



**Politecnico
di Torino**

Technische
Universität
München



Department of Mechanical and Aerospace Engineering

Master of science program in AUTOMOTIVE ENGINEERING

Master of Science Thesis

Developing a parameter adaptive non-linear model predictive control for motion control of semi-trailer trucks

Supervisors:
Prof. Daniela Anna Misul
M.Sc. Baha Zarrouki

Candidate:
Mohsen Talebi

Politecnico di Torino & Technical University of Munich
February 2025

Acknowledgments

I wish to express my deepest gratitude to my supervisors, Prof. Daniela Anna Misul and M.Sc. Baha Zarrouki, for their invaluable guidance and unwavering support throughout the past year. Professor Misul has been exceptionally kind and encouraging, sharing her extensive knowledge in the field of control, which has significantly contributed to the development of this thesis. I am sincerely grateful for her insightful advice and continuous encouragement.

I would also like to extend my heartfelt appreciation to Mr. Zarrouki for giving me the opportunity to be part of the TUM-Control team and contribute to the development of the fascinating EDGAR project. Our open discussions and technical exchanges have been incredibly valuable, helping me not only deepen my expertise in control systems but also grow as a professional engineer. I truly appreciate his mentorship, which has had a profound impact on both my technical and professional development.

Special thanks go to Politecnico di Torino and the Department of Mechanical and Aerospace Engineering for providing such a well-structured and comprehensive master's program in Automotive Engineering. The wide variety of topics covered in this program has allowed me to expand my knowledge in the field and equipped me with essential skills for my future career.

Finally, I would like to express my heartfelt appreciation to my friends and family for their unwavering support, patience, and encouragement throughout this incredible journey. Their belief in me has been a source of strength and motivation, and I am truly grateful for their presence in my life.

**Mohsen Talebi,
February 2025.**

Abstract

The widespread use of semi-trailer trucks as a critical mode of freight transportation, owing to their high payload capacity and operational flexibility, necessitates advancements in motion control systems to enhance their safety, efficiency, and adaptability to dynamic road conditions. Model Predictive Control (MPC) is an advanced control strategy extensively employed in the field of motion control for autonomous vehicle guidance. The performance of the MPC method directly depends on how accurately the prediction model within the MPC can replicate real vehicle dynamics. Using a precise parameter set is crucial to achieving a reliable and robust controller, as any parameter mismatch between the controller and the real vehicle could lead to MPC failure. Therefore, for systems that must operate under a wide range of conditions and environments, it is essential to incorporate the ability to adapt working parameters in real time.

According to the literature, one widely used method for parameter estimation in complex nonlinear systems is Moving Horizon Estimation (MHE). However, implementing this method requires precision and careful consideration of various factors. This thesis proposes an adaptive Nonlinear Model Predictive Control (NMPC) approach for semi-trailer trucks to enhance the controller's performance in the presence of parameter mismatches.

First, a single-track dynamic model was developed based on the equations of motion for the semi-trailer truck. Subsequently, an adaptive controller integrating Nonlinear Model Predictive Control (NMPC) and Moving Horizon Estimation (MHE) was designed and tested. The results indicate that the proposed controller significantly enhances performance. In open-loop prediction, the Average Relative Distance Error (ARDE) metric improved by 80% to 93%. For closed-loop behavior, the average lateral deviation was reduced by at least 99% and up to 102% compared to a standard MPC controller without adaptivity. Furthermore, MHE proved valuable for state estimation, particularly when certain states were not directly measurable.

Contents

- List of Abbreviations III
- Formula Symbols V
- 1 Introduction 1**
 - 1.1 Motivation 1**
 - 1.2 Problem Definition 1**
 - 1.3 Outline..... 2**
- 2 Literature Review and State of the Art..... 3**
 - 2.1 Semi-Trailer Trucks 3**
 - 2.1.1 Introduction..... 3
 - 2.1.2 Dynamic Modeling Literature Review 4
 - 2.2 Parameter Mismatch State of the Art..... 4**
 - 2.2.1 Learning-based Approaches 7
 - 2.2.2 Model-Based Approaches 9
- 3 Theoretical Background 11**
 - 3.1 Dynamic Single-Track Model for Semi-Trucks 11**
 - 3.2 Model Predictive Control and Moving Horizon Estimation 16**
 - 3.2.1 Model Predictive Control 16
 - 3.2.2 Moving Horizon Estimation 17
 - 3.2.3 Application of MHE in Parameter Estimation..... 18
- 4 Methodology 20**
 - 4.1 Semi-Truck Prediction Model Development..... 20**
 - 4.2 Nonlinear Model Predictive Control for Semi-Trucks 22**
 - 4.2.1 Initialization..... 22
 - 4.2.2 Prediction 23
 - 4.2.3 Optimization Problem Solving..... 24
 - 4.2.4 Setting Control Command 27
 - 4.3 Moving Horizon Estimation for Semi-Trucks 27**
 - 4.3.1 Circular Data Buffer 29
 - 4.3.2 Data Preparation..... 29
 - 4.3.3 Recall Estimation Model..... 30
 - 4.3.4 Optimization Problem Solving..... 31
 - 4.3.5 Estimated Parameters Application..... 34
 - 4.4 Tools and Implementation 34**
- 5 Validation and Performance Evaluation 36**
 - 5.1 Prediction Model Validation 36**
 - 5.2 Simulation Environment and Test Track..... 37**
 - 5.3 Analyzing standard Nonlinear Model Predictive Control (NMPC) developed for the semi-truck..... 38**
 - 5.4 The Scenarios and Parameter Mismatch Problem..... 40**
 - 5.5 Moving Horizon Estimation (MHE) Evaluation 43**

5.5.1	Open-loop tests	46
5.5.2	Closed-loop tests	62
6	Conclusion and Outlook	72
	List of Figures	i
	List of Tables	iii
	Bibliography	iv

List of Abbreviations

ACADOS	Automatic Control And Dynamic Optimization Software
AGVs	autonomous ground vehicles
ARDE	Average Relative Distance Error
ARHAE	Average Relative Hitch Angle Error
ARLAVE	Average Relative Lateral Velocity Error
ARLOVE	Average Relative Longitudinal Velocity Error
AROE	Average Relative Orientation Error
ARSE	Average Relative State Error
ARYRE	Average Relative Yaw Rate Error
AVS	Autonomous Vehicle Systems
BiLSTM	Bidirectional Long Short-Term Memory
CasADi	Computer Algebra for Systems Analysis and Design Interface
CKF	Cubature Kalman Filter
DC	Direct Collocation
EDGAR	Excellent Drive GARching
EKF	Extended Kalman Filter
FNN	Feedforward Neural Network
GP	Gaussian Process
GPR	Gaussian Process Regression
GrBAL	Gradient-Based Adaptive Learning
IPM	Interior-Point Method
IRK4	Implicit Runge-Kutta 4th Order
KF	Kalman Filter
LCV	Long Combination Vehicle
Meta-RL	Meta Reinforcement Learning
MHE	Moving Horizon Estimation
MPC	Model Predictive Control
MSE	Mean Squared Error
NHTSA	National Highway Traffic Safety Administration
NLP	Non Linear Programming
NMPC	Nonlinear Model Predictive Control
NN	Neural Network
PEM	Prediction Model Error
ReBAL	Recurrence-Based Adaptive Learning
RL	Reinforcement Learning
RLS	Recursive Least Squares
RMPC	Robust Model Predictive Control
SCKF	Square-Root Cubature Kalman Filter

SMPC	Stochastic Model Predictive Control
SQP	Sequential Quadratic Programming
SQP-RTI	Sequential Quadratic Programming Real Time Iteration
TUM	Technical University of Munich
UKF	Unscented Kalman Filter

Formula Symbols

Formula Symbols	Unit	Description
$f_{\text{cont}}(\dot{x}, x, u)$	—	Continuous state-space model
x, x_k	—	Continuous state vector and state vector at discrete time k
u, u_k	—	Continuous control input and control input issued at discrete time k
x_{pos}	m	X-position of tractor's center of mass
y_{pos}	m	Y-position of tractor's center of mass
ψ	rad	Yaw angle of tractor
θ	rad	Hitch point angle
v_{long}	m s^{-1}	Longitudinal velocity of tractor's center of mass
v_{lat}	m s^{-1}	Lateral velocity of tractor's center of mass
$\dot{\psi}$	rad s^{-1}	Yaw rate of tractor
$\dot{\theta}$	rad s^{-1}	Hitch rate
δ_f	rad	Steering angle at the front wheel of tractor
a	m s^{-2}	Longitudinal acceleration
a_{lat}	m s^{-2}	Lateral acceleration
ω_f	rad s^{-1}	Steering rate at the front wheel of tractor
j	m s^{-3}	Longitudinal jerk acting on the driving axle
ψ_{trailer}	rad	Yaw angle of trailer
$x_{\text{pos,trailer}}$	m	X-position of trailer's center of mass
$y_{\text{pos,trailer}}$	m	Y-position of trailer's center of mass
$v_{\text{long,trailer}}$	m s^{-1}	Longitudinal velocity of trailer's center of mass
$v_{\text{lat,trailer}}$	m s^{-1}	Lateral velocity of trailer's center of mass
l_{f1}	m	Distance between the center of mass of the tractor section and it's front axle
l_{r1}	m	Distance between the center of mass of the tractor section and it's rear axle
l_{c1}	m	Distance between the center of mass of the tractor section and the hitch point

l_{c2}	m	Distance between the center of mass of the trailer section and the hitch point
l_{r2}	m	Distance between the center of mass of the trailer section and it's rear axle
m_1	kg	Mass of tractor
m_2	kg	Mass of trailer
I_{z1}	kg m ²	Moment of inertia of tractor
I_{z2}	kg m ²	Moment of inertia of trailer
ρ	kg m ⁻³	Air density
A	m ²	Frontal area of the semi-truck
c_d	—	Drag coefficient
$F_{x,aerodynamics}$	N	Longitudinal aerodynamic force
α_f	rad	Side slip angle of the tractor's front tire
α_{r1}	rad	Side slip angle of the tractor's rear tire
α_{r2}	rad	Side slip angle of the trailer's rear tire
C_f	N rad ⁻¹	Total cornering stiffness of the front axle tires of the tractor section
C_{r1}	N rad ⁻¹	Total cornering stiffness of the rear axle tires of the tractor section
C_{r2}	N rad ⁻¹	Total cornering stiffness of the rear axle tires of the trailer section
$F_{y,f}$	N	Lateral tire force of front axle tires of the tractor section
$F_{y,r1}$	N	Lateral tire force of rear axle tires of the tractor section
$F_{y,r2}$	N	Lateral tire force of rear axle tires of the trailer section
$F_{x,f}$	N	Longitudinal tire force of front axle tires of the tractor section
$F_{x,r1}$	N	Longitudinal tire force of rear axle tires of the tractor section
$F_{x,r2}$	N	Longitudinal tire force of rear axle tires of the trailer section
$F_{r,f}$	N	Rolling resistance force of front axle tires of the tractor section
$F_{r,r1}$	N	Rolling resistance force of rear axle tires of the tractor section
$F_{r,r2}$	N	Rolling resistance force of rear axle tires of the trailer section
F_d	N	Driving force acting on the driving axle
$f_{tractor}$	—	Rolling resistance coefficient for the tractor section
$f_{trailer}$	—	Rolling resistance coefficient for the trailer section
$F_{z,f}$	N	Vertical load acting on the front axle of the tractor section
$F_{z,r1}$	N	Vertical load acting on the rear axle of the tractor section
$F_{z,r2}$	N	Vertical load acting on the rear axle of the trailer section
g	m.s ⁻²	Gravitational acceleration
p, p_k	—	Continuous parameter vector and parameter vector at discrete time k

T_s	s	Sampling time
k	—	Discrete time index
\tilde{x}_k	—	Prediction of x_k
γ	—	Number of steps in multi-step prediction
$X_{k_1:k_2}$	—	Series of discrete state vectors starting from k_1 and including k_2
$U_{k_1:k_2}$	—	Series of discrete control inputs starting from k_1 and including k_2
$P_{k_1:k_2}$	—	Series of parameter vectors starting from k_1 and including k_2
J	—	Cost function
N	—	Prediction horizon steps
Q	—	State weighting matrix
R	—	Input command weighting matrix
i	—	Vector index
$x_{\text{ref},k}$	—	Reference (desired) state vector
$u_{\text{ref},k}$	—	Reference (desired) input command vector
$x_{\text{min}}, x_{\text{max}}$	—	Lower and Upper state limits
$u_{\text{min}}, u_{\text{max}}$	—	Lower and Upper input command limits
w_k	—	process noise vector at time k
v_k	—	measurement noise vector at time k
σ_w	—	Standard deviation of w_k
σ_v	—	Standard deviation of v_k
\hat{x}_k	—	Estimation of x_k
N_e	—	Estimation horizon steps
Q_w	—	Process noise weighting matrix
Q_v	—	Measurement noise weighting matrix
\bar{x}_k	—	Augmented state vector at discrete time k
$\hat{\bar{x}}_k$	—	Estimation of \bar{x}_k
T	s	Prediction horizon
T_d	s	Discretization time
J_{nmpc}	—	NMPC cost function
Q_e	—	Terminal state weighting matrix
ξ^-, ξ^+	—	Lower and Upper bounds of slack variables
$u_{\text{opt},k}$	—	optimal control input at discrete time k
x_{est}	—	Estimation model state vector

$X_{\text{est},k_1:k_2}$	—	Series of discrete estimation model state vectors starting from k_1 and including k_2
$\hat{X}_{\text{est},k_1:k_2}$	—	Series of discrete estimated state vectors starting from k_1 and including k_2
$x_{\text{buff},k}$	—	Buffer state vector at discrete time k
$X_{\text{buff},k_1:k_2}$	—	Series of discrete buffer state vectors starting from k_1 and including k_2
$U_{\text{buff},k_1:k_2}$	—	Series of discrete buffer input vectors starting from k_1 and including k_2
$P_{\text{buff},k_1:k_2}$	—	Series of discrete buffer parameters vectors starting from k_1 and including k_2
J_{mhe}	—	MHE cost function
J_0, J_i, J_e	—	initial stage, intermediate stage and final stage of MHE cost function
$Q_{\text{mhe}0}$	—	previously estimated states weighting matrix
y_0, y_i, y_e	—	target optimization vectors at initial stage, intermediate stage and final stage of MHE cost function
$y_{\text{ref}0}, y_{\text{ref}i}, y_{\text{ref}e}$	—	reference vectors at initial stage, intermediate stage and final stage of MHE cost function
W_0, W, W_e	—	Weighting matrices at initial stage, intermediate stage and final stage of MHE cost function
T_{est}	s	Estimation horizon
T_e	s	Estimator discretization time
ϵ_k	—	Average Relative State Error (ARSE) for discrete time k
$\epsilon_{D,k}$	—	Average Relative Distance Error (ARDE) for discrete time k
I	%	Relative improvement
$I_{\text{path-tracking}}$	%	Relative improvement for closed-loop path tracking

1 Introduction

1.1 Motivation

Transportation systems play a vital role in modern society, serving as essential mechanisms for the movement of both goods and people in response to diverse demands [1]. Semi-trailer articulated vehicles are a common means of transporting industrial products across various industries worldwide. Their significance is underscored by the expanding semi-trailer truck market, which is projected to surpass USD 41.4 billion by 2032, driven by technological advancements and the growing demand for sustainable transport solutions [2].

