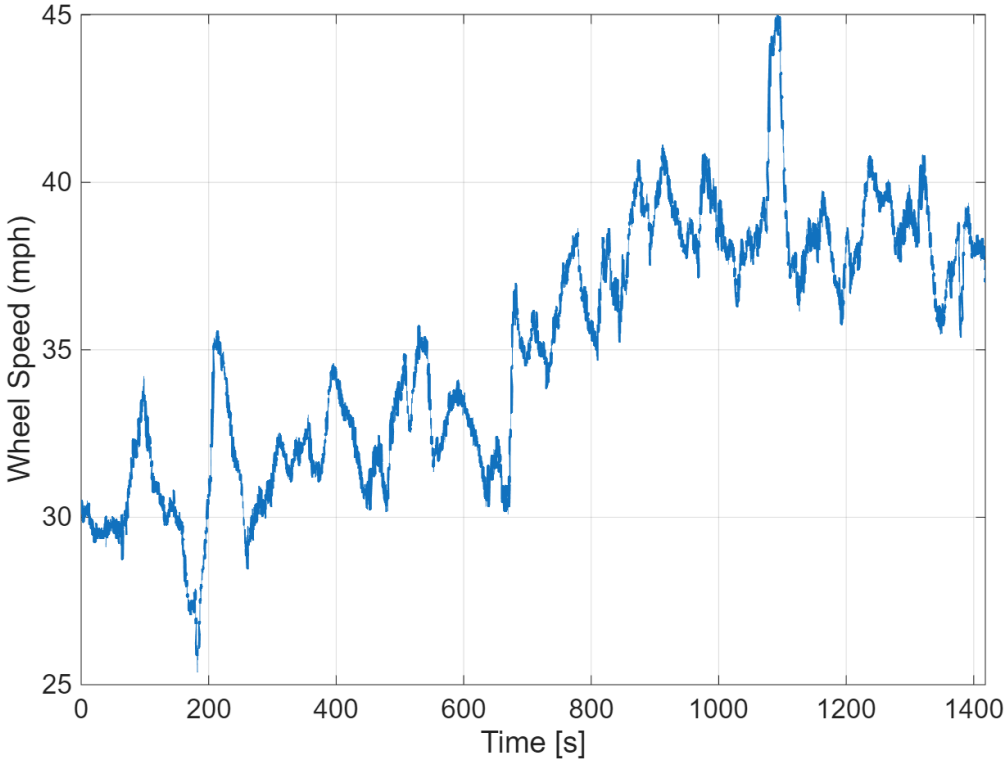

Figure B.1: Wheel Speed

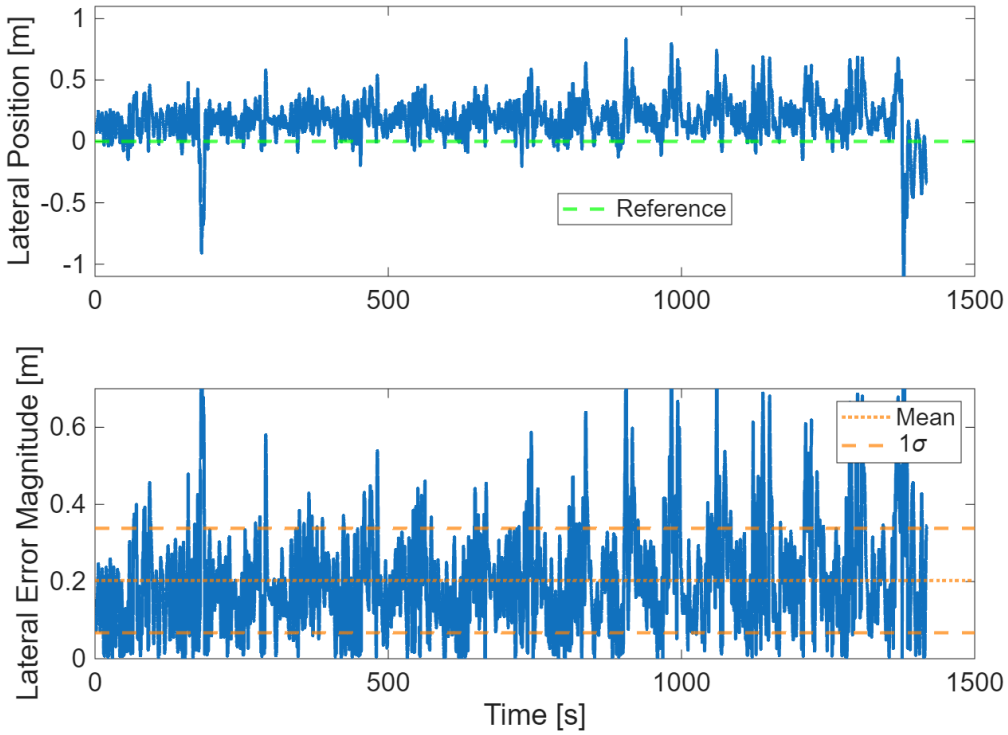

Figure B.2: Lateral Errors