The automotive industry is transitioning into a new era of autonomous ground vehicles (AGVs), which have the potential to reduce human errors in accidents, thereby significantly enhancing the safety of passengers, drivers, and pedestrians while also lowering financial costs [3]. As a result, autonomous driving has emerged as a significant area of study, attracting billions of dollars in investment and extensive research from both industry and academic institutions. More recently, end-to-end driving has started to emerge as an alternative to modular approaches, although numerous challenges remain to be solved to fully realize this future [4].

For this reason, the Chair of Automotive Engineering at Technical University of Munich (TUM) developed an autonomous research vehicle, Excellent Drive GARching (EDGAR), to contribute to advancements in autonomous driving research. They also created a platform called TUM-Control, which serves as the foundation for the work presented in this thesis [5].

1.2 Problem Definition

Autonomous vehicle systems have become a prominent area of research. The technical readiness of self-driving vehicles is steadily advancing, driven by increased computing power and decreasing sensor costs. Regardless of the type of vehicle, a self-driving system's core comprises three main components: perception, planning, and control. First, the vehicle recognizes its surrounding environment and its own status using various onboard sensors. It then analyzes this data to create a trajectory through advanced planning algorithms, ultimately controlling the vehicle's movement autonomously [6]. This thesis focuses on the motion control component.

In automated driving systems, the path tracking layer defines the actuator commands required to follow the reference path and speed profile. Model Predictive Control (MPC) is widely used for trajectory tracking due to its ability to handle multi-variable problems, systematically account for constraints on states and control actions, and consider the expected future behavior of the system [7]. For these reasons, MPC is used as the main controller in this study.

The basic MPC algorithm computes an optimal control action, u , at each discrete time step, k , by solving an optimization problem [8]. While the underlying prediction model and cost function formulation are the primary factors influencing MPC performance, other elements, such as cost function weights, prediction horizon length, and external disturbances, also significantly impact its effectiveness.

However, system dynamics can change during runtime due to varying environmental conditions or uncertainties in the model design phase. These changes can lead to challenges such as undefined models or parameter mismatches. In such cases, the underlying process model cannot be fully predetermined or identified in advance, as it is not static. In the context of autonomous vehicle motion control, these issues may arise from a lack of knowledge about the current load (e.g., mass and its distribution), inertia, or variations in road-tire friction. These factors directly affect driving dynamics and can significantly influence the performance of the implemented controller.

On the other hand, since such model mismatches and varying environmental conditions exhibit high statistical variance, treating them as pure white noise is inefficient. Therefore, a method is needed that enables the controller to adapt online, as parameter uncertainties can evolve over time.

This study aims to address the challenge of adapting NMPC, specifically developed for semi-trucks, in the presence of parameter mismatches or varying environmental conditions that cannot be characterized as Gaussian-distributed disturbances. The proposed solution seeks to enhance the controller's prediction quality, which is critical for selecting more accurate input commands. Furthermore, the method will be tested under light disturbances to evaluate its robustness.

1.3 Outline

This thesis is organized as follows: Chapter 2 provides a comprehensive literature review on semi-truck dynamics modeling and examines state-of-the-art approaches to parameter adaptivity in controllers. Chapter 3 covers the theoretical foundations of the study, including the dynamic equations of motion for semi-trailer tractors, MPC, and Moving Horizon Estimation (MHE). Chapter 4 outlines the methodology and procedures used to develop an adaptive NMPC system for semi-trucks. Chapter 5 presents the validation and evaluation results, accompanied by an in-depth analysis. Finally, Chapter 6 summarizes the key findings and proposes directions for future research.

2 Literature Review and State of the Art

2.1 Semi-Trailer Trucks

2.1.1 Introduction

Long Combination Vehicles (LCVs) are an essential component of modern freight transportation systems, offering significant advantages in terms of efficiency and cargo capacity. Among the various types of LCVs, semi-trailer trucks play a crucial role. A semi-trailer truck consists of a tractor unit (also called a truck or prime mover) and a semi-trailer, where the latter has only one end supported by the truck's rear axle, while the front is typically supported by a fifth wheel mounted on the tractor. This design allows semi-trailers to carry large and heavy cargo, making them especially important in the transportation of goods over long distances [9].

The importance of semi-trailer trucks in freight transport cannot be overstated. They are capable of carrying a wide variety of goods, ranging from perishable items to machinery and raw materials, and they form the backbone of freight logistics worldwide. In fact, they account for a substantial share of freight transport in countries with well-developed infrastructure [10]. Additionally, semi-trailer trucks are designed for high maneuverability, which makes them suitable for navigating urban areas and narrow roads, a critical factor in delivery logistics [11]. Figure 2.1 presents a simplified schematic of a semi-trailer truck.

The modular concept is a way of building vehicle combinations with the help of different types of trailers and trucks. The types of trailers are usually full trailers, semi-trailers, link trailers, converter dollies, and center-axle trailers, while the trucks typically include rigid trucks and tractors. These different components of road trains are designed to provide the flexibility needed to meet varying transportation needs. Figure 2.2 illustrates these road train components in a detailed table, showing the diversity and inter-connectivity of the parts that form LCVs, highlighting their adaptability to different freight requirements [12].

While semi-trailer trucks are indispensable for freight transportation, their operation is not without hazards. Due to their large size, heavy weight, and complex dynamics, these vehicles pose significant risks on the road, particularly in terms of accidents. According to studies, semi-trailer trucks are involved in a disproportionate number of road accidents compared to smaller vehicles. The National Highway Traffic Safety Administration (NHTSA) highlights that in the United States alone, large trucks were involved in nearly 9% of all fatal crashes, despite representing only 4% of registered vehicles [13]. Key hazards associated with semi-trailer trucks include their longer braking distances, reduced maneuverability, and susceptibility to rollovers, jackknifing, and side-swipe collisions. These hazards are exacerbated by adverse weather conditions, driver fatigue, and high-speed operation, which significantly increase the likelihood of accidents [14].

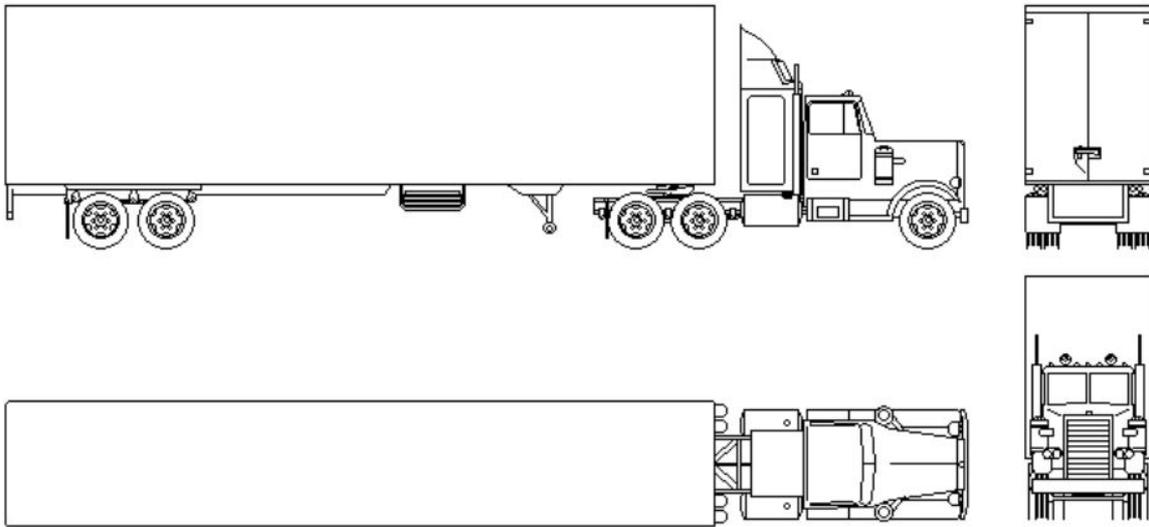


Figure 2.1: Semi-truck simplified schematic

2.1.2 Dynamic Modeling Literature Review

Given the significant safety risks associated with semi-trailer trucks, the development of advanced motion control algorithms has become essential for enhancing road safety, with vehicle dynamics modeling serving as a core component of this effort. Vehicle models are divided into two categories: kinematic models and dynamic models [15].

Table 2.1 summarizes the most recent works utilizing various semi-truck vehicle models and their objectives. Note that in the table, KM refers to the Kinematic Model, LDM to the Lateral Dynamic Model (excluding longitudinal equations of motion), and FDM to the Full Dynamic Model.

Based on the studies listed in Table 2.1, it can be observed that both kinematic and dynamic models have been utilized in previous research. However, for tasks such as path tracking and ensuring directional stability, particularly at high speeds, the use of a full dynamic model is essential. This approach has been adopted in the current study and will be discussed in detail in the following chapters.

2.2 Parameter Mismatch State of the Art

Parameter mismatch in vehicle motion control refers to discrepancies between the assumed parameters in a control model and the actual parameters of the vehicle. These mismatches can significantly impact control performance, leading to issues such as steady-state errors, reduced tracking accuracy, and compromised stability.

However, this issue can often be adequately addressed by employing a suitable control algorithm. In the case of MPC, this typically involves the use of Robust Model Predictive Control (RMPC) or Stochastic Model Predictive Control (SMPC). Both approaches have demonstrated their effectiveness in successfully accounting for uncertainties [16, 17]. Nevertheless, both RMPC and SMPC rely on assumptions about the distribution of the residual uncertainties to ensure feasibility and constraint satisfaction [18].

Rigid Truck	Towing vehicles with cargo space 
Tractor	Towing vehicles without their own cargo space. It can be connected to a trailer 
Semi-Trailer	Trailer with axles at the rear and coupling pin (Hitch point) at the front 
Full-Trailer	Trailer with axles both at the back and at the front. 
Link-Trailer	link-trailer with turntable at the rear for connection of an additional semi-trailer 
Center-axle Trailer	Drawbar trailer, and the axles centered roughly around the center of the cargo area. 
Converter Dolly	Trailer without its own cargo space. It has a drawbar at the front and a turntable over the axles for connecting a trailer. 
Vehicle Combination	A towing vehicle with a number of attached trailers. Example Truck with double trailers, type AB double 

Figure 2.2: Components of vehicle combination

In RMPC, uncertainties are assumed to belong to a compact uncertainty set, and the algorithm calculates control inputs that ensure constraint satisfaction for all possible values within this set, prioritizing robustness. SMPC, on the other hand, treats uncertainties as probabilistic, relying on estimated probability distributions. It calculates optimal control inputs that satisfy constraints with a high probability, balancing performance and stability. The choice of uncertainty assumptions is crucial for both approaches. Large uncertainty sets or high variances often lead to overly conservative behavior, reduced performance, or infeasibility. On the other hand, small sets or tight variances may overlook critical uncertainties, risking constraint violations, and instability. Thus, RMPC or SMPC can only guarantee near-optimal performance and constraint satisfaction if the true uncertainty set (or the true statistical variance) is provided or can be assumed to be negligible [16, 17].

In our case, we encounter large and sometimes consistent changes in vehicle parameters or environmental conditions, which makes the previously mentioned methods unfeasible. Therefore, instead of relying on

Table 2.1: Literature review of semi-truck's different vehicle models

Reference	year	KM	LDM	FDM	Aims
[19]	2017	✓		✓	<ul style="list-style-type: none"> • Developing a composite path tracking control strategy for under-actuated tractor-trailer vehicles. • Combining MPC and adaptive fuzzy-based dynamic control.
[20]	2017	✓			<ul style="list-style-type: none"> • Designing a MPC for a tractor-trailer system to improve path tracking performance. • Using an augmented kinematic model with control delay, the controller tries to minimize tracking errors.
[21]	2021		✓		<ul style="list-style-type: none"> • Developing an advanced path-tracking strategy for articulated vehicles with a trailer. • Combining MPC and optimal curvature preview control (OCP) techniques.
[22]	2020	✓	✓		<ul style="list-style-type: none"> • Develop a unified lane-keeping and forward collision avoidance system for semi-trailer trucks. • Employing a hitch angle estimation algorithm and a MPC with separate constraints for lane-keeping and collision avoidance.
[14]	2022	✓			<ul style="list-style-type: none"> • Designing a nonlinear model predictive control (NMPC) local planner for autonomous tractor-trailer vehicles. • Addressing challenges such as jackknifing, obstacle avoidance, and path constraints.
[23]	2020	✓	✓		<ul style="list-style-type: none"> • Investigating the modeling and lateral control of tractor-trailer vehicles during aggressive maneuvers. • Proposing and comparing lateral controllers based on MPC and Linear-Quadratic (LQ) techniques.
[24]	2012		✓		<ul style="list-style-type: none"> • Developing a method for assessing the dynamic performance of truck and trailer combinations by using linear frequency domain models to analyze stability and rearward amplification (RA) characteristics
[25]	2012		✓		<ul style="list-style-type: none"> • Proposing a closed-loop dynamic simulation-based design method for articulated heavy vehicles (AHVs) with active trailer steering (ATS) systems.
[26]	2019		✓		<ul style="list-style-type: none"> • Addressing the trade-off between maneuverability at low speeds and stability at high speeds. • Developing and evaluate an adaptive control system for autonomous tractor-trailer vehicles, focusing on improving lane-keeping performance with using an LQR error state controller and an LQR-MRAC controller.
[27]	2014			✓	<ul style="list-style-type: none"> • Developing single-track models for an A-double vehicle combination, focusing on high-speed cornering. • Using simplified equations of motion derived through Lagrangian formalism.
[28]	2019			✓	<ul style="list-style-type: none"> • Design trajectory planners for a truck with a semi-trailer, focused on enabling the vehicle to safely navigate sharp corners and roundabouts at low velocities, using MPC
[29] main reference	2020			✓	<ul style="list-style-type: none"> • Investigating the directional instability of a truck and trailer combination. • Designing two control strategies, an integral-plus-state feedback controller and a sliding mode controller.

control algorithms that use static models, an adaptive solution is necessary. To achieve this goal, several methods have been developed, which are summarized in this section.

These methods can be broadly categorized into two main approaches: learning-based and model-based. While this classification is not rigid, it provides a useful framework for understanding the methodologies.

2.2.1 Learning-based Approaches

This set of approaches can be further divided into three subclasses: Fully Data-Driven Learning, Unmodeled Dynamic Learning, and Parameter Learning (or Parameter Estimation).

Fully Data-Driven Learning

These methods are capable of handling a wide range of environmental condition changes that fall within the design domain. Also, an explicit adaptation process can be omitted, as environmental changes are implicitly accounted for by processing online-generated data based on the current vehicle behavior.

In [30], a Neural Network (NN) is employed to address the problem of precisely predicting vehicle dynamics at the edge of tire saturation across a broad range of road conditions, with a focus on tire-ground friction. The NN predicts yaw and lateral accelerations based on current and past states, enabling a control algorithm to generate optimal driving commands. Trained on diverse surface data, it implicitly learns environmental conditions, achieving race-car-driver-level performance. Compared to a static dynamics model, the NN excels under varying and constant conditions due to its ability to handle complex dynamics. Although this approach demonstrates clear improvements over a simple static model, some challenges remain. First, it is highly dependent on the quality and quantity of data. Furthermore, it is not evident that a neural network would outperform a refined vehicle dynamics model with adaptive parameters. Both aim to approximate real driving dynamics, but neural networks do not utilize the inherent mechanical knowledge of vehicle behavior.

Another fully data-driven model structure is Gaussian Process Regression (GPR), as used in [31]. GPR is a powerful machine learning method used for making predictions, especially useful when dealing with uncertainties and complex behaviors in systems. This approach is employed due to the significant cost and challenges associated with developing an accurate plant model, as well as its ability to reduce conservatism by quantifying residual modeling uncertainty and continuously refining the model. This approach excels in providing and continuously updating an accurate plant model while reducing the conservatism of the controller. However, the adaptation process prioritizes model refinement over rapid adaptation to quickly changing parameters. The Bayesian nature of Gaussian Process (GP) updates can intensify this issue. Moreover, the primary motivation for using GP, addressing the difficulty or expense of developing an accurate model, is less relevant in the context of vehicle dynamics, where a wide range of established models are available.

Unmodeled Dynamic Learning

Unmodeled dynamic learning methods are utilized for tasks involving model components or precise structures that are unknown beforehand or difficult to incorporate into a single model. In such cases, the precise dynamics of the system are discovered and adapted online.

A common refinement technique, as proposed in [32] and [33], involves introducing a two-part state-space model, expressed in a form similar to:

$$x_{k+1} = f(x_k, u_k) + g(x_k, u_k) \quad (2.1)$$

Here, the right-hand side is a sum of one term representing a simple approximation of the underlying physics, f , and a second term representing the model error, g . This second term is adapted online to approximate the current, true dynamics.

In [32], the error term is constructed from nonlinear basis functions that are pre-trained on diverse tasks and combined online using a Kalman Filter (KF). This approach addresses the variability and unpredictability of

robotic tasks, where complete dynamics models with adaptive parameters often struggle. While effective in correcting model mismatches and representing unknown system components, the method lacks stability guarantees and complicates the analysis of predictions, stability, and constraint satisfaction. Its strength lies in adaptability to unknown tasks, but this advantage diminishes if an adaptive dynamics model can fully capture task diversity. Nevertheless, unmodeled effects and mismatches may persist, which this approach can help mitigate, though adaptation may require time and exploration.

The error term, g , can also be handled using GPs, as demonstrated in [33]. The primary objective of this approach is to enhance a vehicle dynamics model integrated into an MPC algorithm to mitigate systematic disturbances. In [33], the authors decouple safety and performance improvements by introducing double prediction sequences: one based on the prior nominal model and the other derived from the learned model obtained through GPR compensation. This method has demonstrated its effectiveness in addressing the safety-performance trade-off while offering provable safety guarantees. However, the proposed scheme retains the significant computational burden of traditional NMPC and remains conservative in its representation of uncertainty bounds.

Another approach for error term derivation is Reinforcement Learning (RL), with a particular focus on Meta Reinforcement Learning (Meta-RL). In [34], the challenge of adapting to dynamic, real-world environments with unexpected perturbations and novel situations is addressed. Recognizing the impracticality of training separate policies for every possible scenario an agent may encounter in the real world, this work proposes a method for learning to quickly and effectively adapt online to new tasks. This approach combines a pre-trained prior model and hyper-parameters, trained jointly via RL, to enable fast adaptation to new tasks. The prior model, a neural network describing system dynamics, is adapted online using Gradient-Based Adaptive Learning (GrBAL) or Recurrence-Based Adaptive Learning (ReBAL) policies, optimizing prediction performance over future time steps. While effective for dynamic environments and diverse tasks, challenges remain in ensuring stability, constraint satisfaction, and prediction accuracy due to the changing model structure. The method excels when unmodeled effects or mismatches exist but loses its advantage if a single adaptive model can cover the entire design domain.

Overall, the primary drawbacks of unmodeled dynamic learning approaches include the difficulty in analyzing their behavior and ensuring accurate predictions. However, these methods offer significant flexibility and the ability to account for unknown or unmodeled effects. They can be optimized to provide proactive, rather than merely reactive, predictions. Computational efficiency and feasibility can be enhanced through pre-training and the use of explicit algorithms, such as Feedforward Neural Networks (FNNs), during runtime.

Parameter Learning (Parameter Estimation)

One idea for estimating the model parameters at runtime is the use of a neural network, as done in [35]. In the context of tractor-semitrailer systems, accurately estimating vehicle mass remains a significant challenge. This difficulty stems from traditional methods being hindered by model uncertainty, model mismatch, and insufficient training data, as well as a lack of algorithm generalization to accommodate real-world traffic scenarios. To overcome these limitations, a hybrid algorithm for vehicle mass estimation, integrating Bidirectional Long Short-Term Memory (BiLSTM) networks and a Square-Root Cubature Kalman Filter (SCKF), is proposed. Utilizing vehicle time-series data, the BiLSTM-based mass estimator is designed to address model uncertainty effectively. To further improve estimation accuracy and robustness, the hybrid BiLSTM-SCKF method is developed, combining the strengths of data-driven learning and probabilistic filtering. The experimental results in real-world environments show that the hybrid BiLSTM-SCKF algorithm outperforms single algorithms, particularly under medium and low loads.

The research gap of the explained study, in relation to our objectives, lies in the fact that, while their parameter estimation approach is conducted in real time, the estimated parameters are not integrated into an MPC framework. This limits its applicability for adaptive control in dynamic environments, where real-time parameter updates are critical for enhanced system performance.

2.2.2 Model-Based Approaches

Model-based approaches typically utilize vehicle dynamic and kinematic models for parameter estimation. Methods such as Recursive Least Squares (RLS) and KF algorithms are commonly employed to estimate vehicle mass by leveraging the correlation between the driving force and longitudinal acceleration. However, these approaches often suffer from significant errors when measurement noise is present. To address this limitation, nonlinear KF methods—such as the Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), and Cubature Kalman Filter (CKF)—have been widely adopted for vehicle mass estimation. These nonlinear methods are favored due to their relatively simple computation and fast convergence, making them particularly effective for complex, nonlinear dynamical systems [35, 36]. In [37], the offset-free MPC control method is proposed to systematically address the steady-state error problem. The core concept of this approach is to treat model mismatches, control input offsets, and external disturbances as disturbance terms. These disturbances are then observed using filters, allowing their effects to be mitigated during the MPC solution stage. In their study, the KF is utilized as the disturbance observer and integrated into the newly designed MPC solver to ensure effective elimination of steady-state errors. This study focuses solely on steady-state error elimination and is limited in its applicability to highly dynamic maneuvers, making it unsuitable for scenarios that require real-time adaptation.

Additionally, two other model-based adaptation methods, Prediction Model Error (PEM) and MHE, show significant potential for this study, which will be discussed below.

Prediction Error Method (PEM)

PEMs, a class of parameter estimation techniques, aim to reduce the difference between the observed and predicted states by optimizing the parameters of the prediction model [38]. Thus, an estimated parameter \hat{p} at time step N can be expressed as:

$$\hat{p} = \arg \min_p J(p) \quad \text{with} \quad J(p) = \sum_{k=1}^N d(x_k - \tilde{x}_k(x_{k-1}, u_{k-1}, p)) \quad (2.2)$$

where J is the cost function and d is a suitable distance measure. The term $\tilde{x}_k(x_{k-1}, u_{k-1}, p)$ represents a one-step prediction of the state x_k , based on the unknown parameters p and the previous control input u_{k-1} . This process can be naturally applied to a recent subset of the collected data, focusing on the most relevant or recent observations for parameter estimation [38, 39].

PEMs differ in error measures and minimization techniques. One of the simplest PEMs uses the squared 2-norm of the difference between observed and predicted states as the error function, which can be minimized via gradient descent or the Gauss-Newton method. Maximum likelihood estimation handles state transitions with additive noise and simplifies to a logarithmic error measure for fully observable states, enabling minimization while integrating knowledge of state transitions and model accuracy. PEMs are versatile and effective for accurate static parameter estimation across various model structures, supported by advanced minimization algorithms. However, they face challenges such as convergence issues, non-convex optimization, high computational demands, and dependence on good initial guesses. PEMs are unsuitable for dynamic parameters, as they provide batch averages rather than future-accurate estimates. This leads to a trade-off between larger batches for precision and smaller batches for timeliness [38].

Moving Horizon Estimation (MHE)

Another model-based method, similar to MPC in its approach of minimizing the discrepancy between predicted and observed data, is MHE. Like MPC, MHE optimizes a cost function based on observed system inputs and outputs over a moving time horizon to estimate unknown components of the state and parameters. This approach can accommodate various types of constraints and handle random disturbances effectively. It can be employed for state estimation, joint state-parameter estimation in dynamical systems, or purely for parameter estimation [40, 41].

Compared to PEM, MHE offers more diverse optimization goals but behaves similarly when the state is fully observable. It provides accurate, robust estimates, supports constraints, and is commonly used in real-time applications. However, MHE struggles with dynamic parameters, requiring the moving horizon to fully adapt after sudden changes, leading to a similar trade-off between accuracy and timeliness. Additional challenges include selecting hyper-parameters, handling arrival cost, feasibility issues, and high computational demands. MHE is primarily suited for state or joint state and parameter estimation [42, 43].

To conclude, model-based approaches, particularly parameter estimation methods, are inherently limited by their fixed model structure, which restricts their ability to capture unmodeled effects. Additionally, methods such as PEMs and MHE, which rely on solving optimization problems, often face the computational challenges associated with non-convex optimization. However, these approaches offer a clear and well-understood model structure with parameters that typically have physical significance. This allows for the imposition of constraints and the analysis of system behavior to ensure accurate and realistic predictions. Moreover, these methods have already been successfully implemented in various applications.

Building upon the knowledge of adaptation methods provided above, this study employs the model-based approach of MHE to develop an adaptive NMPC framework. This approach aims to address model parameter mismatches in the motion control of semi-trucks effectively.

3 Theoretical Background

3.1 Dynamic Single-Track Model for Semi-Trucks

Accurate vehicle dynamics modeling is crucial for precise vehicle control. Consequently, the corresponding differential equations of motion must satisfy both practical and analytical requirements. On the one hand, the model must be computationally efficient enough to enable real-time execution on typical onboard computers. On the other hand, it must adequately represent the key physical components that govern motion control [29]. The single-track model, also known as the bicycle model, is a widely adopted simplification in vehicle control systems utilizing MPC. This model represents the vehicle dynamics by approximating it with a single front tire and a single rear tire, significantly reducing complexity. Single-track models are generally divided into two categories: kinematic models and dynamic models [15]. While both are essential tools in automotive engineering, they differ fundamentally in complexity, use cases, and the level of detail they provide (see Table 3.1). Consequently, as noted in the previous chapter, a dynamic single-track model for a semi-truck, as

Table 3.1: Kinematic vs Dynamic vehicle model comparison[44, 45]

Feature	Kinematic Model	Dynamic Model
Complexity	Simple and lightweight	Complex and computationally intensive
Physics Involved	Geometric relationships only	Includes forces, inertia, and tire dynamics
Accuracy	Accurate at low speeds	Accurate at all speeds, especially high speeds
Use Cases	Path planning, low-speed control	High-speed control, stability analysis

outlined in [29], with minor modifications, is employed throughout this thesis. It is represented as a continuous state-space model, $0 = f_{\text{cont}}(\dot{x}, x, u)$. The state vector

$$\mathbf{x} = [x_{\text{pos}} \quad y_{\text{pos}} \quad \psi \quad \theta \quad v_{\text{long}} \quad v_{\text{lat}} \quad \dot{\psi} \quad \dot{\theta} \quad \delta_f \quad a]^T \quad (3.1)$$

consists of ten states. x_{pos} , y_{pos} , and ψ represent the position of the tractor section's center of mass and its yaw angle, respectively. θ denotes the articulation point (also known as the hitch point) angle, while v_{long} and v_{lat} describe the longitudinal and lateral velocities of the tractor section's center of mass. $\dot{\psi}$ represents the yaw rate of the tractor section, and $\dot{\theta}$ is the hitch rate. δ_f refers to the steering angle, and a is the longitudinal acceleration. By considering the first eight states, the position and velocity of the semi-trailer's center of mass can also be derived using the following equations.

$$\psi_{\text{trailer}} = \psi - \theta \quad (3.2)$$

$$x_{\text{pos,trailer}} = x_{\text{pos}} - l_{c1} \cos(\psi) - l_{c2} \cos(\psi - \theta) \quad (3.3)$$

$$y_{\text{pos,trailer}} = y_{\text{pos}} - l_{c1} \sin(\psi) - l_{c2} \sin(\psi - \theta) \quad (3.4)$$

$$v_{\text{long,trailer}} = v_{\text{long}} \cos(\theta) - (v_{\text{lat}} - l_{c1} \dot{\psi}) \sin(\theta) \quad (3.5)$$

$$v_{\text{lat,trailer}} = (v_{\text{lat}} - l_{c1} \dot{\psi}) \cos(\theta) + v_{\text{long}} \sin(\theta) - l_{c2} (\dot{\psi} - \dot{\theta}) \quad (3.6)$$

The input vector

$$\mathbf{u} = [j \quad \omega_f]^T \quad (3.7)$$

consists of two control inputs: the steering rate at the front wheel of the tractor section, ω_f and the longitudinal jerk, j . The underlying differential equations describing the system dynamics are

$$0 = -\dot{x}_{\text{pos}} + v_{\text{long}} \cos(\psi) - v_{\text{lat}} \sin(\psi) \quad (3.8)$$

$$0 = -\dot{y}_{\text{pos}} + v_{\text{long}} \sin(\psi) + v_{\text{lat}} \cos(\psi) \quad (3.9)$$

$$0 = -\dot{\psi}_{\text{derivative of state}} + \dot{\psi} \quad (3.10)$$

$$0 = -\dot{\theta}_{\text{derivative of state}} + \dot{\theta} \quad (3.11)$$

$$\begin{aligned} 0 = & (m_1 + m_2)(\dot{v}_{\text{long}} - \dot{\psi} v_{\text{lat}}) + m_2(l_{c2}((\ddot{\psi} - \ddot{\theta}) \sin(\theta) \\ & + (\dot{\psi} - \dot{\theta})^2 \cos(\theta)) + l_{c1} \dot{\psi}^2) \\ & - F_{x,f} \cos(\delta_f) + F_{y,f} \sin(\delta_f) - F_{x,r1} \\ & - F_{x,r2} \cos(\theta) - F_{y,r2} \sin(\theta) + F_{x,\text{aerodynamics}} \end{aligned} \quad (3.12)$$

$$\begin{aligned} 0 = & (m_1 + m_2)(\dot{v}_{\text{lat}} + \dot{\psi} v_{\text{long}}) - m_2(l_{c2}((\ddot{\psi} - \ddot{\theta}) \cos(\theta) \\ & + (\dot{\psi} - \dot{\theta})^2 \sin(\theta)) + l_{c1} \ddot{\psi}) \\ & - F_{x,f} \sin(\delta_f) - F_{y,f} \cos(\delta_f) \\ & - F_{y,r1} + F_{x,r2} \sin(\theta) - F_{y,r2} \cos(\theta) \end{aligned} \quad (3.13)$$

$$\begin{aligned} 0 = & (I_{z1} + (m_2 l_{c1}^2)) \ddot{\psi} - m_2 l_{c1} (\dot{v}_{\text{lat}} + (\dot{\psi} v_{\text{long}})) + m_2 l_{c1} l_{c2} (\cos(\theta) (\ddot{\psi} - \ddot{\theta}) \\ & + \sin(\theta) (\dot{\psi} - \dot{\theta})^2) - l_{f1} \sin(\delta_f) F_{x,f} - l_{f1} \cos(\delta_f) F_{y,f} + l_{r1} F_{y,r1} \\ & - F_{x,r2} l_{c1} \sin(\theta) + F_{y,r2} l_{c1} \cos(\theta) \end{aligned} \quad (3.14)$$

$$\begin{aligned} 0 = & (I_{z2} + m_2 l_{c2} (l_{c2} - (l_{c1} \cos(\theta)))) \ddot{\theta} + (I_{z1} - I_{z2} + (m_2 (l_{c2}^2 - l_{c1}^2))) \ddot{\psi} \\ & + m_2 (l_{c2} \cos(\theta) - l_{c1}) (\dot{v}_{\text{lat}} + (\dot{\psi} v_{\text{long}})) + m_2 l_{c2} \sin(\theta) (\dot{v}_{\text{long}} - (\dot{\psi} v_{\text{lat}})) \\ & + m_2 l_{c1} l_{c2} \sin(\theta) ((\dot{\psi} - \dot{\theta})^2 + \dot{\psi}^2) - l_{f1} \sin(\delta_f) F_{x,f} - l_{f1} \cos(\delta_f) F_{y,f} \\ & + l_{r1} F_{y,r1} - F_{x,r2} l_{c1} \sin(\theta) + F_{y,r2} l_{c1} \cos(\theta) - (l_{c2} + l_{r2}) F_{y,r2} \end{aligned} \quad (3.15)$$

$$0 = -\dot{\delta}_f + \omega_f \quad (3.16)$$

$$\alpha_{r2} = \arctan\left(\frac{-v_{\text{lat, trailer}} + l_{r2}(\dot{\psi} - \dot{\theta})}{v_{\text{long, trailer}}}\right) \quad (3.21)$$