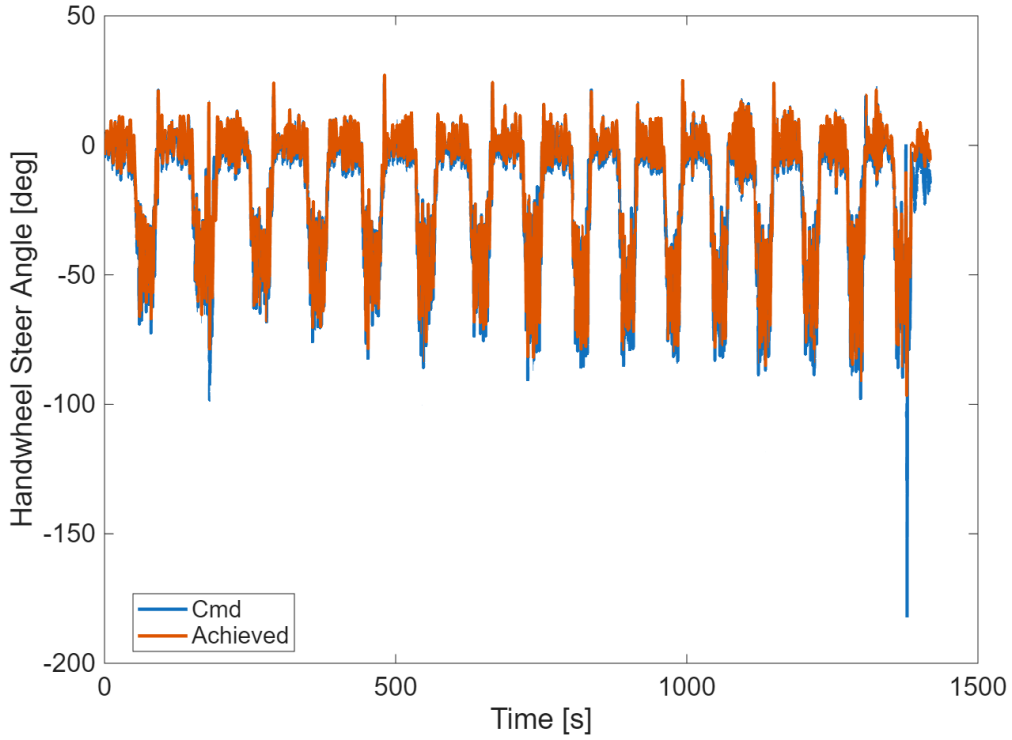

Figure B.3: Steer Angle

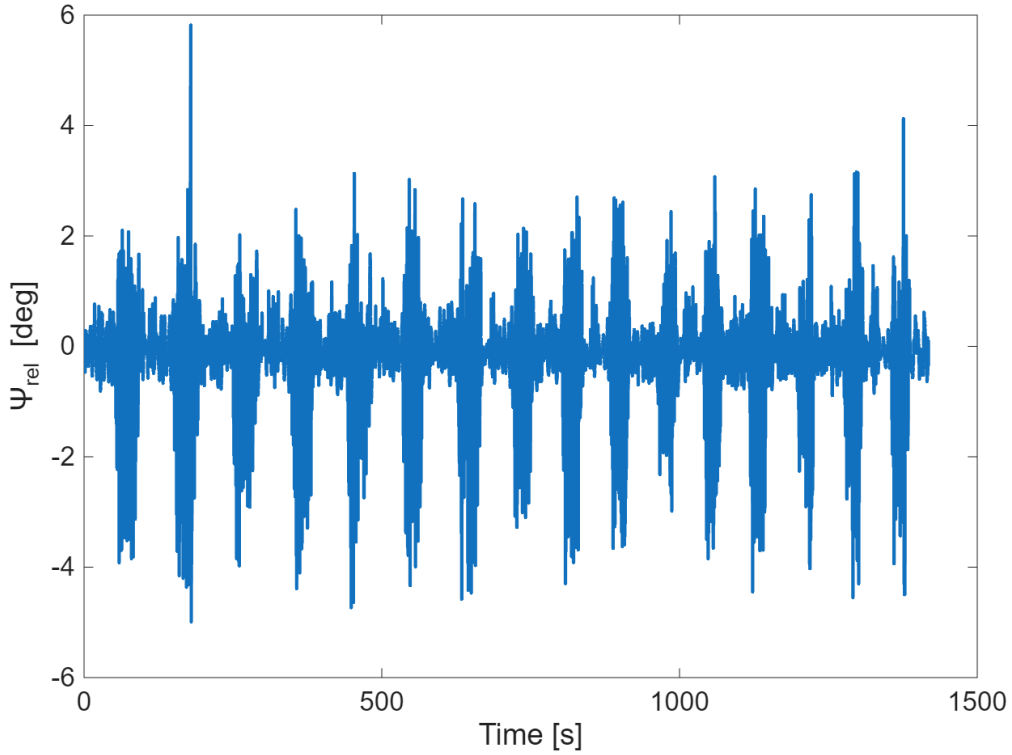

Figure B.4: Relative Heading