Due to the singularity issue with longitudinal velocities near zero in the above formula, the tire sideslip angle is assumed to be negligible at low velocities. For simplicity, and by assuming linear tire behavior, the lateral tire forces are calculated using the following equations.

$$F_{y,f} = C_f \alpha_f \quad (3.22)$$

$$F_{y,r1} = C_{r1} \alpha_{r1} \quad (3.23)$$

$$F_{y,r2} = C_{r2} \alpha_{r2} \quad (3.24)$$

While C_f , C_{r1} , and C_{r2} represent the total cornering stiffness of the front axle tires of the tractor section, the total cornering stiffness of the rear axle tires of the tractor section, and the total cornering stiffness of the rear axle tires of the semi-trailer section, respectively. The longitudinal forces $F_{x,f}$, $F_{x,r1}$, and $F_{x,r2}$ are defined by:

$$F_{x,f} = -F_{r,f} \quad (3.25)$$

$$F_{x,r1} = -F_d - F_{r,r1} \quad (3.26)$$

$$F_{x,r2} = -F_{r,r2} \quad (3.27)$$

where $F_d = (m_1 + m_2)a$ is the driving force at the rear axle of the tractor section, and $F_{r,f}$, $F_{r,r1}$, and $F_{r,r2}$ are the rolling resistance forces, calculated as follows:

$$F_{r,f} = f_{\text{tractor}} F_{z,f} \quad (3.28)$$

$$F_{r,r1} = f_{\text{tractor}} F_{z,r1} \quad (3.29)$$

$$F_{r,r2} = f_{\text{trailer}} F_{z,r2} \quad (3.30)$$

Here, f_{tractor} and f_{trailer} represent the rolling resistance coefficients for the tractor and trailer, respectively, while $F_{z,f}$, $F_{z,r1}$, and $F_{z,r2}$ are the vertical forces acting on the respective axles, as follows [12]:

$$F_{z,r2} = \frac{m_2 l_{c2} g}{l_{c2} + l_{r2}} \quad (3.31)$$

$$F_{\text{prime}} = \frac{m_2 l_{r2} g}{l_{c2} + l_{r2}} \quad (3.32)$$

$$F_{z,f} = \frac{(m_1 g + F_{\text{prime}}(1 - (\frac{l_{c1}}{l_{r1}})))l_{r1}}{l_{c2} + l_{r2}} \quad (3.33)$$

$$F_{z,r1} = F_{z,f}(\frac{l_{r1}}{l_{r1}}) + F_{\text{prime}}(\frac{l_{c1}}{l_{r1}}) \quad (3.34)$$

$$f_{\text{tractor}} = f_{\text{trailer}} = \frac{55}{g \times 1000} \quad (3.35)$$

All base parameters are listed in Table 3.2 [12, 21]. However, most model parameters are influenced by factors such as weather, temperature, terrain, or vehicle loading, meaning the identified values are only valid under the specific conditions reported in [12, 21]. In this study, to develop an adaptive NMPC for a semi-trailer truck, we assume that only two parameters—the trailer's mass m_2 and its moment of inertia I_{z2} —are mutable. However, the proposed method can be extended to other parameters by following the same procedure. All such mutable parameters are grouped into a parameter vector, p , within the prediction model.

$$\mathbf{p} = [m_2 \quad I_{z2}]^T \quad (3.36)$$

To adapt the dynamics model for use in motion control, it is discretized using the Implicit Runge-Kutta 4th

Table 3.2: Vehicle and Tire parameters[12, 21]

Symbol	Unit	Nominal Value	Mutable
General Vehicle Parameters			
m_1	kg	8450	
m_2	kg	37255	✓
I_{z1}	kgm ²	20610	
I_{z2}	kgm ²	700502	✓
l_{f1}	m	1.385	
l_{r1}	m	4.25	
l_{c1}	m	4.25	
l_{r2}	m	4.72	
l_{c2}	m	5.5	
ρ	kg/m ³	1.225	
A	m ²	10.0	
c_d	-	0.8	
General Tire Parameters			
f_{tractor}	-	0.0056	
f_{trailer}	-	0.0056	
Linear Tire Parameters			
C_f	N/rad	135010	
C_{r1}	N/rad	477620	
C_{r2}	N/rad	550360	

Order (IRK4) with a sampling time T_s . The resulting discretized, parameter-dependent one-step prediction

model will hereinafter be referred to as:

$$\tilde{x}_k(x_{k-1}, u_{k-1}, p_{k-1}) \quad \text{respectively} \quad \tilde{x}_k(x_{k-1}, u_{k-1}, p) \quad (3.37)$$

depending on whether the parameter set at the time $k - 1$ is known as p_{k-1} or used in the form of an unknown variable as p . Similarly, multi-step predictions, obtained by iteratively feeding one-step predictions back into the model, are referred to as:

$$\tilde{x}_k(x_{k-\gamma}, U_{k-\gamma:k-1}, P_{k-\gamma:k-1}) \quad \text{respectively} \quad \tilde{x}_k(x_{k-\gamma}, U_{k-\gamma:k-1}, P) \quad (3.38)$$

where $U_{k-\gamma:k-1}$ is defined as a series of control inputs starting at the discrete time $k - \gamma$ and including the discrete time $k - 1$, with γ determining the number of steps. Similarly, $P_{k-\gamma:k-1}$ refers to a series of parameters. However, to reduce the number of variables, it is also possible to utilize the same single parameter set, rather than a series of parameters, for every step of the multi-step prediction. In that case, the convention is to use the lowercase letter p instead of the capital P , which would indicate a series of values.

3.2 Model Predictive Control and Moving Horizon Estimation

3.2.1 Model Predictive Control

MPC is an advanced control strategy that optimizes the control inputs of a system by predicting its future behavior. Unlike traditional control methods, MPC relies on a dynamic model of the system and incorporates constraints on states and inputs. At each time step, MPC solves an optimization problem over a finite prediction horizon, selecting the optimal sequence of control actions. Only the first control action is implemented, and the process is repeated in a receding-horizon fashion, updating predictions with new measurements [46, 47].

General formulation of MPC

The general formulation of MPC involves the following components:

1. **System Model:** A dynamic model of the system, typically in discrete-time form:

$$x_{k+1} = f(x_k, u_k) \quad (3.39)$$

where $x_k \in \mathbb{R}^n$ represents the state vector, $u_k \in \mathbb{R}^m$ represents the control inputs, and $f(\cdot)$ defines the system dynamics.

2. **Cost Function:** A quadratic cost function to minimize deviations from desired states and control efforts (if any):

$$J = \sum_{i=0}^{N-1} \left(\|x_{k+i} - x_{\text{ref},k+i}\|_Q^2 + \|u_{k+i} - u_{\text{ref},k+i}\|_R^2 \right) \quad (3.40)$$

where Q and R are weighting matrices, and N is the prediction horizon steps.

3. **Constraints:** Physical or operational constraints on states and inputs:

$$x_{\min} \leq x_k \leq x_{\max}, \quad u_{\min} \leq u_k \leq u_{\max} \quad (3.41)$$

4. **Optimization Problem:** At each time step k , MPC solves:

$$\min_{u_{k:k+N-1}} J \quad \text{subject to } x_{k+1} = f(x_k, u_k), x_k, u_k \in \mathcal{C}. \quad (3.42)$$

3.2.2 Moving Horizon Estimation

MHE is an optimization-based state estimation method that provides optimal estimates by solving an optimization problem over a finite estimation horizon. It infers the current state of the system based on a finite sequence of past measurements, using a dynamic model of the system, measurements, and noise characteristics. MHE minimizes a cost function that accounts for estimation errors and the influence of process and measurement noise. Unlike classical estimators like the KF, MHE explicitly handles constraints on states and noise, making it particularly suitable for systems with nonlinearity or operational constraints. In many ways, MHE serves as the counterpart to MPC [40, 48]. Figure 3.2 provides a simplified representation of the MPC and MHE frameworks.

General formulation of MHE

The general formulation of MHE involves:

1. **System Model:** The dynamic model, often expressed in discrete-time form:

$$x_{k+1} = f(x_k, u_k) + w_k, \quad y_k = h(x_k) + v_k \quad (3.43)$$

where w_k and v_k represent process and measurement noise, respectively.

2. **Cost Function:** MHE minimizes a cost function over the estimation horizon steps N_e :

$$J = \sum_{i=k-N_e}^{k-1} \|x_{k-i} - \hat{x}_{k-i}\|_Q^2 + \sum_{i=k-N_e}^{k-1} \|w_i\|_{Q_w}^2 + \sum_{i=k-N_e}^k \|v_i\|_{Q_v}^2 \quad (3.44)$$

where \hat{x}_{k-i} is the estimated state at $k-i$, and Q , Q_w , and Q_v are weighting matrices.

3. **Constraints:** Incorporates constraints on states and noise:

$$x_{\min} \leq x_k \leq x_{\max}, \quad w_k \in \mathcal{W}, \quad v_k \in \mathcal{V}. \quad (3.45)$$

4. **Optimization Problem:** At time k , MHE solves:

$$\min_{x_{k-N:k-1}, w_{k-N:k-1}, v_{k-N:k}} J \quad \text{subject to system dynamics and constraints.} \quad (3.46)$$

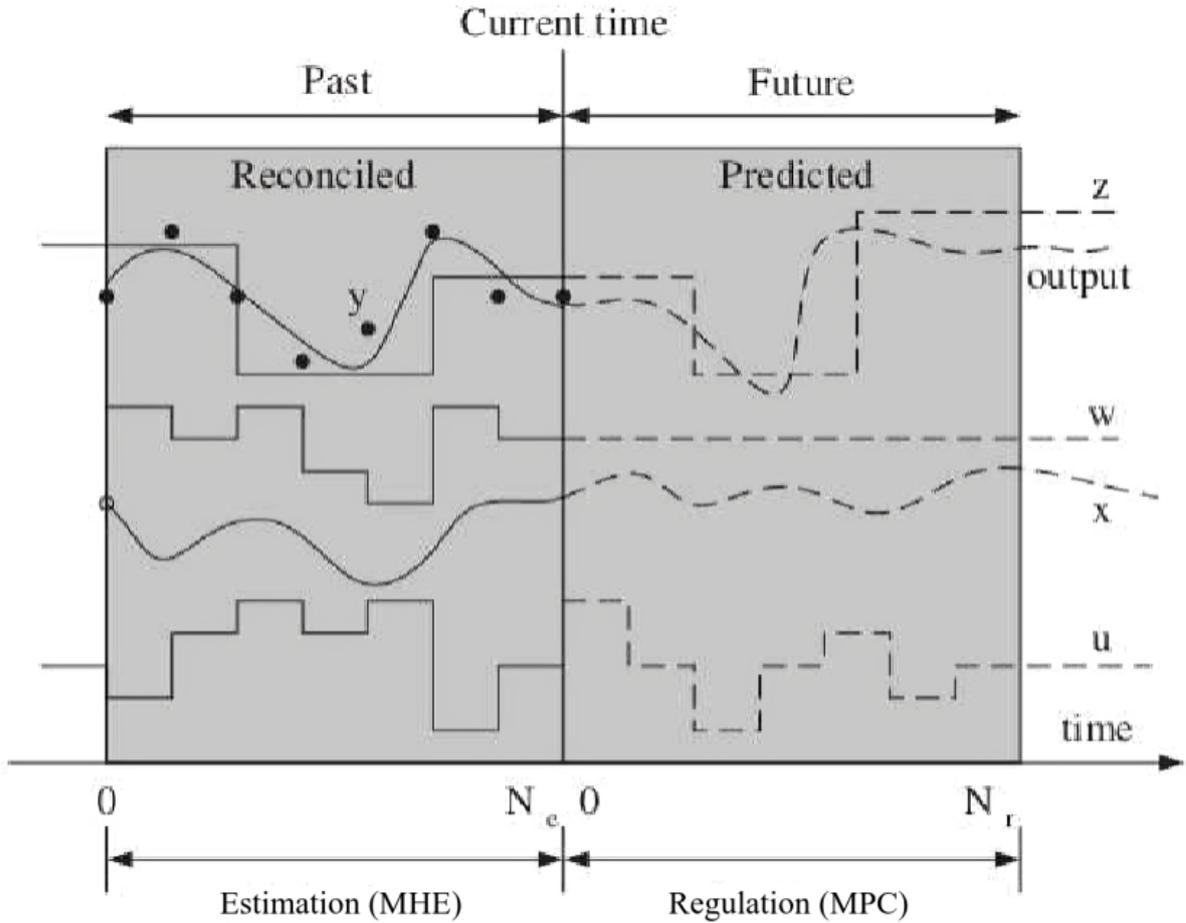


Figure 3.2: MHE and MPC frameworks

3.2.3 Application of MHE in Parameter Estimation

In addition to state estimation, MHE can be extended to estimate unknown parameters of a system by augmenting the state vector [40, 49]. For instance, consider a system with parameters $p \in \mathbb{R}^p$ to be estimated:

$$x_{k+1} = f(x_k, u_k, p) + w_k, \quad y_k = h(x_k, p) + v_k. \quad (3.47)$$

1. **Augmented State Vector:** In this study, the state vector and parameter vector are combined to form an augmented vector that captures both the system's dynamic states and the mutable parameters:

$$\bar{x}_k = \begin{bmatrix} x_k \\ p \end{bmatrix} \quad (3.48)$$

2. **Cost Function:** Minimize estimation errors over the augmented vector:

$$J = \sum_{i=k-N_e}^{k-1} \|\bar{x}_{k-i} - \hat{\bar{x}}_{k-i}\|_Q^2 + \sum_{i=k-N_e}^{k-1} \|w_i\|_{Q_w}^2 + \sum_{i=k-N_e}^k \|v_i\|_{Q_v}^2. \quad (3.49)$$

3. **Optimization:** The same optimization framework as state estimation is used, but now includes parameter dynamics (if any).

4 Methodology

Reflecting on the problem definition in Section 1.2, the development of an adaptive NMPC for any dynamic system requires a model capable of adequately capturing highly dynamic behavior, as introduced in Section 3.1. To address the challenge of parameter mismatches between the controller strategy and the real vehicle, various methods described in Chapter 2 have been evaluated. Since this study specifically focuses on two vehicle parameters—the mass and inertia of the trailer section—and the derived semi-truck dynamic model is sufficiently complex, MHE, as briefly outlined in Subsection 3.2.3, has been chosen to enable NMPC to adapt.

However, as noted in Subsection 2.2.2, implementing MHE involves several challenges, including the selection of hyper-parameters, managing arrival costs, addressing feasibility issues, and mitigating high computational demands. This chapter is dedicated to presenting the methodological procedure undertaken in this study. Figure 4.1 illustrates the three critical steps undertaken to develop the complete architecture required for adapting the motion control of a semi-truck. These steps are as follows:

1. **Prediction Model Development:** Creating a model capable of accurately representing the semi-truck's dynamic behavior.
2. **Designing Nonlinear Model Predictive Control (NMPC):** Developing a control strategy tailored to the semi-truck's motion control requirements.
3. **Implementation of Moving Horizon Estimation (MHE):** Incorporating an adaptive mechanism to handle parameter mismatches effectively.

Each of these steps is described in detail in the following sections.

4.1 Semi-Truck Prediction Model Development

As introduced in Section 3.1, a single-track dynamic model for a semi-truck, presented in [29], with minor modifications to the input variables, is employed as the prediction model for this study. Before incorporating this prediction model into the NMPC and MHE frameworks, it must be validated. To achieve this, we utilized the scenario described in [29], with the following system inputs: torque applied to the front wheels of the tractor section, in $[N.m]$, and the front-wheel steering angle, in $[\circ]$, as shown in Figure 4.2.

Since the input variables of our prediction model, based on Equation 3.7, are the longitudinal jerk, in $[\frac{m^3}{s^3}]$, and the front-wheel steering rate, in $[\frac{rad}{s}]$, the inputs from [29] were transformed accordingly. Specifically, the front-wheel steering angle was first converted to radians and then differentiated to obtain the front-wheel steering rate. Similarly, using the effective tire radius, the combined mass of the tractor and trailer, and the following equation:

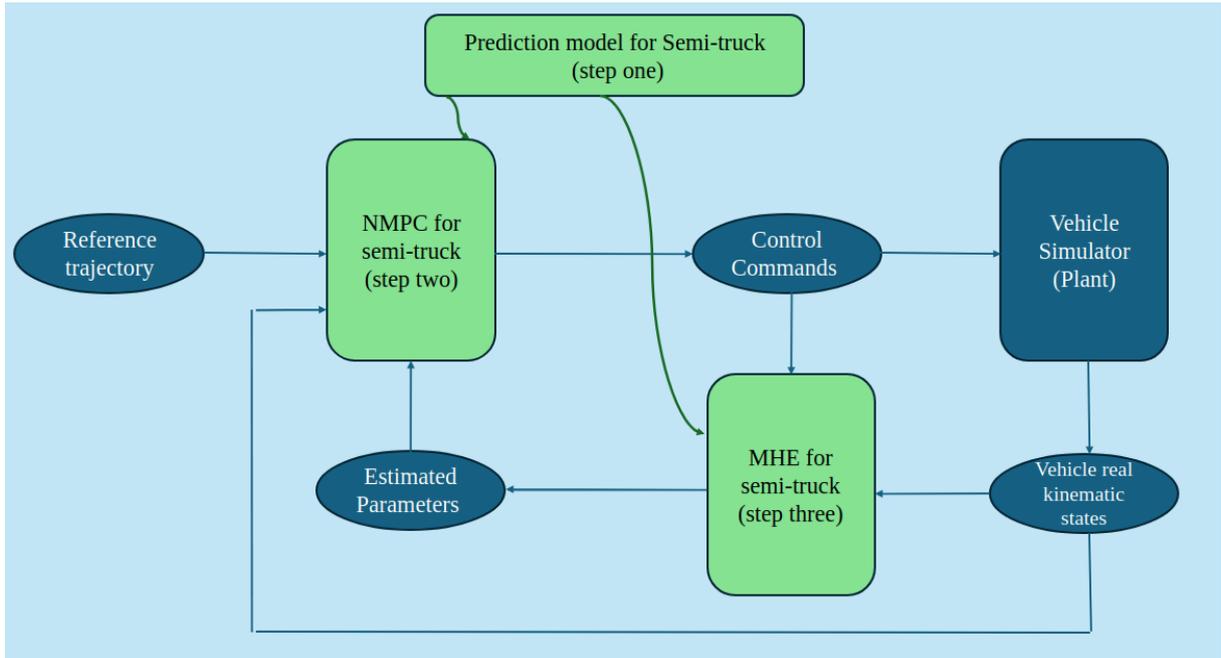


Figure 4.1: Entire motion control architecture and three undertaken steps

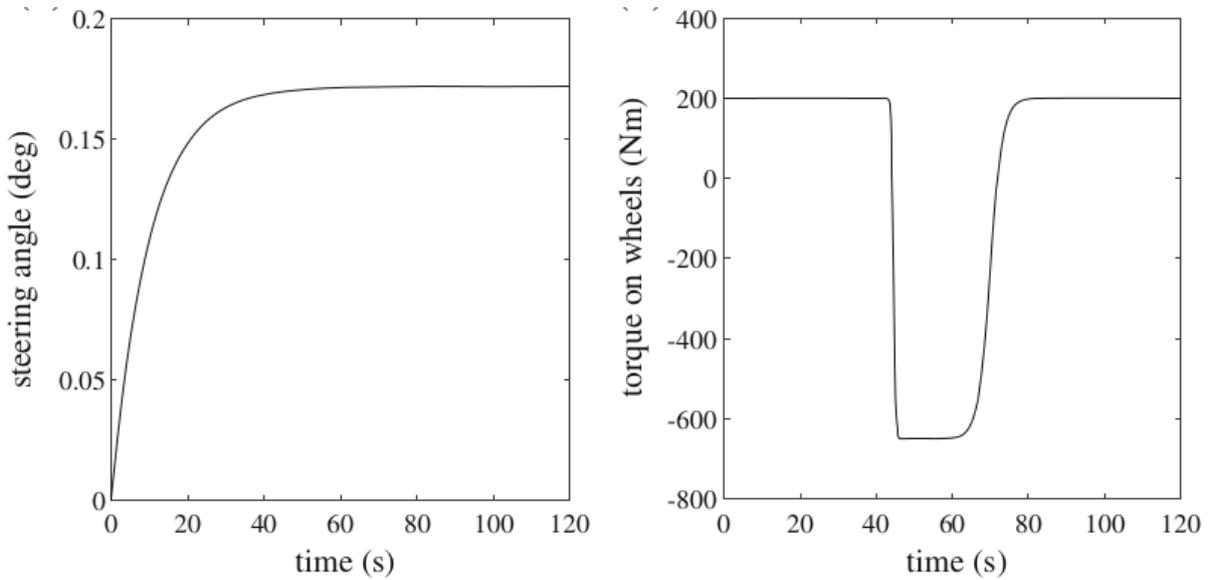


Figure 4.2: Model inputs reported in [29] for validation process

$$a = \frac{torque}{(m_{tractor} + m_{trailer})R_{tire}} \quad (4.1)$$

the longitudinal acceleration a_a was calculated. By differentiating a_a , the longitudinal jerk was derived.

The simulation was then executed using the scenario and vehicle parameters outlined in [29]. The results were compared against the reported outcomes in the reference to ensure the model's correctness. Further details of this validation process are presented in Section 5.1, Prediction Model Validation, of this study.

4.2 Nonlinear Model Predictive Control for Semi-Trucks

The general formulation and key components of an MPC were introduced in Subsection 3.2.1. In this section, we expand upon this framework to develop an NMPC tailored to the trajectory tracking requirements of semi-trucks. This process, illustrated in Figure 4.3, operates at a rate of 50 Hz, corresponding to the update frequency of the vehicle interface. Consequently, the discretization time T_d of the semi-truck dynamics model is set to $T_d = 20$ ms to ensure synchronization with the simulation loop. Each iteration of setting control commands by the NMPC consists of four main steps:

1. **Initialization:** All necessary input data, including the reference trajectory, current states, and estimated parameters (if a parameter adaptation technique is employed), are verified and provided to the controller.
2. **Prediction:** The nonlinear model is used to predict the system's future states over the prediction horizon.
3. **Optimization Problem Solving:** A cost function is formulated, and a constrained optimization problem is solved to determine the optimal control sequence.
4. **Setting Control Command:** The optimal control sequence is retrieved, and only the first control input is applied to the system, as well as sent to the estimator developed based on the MHE method.

In the subsequent sections, each individual part of this four-step process is described in detail.

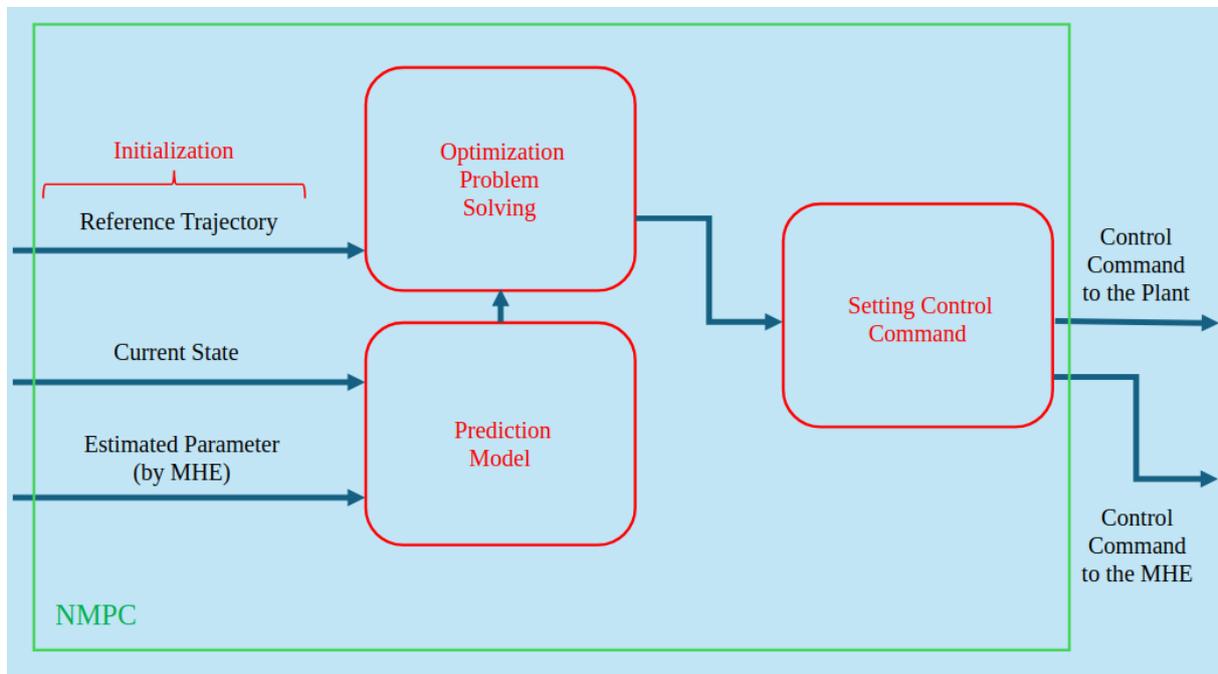


Figure 4.3: NMPC procedure

4.2.1 Initialization

To make predictions and solve the optimization problem, the NMPC requires the following initial data:

Reference Trajectory:

These data include not only the desired longitudinal and lateral positions of the semi-truck that are supposed to be followed, but also all other desired state values considered in the NMPC cost function. In this study, each time step of the reference (desired) trajectory is represented as:

$$x_{\text{ref},k} = [\text{ref}_x, \text{ref}_y, \text{ref}_\psi, \text{ref}_\theta, \text{ref}_{v_{\text{long}}}] \quad (4.2)$$

Here, ref_x , ref_y , ref_ψ , and $\text{ref}_{v_{\text{long}}}$ represent the reference values for the longitudinal position, lateral position, yaw angle, and longitudinal velocity, respectively, all corresponding to the center of mass of the tractor. Additionally, ref_θ is the reference hitch angle. However, in this study, since a reliable reference value for the hitch angle is unavailable, it is included in the reference trajectory vector but excluded from influencing the cost function. Specifically, the weighting coefficient of this parameter is set to zero when forming the cost function (described in subsequent sections), ensuring it does not affect the optimization problem's solution. Nevertheless, the maximum and minimum constraints of the hitch angle are still enforced within the optimization problem. Furthermore, $x_{\text{ref},k}$ represents the reference trajectory vector at time step k . If the prediction horizon consists of N steps, the NMPC requires all reference trajectory vectors from k to $k + N$.

Current State:

The current state vector is utilized in the prediction model to estimate future states over the prediction horizon. The current state can be directly measured, or in cases where certain states are not readily available, they must be estimated using methods such as Moving Horizon Estimation (MHE). Alternatively, previously predicted values can be employed to compensate for the lack of real-time data. In the main part of this study, it is assumed that all states are measurable, as described in Section 3.1.

When dealing with noisy data, it is recommended to use methods such as a moving average filter, which can help reduce the impact of noise. Additionally, this filter can also serve as a state estimator in cases where state data is temporarily unavailable or lost during short time intervals [50].

Estimated Parameter:

Estimated parameters are also crucial for the prediction model to forecast future states over the prediction horizon. These parameters are assumed to remain constant throughout the prediction horizon. As mentioned previously, the parameters to be estimated in this study include the mass and inertia of the trailer section. For estimating these parameters, the MHE method is utilized, which will be discussed in detail in the following sections.

4.2.2 Prediction

The nonlinear model defined in Section 3.1 is employed to forecast the system's future states. Considering the mutability of vehicle parameters, this model can be represented as follows:

$$0 = f_{\text{cont}}(\dot{x}, x, u, p) \quad (4.3)$$

Since the equations are presented in an implicit form, transforming them into the explicit form $\dot{x} = g(x, u, p)$ is computationally intensive and impractical. To obtain the next state, x_{k+1} , at the discretized time step $k + 1$, the Implicit Runge-Kutta 4th-order method (IRK4) is employed to compute the model's evolution. The IRK4

method is described in detail in [51], and further details are omitted here as they are beyond the scope of this study.

4.2.3 Optimization Problem Solving

The optimization problem is the core of NMPC and involves determining the control inputs that optimize system performance while respecting constraints. This step integrates system dynamics, the cost function, and constraints into a solvable mathematical framework. To systematically address this, the process can be broken down into the following key steps:

1. **Cost Function Formulation:** Defining the performance index to be minimized, which typically includes terms related to tracking error, control effort, and other performance objectives.
2. **Constraints:** Incorporating physical and operational constraints, such as state limits, control input bounds, and system safety requirements.
3. **Solving the Optimization Problem:** Employing numerical solvers to compute the optimal control sequence over the prediction horizon.

Cost Function Formulation

The cost function J_{nmPC} (objective function) quantifies the control objectives and provides a measure to minimize. The cost function utilized in the developed NMPC of this study is:

$$J_{\text{nmPC}} = \sum_{k=0}^{N-1} \left(\|\tilde{x}_k - x_{\text{ref},k}\|_Q^2 + \|u_k\|_R^2 \right) + \|\tilde{x}_N - x_{\text{ref},N}\|_{Q_e}^2 \quad (4.4)$$

Terms in detail:

1. State Tracking Term:

$$\sum_{k=0}^{N-1} \|\tilde{x}_k - x_{\text{ref},k}\|_Q^2$$

- Measures the squared error between the predicted state \tilde{x}_k and the reference state $x_{\text{ref},k}$ over the prediction horizon.
- Weighted by the state weighting matrix Q , which assigns importance to individual state variables.

2. Control Effort Term:

$$\sum_{k=0}^{N-1} \|u_k\|_R^2$$

- Penalizes the magnitude of control inputs u_k , encouraging smooth and efficient control.
- Weighted by the control weighting matrix R , which regulates the emphasis on control input minimization.

3. Terminal Cost:

$$\|\tilde{x}_N - x_{\text{ref},N}\|_{Q_e}^2$$

- Encourages the final predicted state \tilde{x}_N to be close to the terminal reference state $x_{\text{ref},N}$.