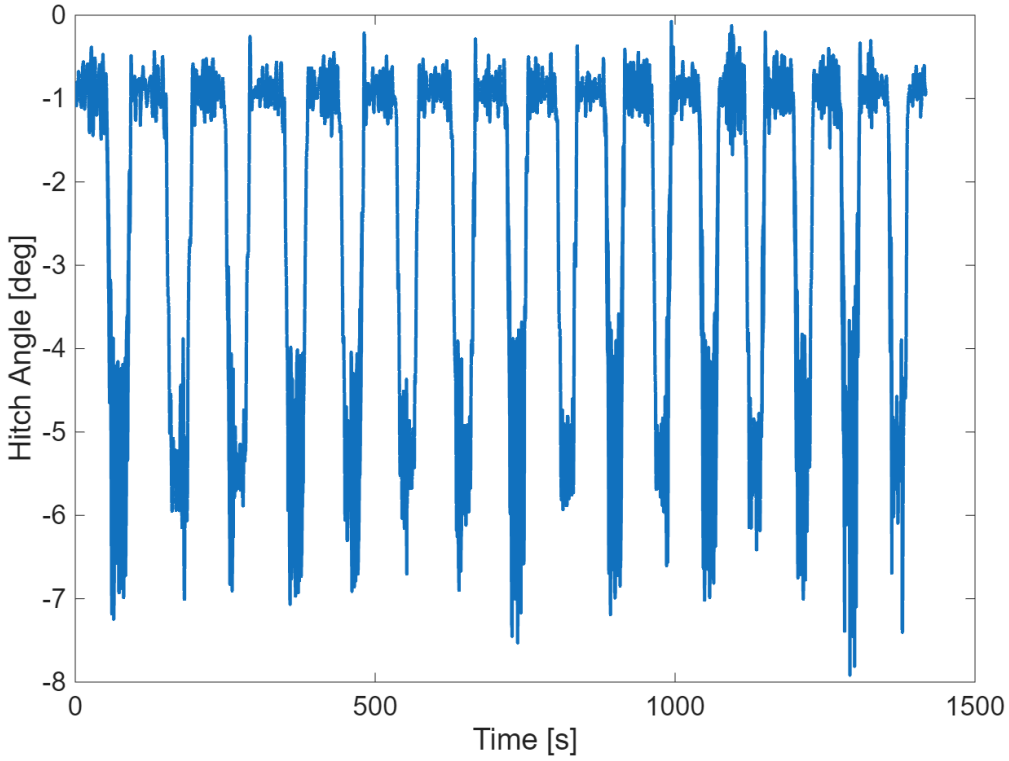

Figure B.5: Hitch Angle

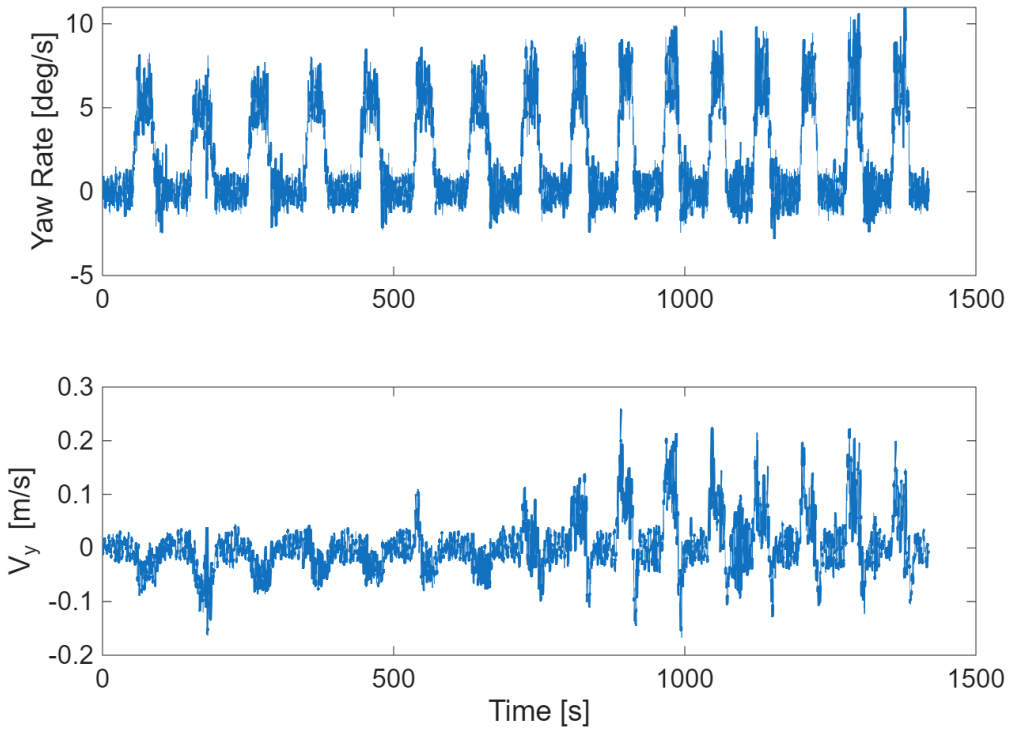

Figure B.6: Yaw Rate and Lateral Velocity