- Weighted by the terminal cost weighting matrix Q_e , reflecting the importance of achieving the desired terminal state.

The norms in the cost function are weighted Euclidean norms defined as:

$$\|\tilde{x} - x_{\text{ref}}\|_Q^2 = (\tilde{x} - x_{\text{ref}})^T Q (\tilde{x} - x_{\text{ref}}) \quad (4.5)$$

and similarly for control effort and terminal cost, using R and Q_e as weighting matrices, respectively.

Constraints

following constraints are considered in defining optimization control problem of a semi-truck:

1. Nonlinear Inequality Constraints:

These constraints regulate the combined lateral acceleration (a_{lat}) and longitudinal acceleration (a_{lon}) based on the vehicle's current velocity, derived from the GGV diagram. The GGV diagram is a graphical representation or data structure used in vehicle dynamics to describe the achievable combinations of longitudinal (G_x) and lateral (G_y) accelerations at different vehicle velocities.

In this study, due to the lack of a reliable GGV diagram for semi-trucks, the GGV diagram of EDGAR is used. Additionally, it is important to note that lateral acceleration values are computed as follows:

$$a_{\text{lat}} = v_{\text{long}} \dot{\psi} \quad (4.6)$$

The inequality constraints are implemented based on a configuration parameter named `combined_acc_limits`, with the available options depicted in Figure 4.4.

- Separated Limits (`combined_acc_limits: 0`)
- Diamond-Shaped Limits (`combined_acc_limits: 1`)
- Circle-Shaped Limits (`combined_acc_limits: 2`)

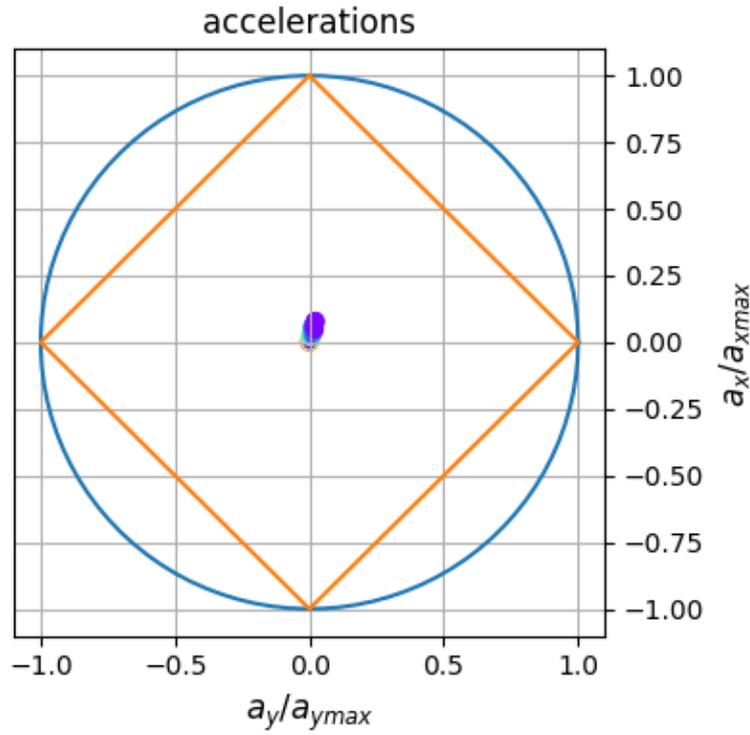


Figure 4.4: Visual representative of nonlinear inequality constraints

2. State Constraints:

Two states, the angle of the hitch point (θ) and the tire steering angle (δ_f), are bounded by the following ranges:

$$-0.785 \leq \theta \leq 0.785 \quad [\text{rad}] \quad (4.7)$$

$$-0.610865 \leq \delta_f \leq 0.610865 \quad [\text{rad}] \quad (4.8)$$

3. Input Constraints:

One input, the tire steering rate (ω_f), is bounded by the following range:

$$-0.322 \leq \omega_f \leq 0.322 \quad [\text{rad/s}] \quad (4.9)$$

In this study, slack variables are also utilized to soften the constraints. These variables are introduced to relax constraints and ensure solver feasibility in cases of slight constraint violations. Slack variables are applied to all types of constraints in the present study. This is achieved by adding a cost term to penalize constraint violations in both the stage and terminal cost functions. For example, if we consider a state constraint as:

$$x_{\min} \leq x_k \leq x_{\max} \quad (4.10)$$

To allow for slight violations of the constraints, slack variables ξ^+ and ξ^- are introduced. The modified constraint is:

$$x_{\min} - \xi^- \leq x \leq x_{\max} + \xi^+ \quad (4.11)$$

where:

- $\xi^- \geq 0$: Represents the violation of the lower bound.
- $\xi^+ \geq 0$: Represents the violation of the upper bound.

4. Considering Model Dynamics:

The system dynamics define the evolution of states under given inputs. For NMPC, these dynamics are embedded as constraints, and, as described before, the continuous-time dynamics are discretized using the IRK4 method to ensure numerical accuracy and stability in the NMPC framework.

Solving the Optimization Problem

The Non Linear Programming (NLP) optimization problem in NMPC is addressed using methods such as Sequential Quadratic Programming (SQP), Interior-Point Methods (IPMs), and Direct Collocation (DC), which effectively handle system dynamics and constraints. Providing a good initial guess, such as the solution from the previous step, enhances convergence speed. Real-time feasibility is ensured through parallel computing, warm-start techniques, and potential simplifications like reducing the prediction horizon.

In this study, the powerful open-source framework Automatic Control And Dynamic Optimization Software (ACADOS) is utilized for efficient optimal control problem-solving. ACADOS offers advanced solvers, including SQP, SQP Real Time Iteration (SQP-RTI), and IPMs. The SQP-RTI solver is specifically selected for its computational efficiency and suitability for real-time applications, making it ideal for fast decision-making in control tasks.

4.2.4 Setting Control Command

At each iteration, solving the optimization problem yields the following two results:

1. **Sequence of Optimal Control Commands:** The sequence of optimal control inputs, $U_{\text{opt},k}$, at the current time step k with a prediction horizon of N steps is given by:

$$U_{\text{opt},k} = [u_{\text{opt},k} \quad u_{\text{opt},k+1} \quad u_{\text{opt},k+2} \quad \dots \quad u_{\text{opt},k+N-1}] \quad (4.12)$$

2. **Sequence of Predicted States:** The sequence of predicted states, \tilde{X}_k , over the prediction horizon is:

$$\tilde{X}_k = [\tilde{x}_{k+1} \quad \tilde{x}_{k+2} \quad \tilde{x}_{k+3} \quad \dots \quad \tilde{x}_{k+N}] \quad (4.13)$$

After computing these results, only the first optimal control input, $u_{\text{opt},k}$, is applied to the system (plant) and sent to the MHE estimator. The predicted states, \tilde{X}_k , can also be utilized to estimate state values when direct measurements of certain states are unavailable or incomplete.

4.3 Moving Horizon Estimation for Semi-Trucks

In subsection 3.2.3, the basics and fundamental formulation of Moving Horizon Estimation (MHE) were introduced. In this section, this method will be expanded in detail to develop a parameter estimation strategy that enables the NMPC to adapt and improve its performance in the presence of parameter mismatch between the controller and the system. This process is illustrated in Figure 4.5 and, similar to NMPC,

operates at a frequency of 50 Hz. Consequently, the estimation discretization time, T_e , of the semi-truck estimation model is set to $T_e = T_d = 20$ ms to ensure synchronization with the entire loop.

It is crucial to emphasize that the primary objective of using MHE or any other parameter estimation method is to enhance the performance of the controller, in this case the NMPC, by improving prediction accuracy. This does not imply that the estimated parameters must match the exact physical values at every iteration. Slight deviations between estimated and actual values are acceptable, as long as they contribute to better prediction and control performance. However, significant discrepancies between the estimated parameters and their true values can adversely affect the controller's overall performance.

Each iteration of parameter estimation using MHE involves five main steps:

1. **Circular Data Buffer:** The estimator receives measurements of the current state, the control input applied to reach this state, and the previously estimated parameters. The system's current states and the previously estimated parameters are combined into a new state vector x_{buff} , which is stored in a circular data buffer along with the corresponding control inputs. The buffer retains the most recent N_e entries, where N_e represents the number of estimation horizon steps.
2. **Data Preparation:** The stored data in the circular buffer is pre-processed to enhance computational efficiency and mitigate the impact of noise, ensuring robust parameter estimation.
3. **Recall Estimation Model:** The estimation model, constructed using the augmented state vector, is retrieved and integrated into the optimization framework of the MHE.
4. **Optimization Problem Solving:** A cost function is formulated, and a constrained optimization problem is solved to estimate the states or parameters exhibiting mismatches.
5. **Estimated Parameters Application:** From the series of estimated states and parameters, the most recent parameter values are extracted. These values are sent to the NMPC for use in subsequent control iterations and stored back in the data buffer for the new iteration.

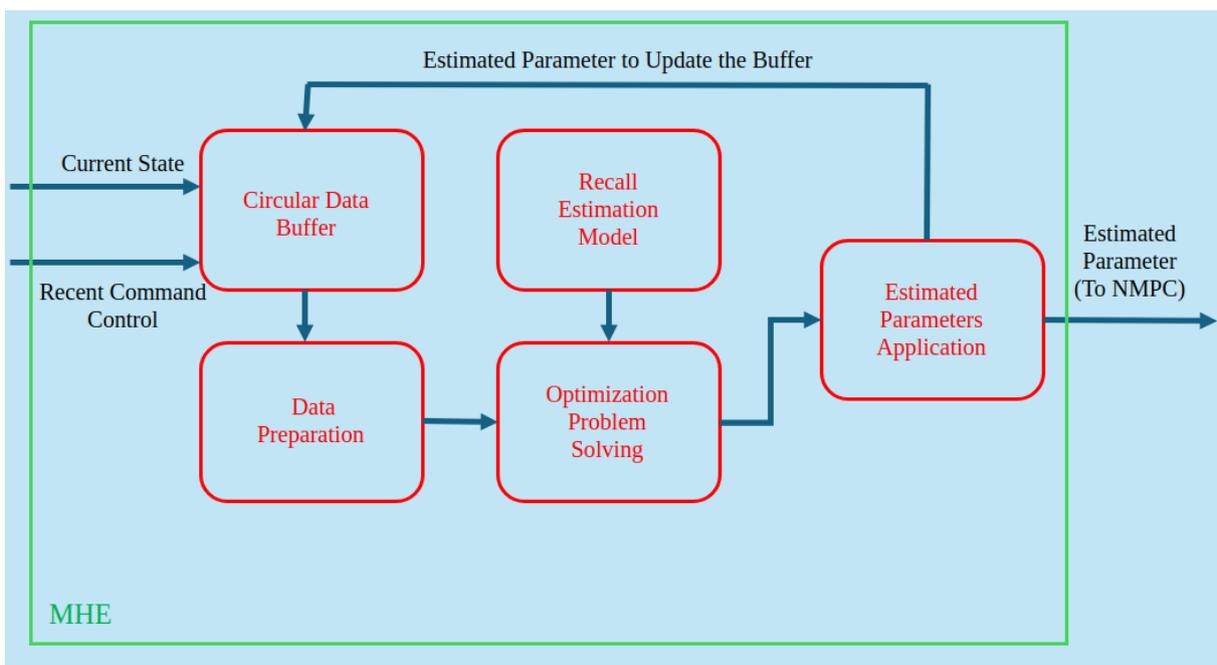


Figure 4.5: MHE procedure

The estimation process begins only after the data buffer has been fully populated, ensuring that sufficient data is available to avoid inaccuracies or miscalculations. In the following sections, each component of the estimation process using MHE is explained in detail. Here and in the following, the current discrete time is considered k .

4.3.1 Circular Data Buffer

The estimator receives measurements of the current state x_k , the control input applied to achieve this state u_{k-1} , and the previously estimated parameters p_{k-1} . At this point, it is assumed that all states are measurable. However, in cases where some states are unavailable, recent predicted values of the missing states or previously estimated states can be used as replacements. It is important to note that even partial replacement of state measurements can significantly degrade the estimation process, as it reduces the available information and may act as a self-fulfilling prophecy, producing data for future estimations using current estimated parameters.

Based on the explanation in Subsection 4.3.3, the state vector used in the developed estimation model, x_{est} , takes the form:

$$x_{\text{est}} = \begin{bmatrix} x \\ p \end{bmatrix} \quad (4.14)$$

Here, x represents the semi-truck dynamic model states introduced in Equation 3.1, and p denotes the mutable parameter vector defined in Equation 3.36.

To ensure consistency with x_{est} , the data buffer is structured as follows: first, the system's current state x_k and the previously estimated parameters p_{k-1} are concatenated into a single vector. This new vector, referred to as $x_{\text{buff},k}$, is then stored alongside the control input u_{k-1} .

To store the data, two circular buffers with a length of N_e are used. Here, N_e represents the number of previous data points that need to be stored, or equivalently, the estimation horizon step. The value of N_e is determined by the requirements formulated in Subsection 4.3.4.

Consequently, the series $X_{\text{buff}, -N_e+1+k:k}$, consisting of N_e consecutive $x_{\text{est},(\cdot)}$, and the series $U_{\text{buff}, -N_e+k:k-1}$, consisting of N_e consecutive $u_{(\cdot)}$, are stored. Simultaneously, the oldest state $x_{\text{buff}, -N_e+k}$ and the oldest control input u_{-1-N_e+k} are removed from the buffer to maintain the buffer size.

The full contents of the data buffer are then passed to the data preparation step.

4.3.2 Data Preparation

In order to estimate both states and parameters using MHE in our estimation model (described in Subsection 4.3.3), all mutable parameters need to be included as part of the system's state. Additionally, the stored control inputs in the circular data buffer serve as constant knowledge for the MHE optimization problem and are not optimization targets, as they are in NMPC.

Based on the nonlinear optimization problem configuration in the ACADOS tool, the stored control inputs $U_{\text{buff}, -N_e+k:k-1}$ must be used in the parameter field. Therefore, during data preparation, the stored control input series is relabeled as $P_{\text{buff}, -N_e+1+k:k}$ for inclusion in the parameter field of the MHE optimization problem.

Another data preparation step utilized is averaging the buffer data. This method mitigates the effects of measurement disturbances by smoothing and filtering the observed data. The adjacent stored states and

control inputs are smoothed based on the following relations:

$$x_{\text{buff},i+k} = \frac{x_{\text{buff},i+k} + x_{\text{buff},i+k+1} + x_{\text{buff},i+k+2}}{3}, \quad \text{with } i \in \{-N_e, \dots, -2\}, \quad (4.15)$$

$$u_{i+k} = \frac{u_{i+k} + u_{i+k+1} + u_{i+k+2}}{3}, \quad \text{with } i \in \{-N_e, \dots, -2\}. \quad (4.16)$$

However, it is essential to note that averaging introduces a certain loss of data fidelity. Therefore, the effects of this method should always be carefully assessed beforehand to ensure it does not compromise the estimation accuracy.

4.3.3 Recall Estimation Model

In the MHE framework, the estimation model plays a critical role in optimization problems by ensuring accurate state estimation despite measurement noise and system uncertainties. The integration of an accurate estimation model within MHE significantly enhances the performance of dynamic optimization in control systems by improving prediction fidelity and operational reliability [52].

The foundation of the estimation model is the dynamic model of the semi-truck, introduced in Section 3.1 and utilized as the prediction model in the NMPC. However, since the primary goal of MHE in this study is to estimate mutable parameters, the method defined in Subsection 3.2.3 has been chosen for implementation. Consequently, the dynamics model requires certain modifications to adapt it for use as the estimation model.

Following the method in [40], the augmented state vector is defined as shown in Equation 4.14. Therefore, the full state vector of the estimation model is represented as:

$$x_{\text{est}} = [x_{\text{pos}}, y_{\text{pos}}, \psi, \theta, v_{\text{long}}, v_{\text{lat}}, \dot{\psi}, \dot{\theta}, \delta_f, a, m_2, I_{z2}]^T \quad (4.17)$$

Additionally, since new states have been introduced, new dynamic equations must be defined to describe the evolution of these states. As described in [40], no evolution is considered for the parameters added as states, which are referred to as zero dynamics. Therefore, the complete set of differential equations governing the system dynamics in the estimation model consists of all equations from Equation 3.8 to Equation 3.17, along with the following two additional equations:

$$0 = \dot{m}_2 \quad (4.18)$$

$$0 = \dot{I}_{z2} \quad (4.19)$$

As described in subsection 4.3.2, the control inputs issued by the NMPC serve as pre-knowledge and are not the target of the MHE optimization problem. Consequently, in the estimation model, they are treated as parameters, as shown below:

$$p = [j \quad \omega_f]^T \quad (4.20)$$

It is important to note that, in this estimation model, there is no explicit input field. Therefore, the input vector is represented as:

$$u = [] \quad (4.21)$$

Table 4.1 summarizes the essential fields of the estimation model based on the ACADOS tool settings. This

Table 4.1: Estimation model fields, defined to be compliant with the ACADOS tool settings

Field Name	Assigned vector
x	$[x_{\text{pos}}, y_{\text{pos}}, \psi, \theta, v_{\text{long}}, v_{\text{lat}}, \dot{\psi}, \dot{\theta}, \delta_f, a, m_2, I_{z2}]^T$
\dot{x}	$[\dot{x}_{\text{pos}}, \dot{y}_{\text{pos}}, \dot{\psi}, \dot{\theta}, \dot{v}_{\text{long}}, \dot{v}_{\text{lat}}, \ddot{\psi}, \ddot{\theta}, \dot{\delta}_f, \dot{a}, \dot{m}_2, \dot{I}_{z2}]^T$
p	$[j \quad \omega_f]^T$
u	$[\]$

model is retrieved and integrated into the optimization framework of the MHE.

4.3.4 Optimization Problem Solving

The Optimization Problem for MHE involves constructing a mathematical framework that incorporates the cost function and system constraints. This approach ensures that the estimated states and parameters are consistent with measurement data, system dynamics, and prior knowledge. A summary of the key steps is provided below and will be explained in detail in the following pages.

1. **Previous Estimations Recall:** The previously stored estimated augmented state sequence is recalled to serve two purposes: As initial guesses and incorporated into the cost function formation.
2. **Cost Function Formulation:** The objective of the MHE strategy is to minimize the difference between the estimated states and the recorded states. Therefore, the cost function includes only tracking errors, with distinct weights and metrics assigned to the initial, intermediate, and final stages.
3. **Constraints:** In addition to the constraints already defined in the NMPC, parameter constraints are also incorporated to ensure physically meaningful estimations.
4. **Solving the Optimization Problem:** An appropriate numerical solver, provided by the ACADOS tool, is employed to compute the estimated augmented state sequence. The solver stores this sequence, and from it, the estimated parameters are subsequently extracted.

Previous Estimations Recall:

Based on the following equation:

$$\hat{X}_{\text{est},-\text{Ne}+k+1:k} = X_{\text{est},-\text{Ne}+k:k-1} = [x_{\text{est},-\text{Ne}+k}, x_{\text{est},-\text{Ne}+k+1}, x_{\text{est},-\text{Ne}+k+2}, \dots, x_{\text{est},k-1}] \quad (4.22)$$

$\hat{X}_{\text{est},-\text{Ne}+k+1:k}$ represents the estimated augmented state sequence computed in the previous iteration. This sequence is essential for two reasons:

1. Due to the iterative nature of solving the optimization problem, an initial guess is required. For the first estimation loop, where no previously estimated states are available, the initial values used in the NMPC combined with an initial guess for the parameters are utilized.
2. During the initial stage of cost function formulation, as shown in Equation 4.24, the first element of $\hat{X}_{\text{est},-\text{Ne}+k+1}, x_{\text{est},-\text{Ne}+k}$, is directly incorporated.

Cost Function Formulation:

Similar to NMPC, the cost function types in ACADOS are set as “NONLINEAR_LS”, which means the cost is computed as a nonlinear least squares problem. The utilized cost function of the MHE is :

$$J_{\text{mhe}} = J_0 + \sum_{i=1}^{N_e-2} J_i + J_e \quad (4.23)$$

Terms in detail:

1. Initial Stage J_0 :

$$J_0 = \|y_0 - y_{\text{ref}0}\|_{W_0}^2 \quad (4.24)$$

If we consider n_x as the number of states in x_{est} , then the target optimization vector at initial stage y_0 is defined as:

$$y_0 = [x_{\text{est},-N_e+1+k}[0 : n_x - 2], \quad x_{\text{est},-N_e+1+k}] \quad (4.25)$$

The reference vector $y_{\text{ref}0}$ is given by:

$$y_{\text{ref}0} = [X_{\text{buff},-N_e+1+k}[0 : n_x - 2], \quad x_{\text{est},-N_e+k}] \quad (4.26)$$

The first element of the previously stored augmented states, $\hat{X}_{\text{est},-N_e+k+1}$, plays a key role in smoothing out the parameter estimation process. Without this term, the estimated parameter values might fluctuate excessively, which could lead to divergence in the optimization process. Since the mutable parameters are part of the estimation model states with zero dynamics, as explained in Subsection 4.3.3, it is sufficient to regulate their value changes within just one stage of the cost function.

The weight matrix W_0 is defined as a block diagonal matrix composed of Q and $Q_{\text{mhe}0}$:

$$W_0 = \text{Block_diag}(Q, Q_{\text{mhe}0}) \quad (4.27)$$

Here:

- Q is a diagonal matrix of order $n_x - 2$, assigning importance to individual state variables.
- $Q_{\text{mhe}0}$ is a diagonal matrix of order n_x , acting as a penalty term to discourage the estimated states at the first stage from deviating significantly from their previously estimated values.

2. Intermediate Stages J_i :

$$J_i = \|y_i - y_{\text{ref}i}\|_W^2 \quad (4.28)$$

In the intermediate stages, the target optimization vector y_i is defined as:

$$y_i = [x_{\text{est},-N_e+1+i+k}[0 : n_x - 2]] \quad (4.29)$$

The reference vector $y_{\text{ref}i}$ is given by:

$$y_{\text{ref}i} = [X_{\text{buff},-N_e+1+i+k}[0 : n_x - 2]] \quad (4.30)$$

For the intermediate stages of the cost function, the reference values $y_{\text{ref},i}$ are set to the actual system states recorded in the circular data buffer. The optimization solver in the MHE estimator aims to minimize

the discrepancy between the estimated states and these real values by selecting an appropriate set of parameters.

The weight matrix W for the intermediate stages is defined as:

$$W = \text{Block_diag}(Q) \quad (4.31)$$

The weights Q and $Q_{\text{mhe}0}$ are crucial for tuning the estimator's performance.

3. Final Stage J_e :

$$J_e = \|y_e - y_{\text{ref}_e}\|_{W_e}^2 \quad (4.32)$$

In the final stage of the cost function, the target optimization vector y_e is defined as:

$$y_e = [x_{\text{est},k}[0 : nx - 2]] \quad (4.33)$$

The reference vector y_{ref_e} is given by:

$$y_{\text{ref}_e} = [X_{\text{buff},k}[0 : nx - 2]] \quad (4.34)$$

In this formulation, the reference vector y_{ref_e} is derived from the most recent recorded system state stored in the circular data buffer. Similar to the intermediate stages, the goal at the final stage is to minimize the deviation between the estimated and actual states. The weight matrix for the final stage, W_e , is selected to be identical to the weight matrix used in the intermediate stages, ensuring a consistent influence of state variables throughout the optimization process.

$$W_e = W = \text{Block_diag}(Q) \quad (4.35)$$

This choice simplifies the tuning process and maintains uniformity in the cost function's impact across all stages, except the first stage, of the estimation horizon.

Constraints:

The MHE optimization problem incorporates constraints to ensure that the estimated states and parameters remain consistent with physical and practical considerations. Many of the constraints applied in the NMPC optimization problem are carried over to the MHE, with some necessary adjustments detailed below.

1. **New State Constraints:** In the estimation model, mutable parameters such as the mass (m_2) and inertia (I_{z2}) of the trailer part were added as new states. To ensure physically meaningful estimations, constraints were defined for these parameters. These new state constraints are incorporated alongside the other state constraints:

$$3725 \leq m_2 \leq 372550 \quad [\text{kg}] \quad (4.36)$$

$$70050 \leq I_{z2} \leq 7005020 \quad [\text{kg.m}^2] \quad (4.37)$$

These limits were selected based on the values reported in Table 3.2. The lower limits are set to be ten times smaller, and the upper limits are set to be ten times larger than the reported values. It is important to note that these limits are not strict and are considered solely for the purposes of this study. The inclusion of these

constraints ensures that parameter estimations remain realistic while allowing for flexibility in the estimation process.

2. **Removing Input Constraints:** Since the control inputs issued by the NMPC are entered into the parameter field of the estimation model and remain fixed during the optimization process, there is no need for input constraints in the MHE formulation. Additionally, as per the design of the estimation model, there is no explicit input vector (u) to constrain.

Solving the Optimization Problem:

For solving the optimization problem in the MHE, similar to the approach used in NMPC, as explained in Subsection 4.2.3, the SQP-RTI solver, provided by the ACADOS tool, has been selected. This solver is chosen due to its ability to efficiently handle the nonlinear optimization problems inherent in the estimation process. The SQP-RTI solver computes the sequence of the estimated augmented states at each iteration $X_{\text{est},-N_{e+k+1}:k}$, ensuring that the MHE provides the best possible estimate of the system's states and parameters by minimizing the tracking error and respecting the system constraints.

This sequence, $X_{\text{est},-N_{e+k+1}:k}$, is also saved to be used in the cost function formulation in the next iteration.

4.3.5 Estimated Parameters Application

Upon achieving the series of estimated augmented states, the estimated parameters are extracted from one of the arrays within $X_{\text{est},-N_{e+k+1}:k}$. These parameters are then sent to the NMPC to update the subsequent parameters within the prediction model. Additionally, the estimated parameters are stored back in the data buffer to be used for the next iteration.

In cases where there is a lack of knowledge regarding some states, the last element of the estimated sequence, $x_{\text{est},k}$, can be replaced with the missing real state values. This replacement will allow the NMPC to utilize the most accurate available data, aiding the system's control strategy in issuing improved control commands.

4.4 Tools and Implementation

All the necessary procedures required in this study, including data generation, experimentation, testing, coding, and simulation development, were conducted within the TUM-Control environment [53].

TUM-Control is a modular simulation framework for Python designed for ultra-rapid prototyping of self-adaptive, stochastic, and robust Nonlinear Model Predictive Control (NMPC) for Autonomous Vehicle Motion Control. It was developed by the TUM CONTROL Team of the Autonomous Vehicle Systems (AVS) Lab at TUM.

This simulation environment also provides a vehicle dynamics model, specifically designed for TUM's autonomous research vehicle EDGAR. This model served as the foundational knowledge for developing a dynamic model tailored to semi-trucks in this study.

The entire coding for this study was implemented in Python. The model formulation, as well as the formulation of the cost functions and optimization problem solving, were accomplished using Computer Algebra for Systems Analysis and Design Interface (CasADi) [54], ACADOS tools [55], and NumPy [56].

ACADOS is a software package specifically designed for solving optimal control problems, particularly in real-time applications [55]. It provides efficient, high-performance solvers for MPC and MHE, which are

critical for controlling dynamic systems with constraints. Its modular structure supports customization and is well-suited for embedded systems, making it a powerful tool in fields such as robotics, automotive engineering, and aerospace.

CasADi, on the other hand, is a symbolic framework for numerical optimization and automatic differentiation [54]. It enables users to define mathematical models and optimization problems in an intuitive and efficient manner. By leveraging symbolic expressions and sparsity patterns, CasADi excels in generating efficient computational graphs, which are crucial for solving large-scale optimization problems.

The connection between ACADOS and CasADi lies in their complementary roles. CasADi is often utilized to symbolically define the system dynamics, cost functions, and constraints, which are then exported to ACADOS for real-time execution. Together, they create a robust pipeline for designing and implementing advanced control strategies with high precision and efficiency.

In the next chapter, all the developments, including the prediction model, NMPC strategy, and MHE for parameter adaptation to enhance the adaptability of NMPC for semi-trucks, are evaluated, and the results are discussed in detail.

5 Validation and Performance Evaluation

In this chapter, we first validate the developed prediction model for semi-trucks. This is followed by a brief introduction to the track and the simulation environment, demonstrated through an example using EDGAR. Next, we evaluate the standard NMPC developed for the semi-truck. Subsequently, the primary challenge addressed in this study is introduced under the title "The Scenarios and Parameter Mismatch Problem." Finally, a significant portion of this chapter is dedicated to an in-depth evaluation of the performance of the proposed estimation method, developed using the MHE strategy.

5.1 Prediction Model Validation

As described in Section 4.1, the scenario introduced in [29], with minor modifications to the inputs to ensure compatibility with the developed prediction model's inputs, was selected. The validation was conducted using the vehicle and tire parameters specified in [29]. Figure 5.1 shows the results obtained from the developed prediction model alongside their corresponding values reported in [29].

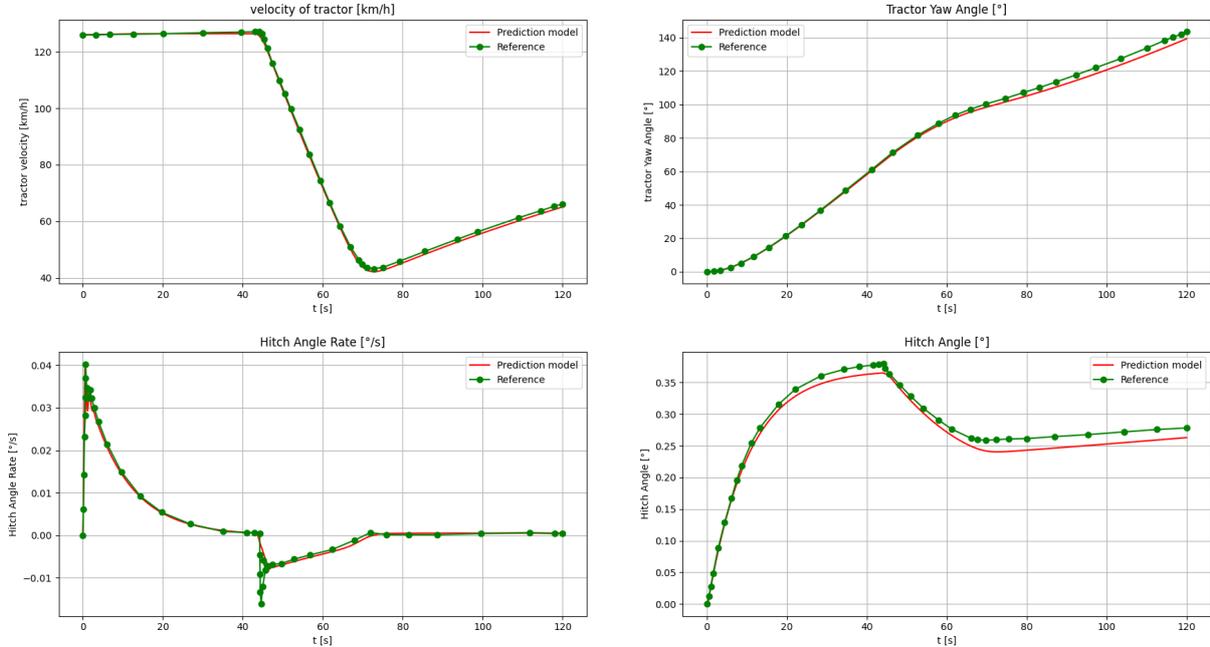


Figure 5.1: Validation results - reference values from [29]

The maximum deviation between the reference and predicted values by the developed model, expressed as a percentage, is reported in Table 5.1. Two primary reasons contribute to these deviations:

Table 5.1: Maximum deviation (Max. Dev.) values regarding the results in Figure 5.1

State	Tractor Velocity [km/h]	Tractor Yaw Angle [deg]	Hitch Angle [deg]	Hitch Angle Rate [deg/s]
Max. Dev. [%]	1.49	3.27	8.18	3.71

The first reason is that in [29], they calculate the lateral tire force using the following equation:

$$F_y = F_y^{\max} \sin[b \tanh(c\alpha)] \quad (5.1)$$

where the quantities $F_y^{\max} = F_y^{\max}(F_z)$, $b = b(F_z)$, and $c = c(F_z)$ are polynomials of the vertical tire load F_z , which varies in the reference study. However, the equations for these quantities were not provided clearly. As a result, the alternative equation based on the linear tire force was used in the developed prediction model.

The second reason lies in the data acquisition procedure used in the reference. Since the reference results in [29] were presented as diagrams rather than numerical values, the data were extracted from the figures in [29], which might have introduced errors.

Despite these reasons, as can be seen in Figure 5.1, the developed prediction model for the semi-truck performs well in the defined scenario. This model will be used in the development of the NMPC and MHE.

5.2 Simulation Environment and Test Track

As introduced in Section 4.4, the TUM-Control simulation environment, developed by the AVS Lab team at TUM, served as the simulation framework for this study. This environment was primarily designed for motion control of TUM's autonomous research vehicle, EDGAR. Figure 5.2 illustrates the TUM-Control simulation environment for EDGAR, which is simplified as a rectangle.

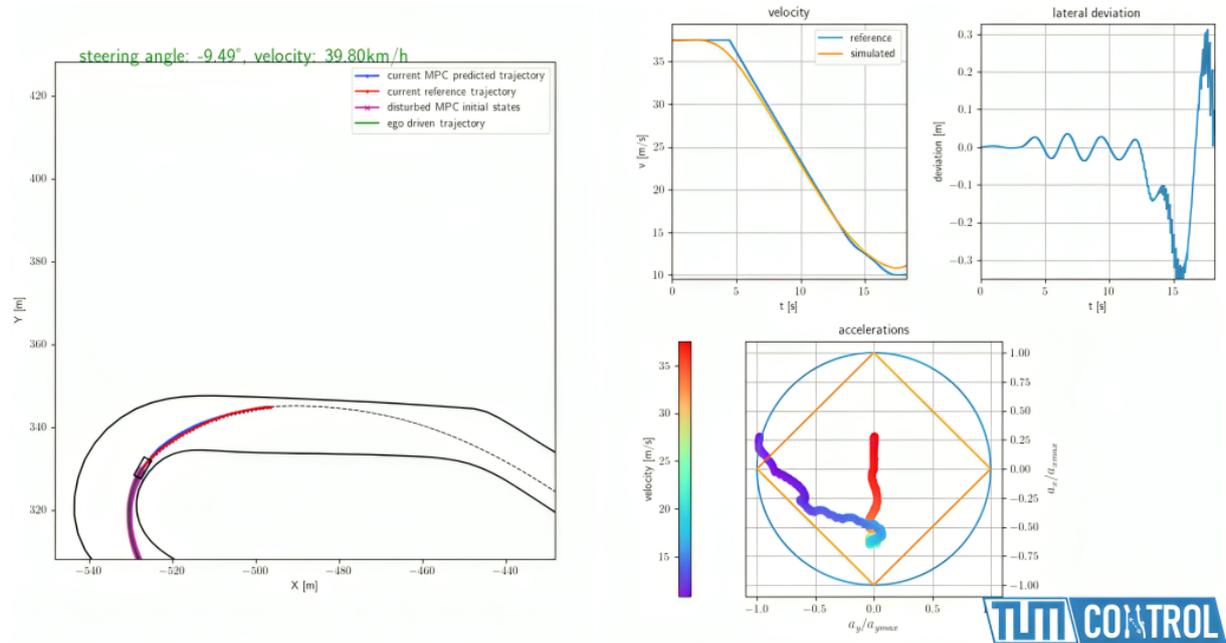


Figure 5.2: TUM-Control simulation environment - example for EDGAR

In the following sections, the environment is adapted and expanded for simulating and testing semi-trailer trucks. This adaptation serves as a foundational step, with significant potential for future enhancements and developments.

All simulations for the semi-truck were conducted on the Montebianco racetrack, depicted in Figure 5.3. The reference trajectory was adopted from the TUM-Control provided trajectories and is specifically designed to test the vehicle at its dynamic limits.

As described in Subsection 4.2.3, testing a vehicle at its dynamic limits requires reliable data, such as the GGK diagram. However, since such data were unavailable for the semi-truck, the trajectory originally generated for EDGAR was used. While using this reference data may lead to suboptimal overall performance, such as increased lateral deviations or higher control command effort, it serves as a reasonable starting point for testing the proposed method.

The simulation begins in the middle of the longest straight section and proceeds clockwise around the track. Each test lasts 133 seconds, corresponding to one full lap.

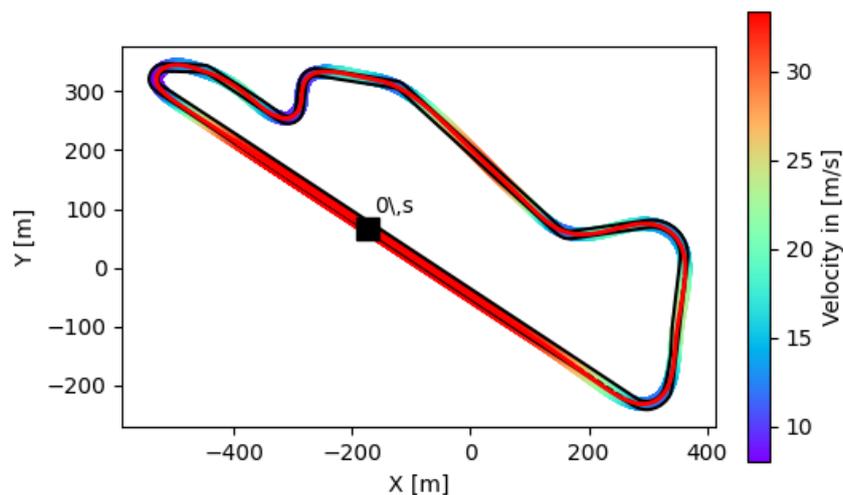


Figure 5.3: Montebianco racetrack with reference trajectory and velocity profile

5.3 Analyzing standard NMPC developed for the semi-truck

Based on the procedure described in Section 4.2, an NMPC for the semi-truck was developed. For the NMPC to function effectively, two key sets of parameters need to be defined and tuned, as they significantly influence the NMPC's performance:

1. **Prediction Horizon (T):** The length of the prediction horizon plays a crucial role in balancing computational load and prediction accuracy. A longer horizon allows the NMPC to consider the vehicle's future behavior over a more extended time but increases computational complexity.
2. **Cost Function Weighting Matrices (Q, R):** These matrices determine the relative importance of minimizing deviations from the reference states and the control effort, respectively. Proper tuning of Q and R is critical for achieving desired tracking performance while avoiding excessive control inputs.

There are different strategies to define the most appropriate and optimized prediction horizon or weighting matrices, including empirical tuning, optimization-based approaches, adaptive methods, sensitivity analysis, reinforcement learning, and H_∞ optimization. These approaches provide systematic ways to balance control performance and computational efficiency while ensuring robustness to system dynamics and constraints [57–59].

However, as the focus of this study is to address the parameter mismatch between the controller and the plant, the aforementioned parameters (T , Q , and R) were defined using a trial-and-error approach. Table 5.2 presents some of the combinations of prediction horizon and cost function weights that were tested during this study. It is important to note that, based on the reasoning provided in Subsection 4.2.1, the weighting

Table 5.2: Tested prediction horizon and cost function weight combinations

Case No.	Prediction Horizon (T)	State Weights (Q)	Input Weights (R)
1	4 sec	Diag([8.8, 8.8, 0.2, 0.0, 0.2])	Diag([710.0, 150.0])
2	2 sec	Diag([8.8, 8.8, 0.2, 0.0, 0.2])	Diag([710.0, 150.0])
3	8 sec	Diag([8.8, 8.8, 0.2, 0.0, 0.2])	Diag([710.0, 150.0])
4	4 sec	Diag([88, 88, 2.0, 0.0, 2.0])	Diag([710.0, 150.0])
5	4 sec	Diag([880, 880, 20, 0.0, 20])	Diag([710.0, 150.0])
6	4 sec	Diag([8.8, 8.8, 0.2, 0.0, 0.2])	Diag([71.0, 15.0])
7	4 sec	Diag([8.8, 8.8, 0.2, 0.0, 0.2])	Diag([7100.0, 1500.0])

factor in Q corresponding to ref_θ is always set to zero. This decision ensures that the optimization problem does not prioritize minimizing deviations in the reference hitch angle, as no reliable reference data for this parameter is available. By excluding this term from the cost function, the focus remains on improving other key parameters for which reliable reference values exist, thereby avoiding unnecessary computational complexity or optimization inaccuracies.

Nevertheless, the maximum and minimum constraints for the hitch angle are still enforced within the optimization problem. These constraints ensure that the hitch angle remains within physically meaningful and safe limits, preserving the feasibility of the solution while preventing potentially unsafe or impractical configurations. This approach balances the trade-off between excluding unreliable reference data and ensuring the physical realism of the system.

If reliable reference hitch angle data becomes available in the future, this weighting factor can be defined and appropriately tuned to further enhance performance. Incorporating a well-defined reference for the hitch angle into the optimization problem would allow the NMPC to achieve more precise control, particularly during scenarios involving sharp turns or complex maneuvers where hitch angle accuracy becomes critical.

The results from these tests provided insights into how variations in these parameters affect NMPC performance. Considering the results depicted in Figure 5.4, four key metrics were evaluated to guide the selection of the best parameter combination from Table 5.2. These metrics include:

1. Average and maximum absolute lateral deviation.
2. Average and maximum absolute speed deviation from the reference speed.
3. Average and maximum absolute applied steering rate.
4. Average and maximum absolute applied jerk.

The primary objective of implementing the NMPC is to achieve the lowest possible state tracking deviation while minimizing command effort. This ensures that the vehicle closely follows the reference trajectory and speed while maintaining smooth and realistic control actions. Based on these criteria, Case Number 1 was selected as the most suitable parameter combination for this study. Case Number 1 was chosen because it consistently demonstrated superior performance across all evaluated metrics, balancing state tracking accuracy with control effort.

It is important to emphasize that the primary focus of this research is addressing the parameter mismatch problem. Consequently, an extensive exploration of all possible parameter combinations was beyond the scope of this study. Instead, a limited yet representative selection of parameter sets was analyzed to identify an optimal configuration for the NMPC under the given conditions.

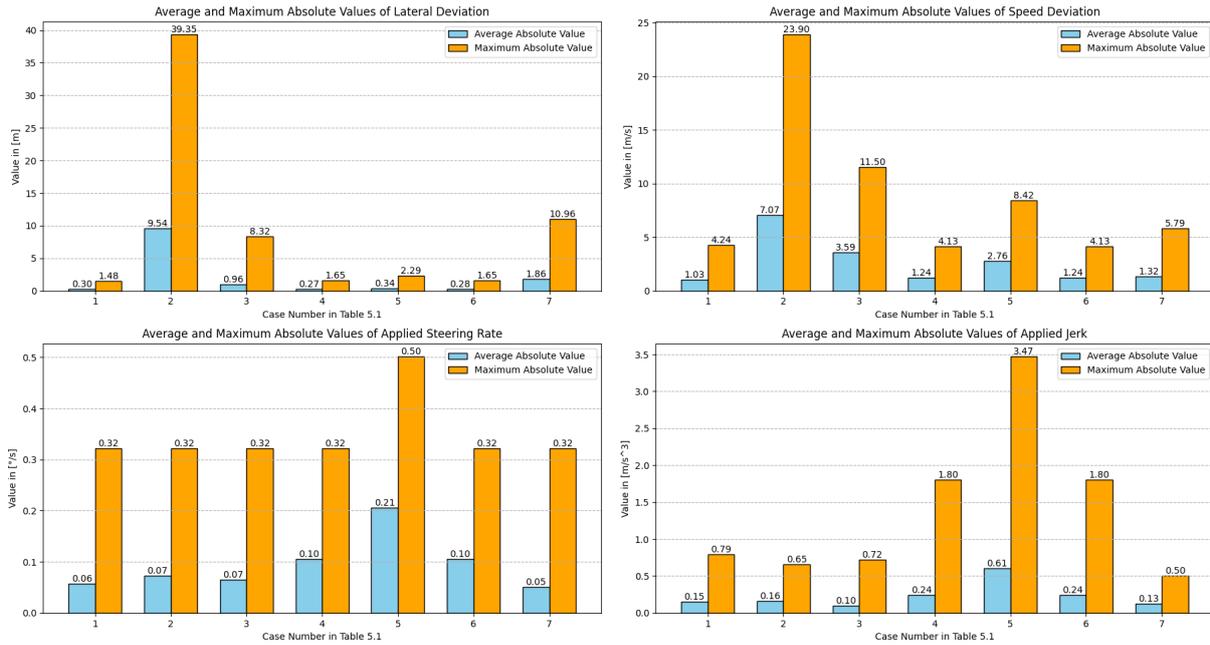


Figure 5.4: Performance results of developed NMPC with different combinations in the Table 5.2

In Figure 5.5, the TUM-Simulation Environment developed for the semi-truck is illustrated. This simulation environment was specifically designed to replicate realistic vehicle dynamics and race track conditions, providing a robust testing platform for the NMPC implementation. Additionally, Figure 5.6 presents the results of the states and variables with constraints during the simulation of the semi-truck motion on the Monteblando race track, as described in Subsection 4.2.3.

The simulation results using the parameter combination from Case 1 clearly show that the NMPC successfully satisfies all constraints, including limits on steering angle and its rate, hitch point angle, and combined longitudinal and lateral acceleration, based on the GGV diagram. This outcome further justifies the selection of Case 1 as the optimal configuration, as it minimizes state tracking errors while ensuring that the vehicle operates within safe and realistic physical limits.

By employing this approach, the study demonstrates the effectiveness of the developed NMPC in achieving reliable and accurate control of the semi-truck under challenging conditions while addressing the parameter mismatch problem.

5.4 The Scenarios and Parameter Mismatch Problem

To address the parameter mismatch issue between the controller and the plant, four scenarios are considered:

1. The NMPC is initialized with the mass and inertia of the trailer (the mutable parameters in this study) set to three times higher than the real parameters, as defined in Table 3.2, in the plant. This scenario assumes no disturbance in the system.
2. The NMPC is initialized with the mutable parameters set to three times lower than their real values, again assuming no disturbance.
3. Similar to the first scenario, but with light disturbance introduced into the system.
4. Similar to the second scenario, but with light disturbance introduced into the system.

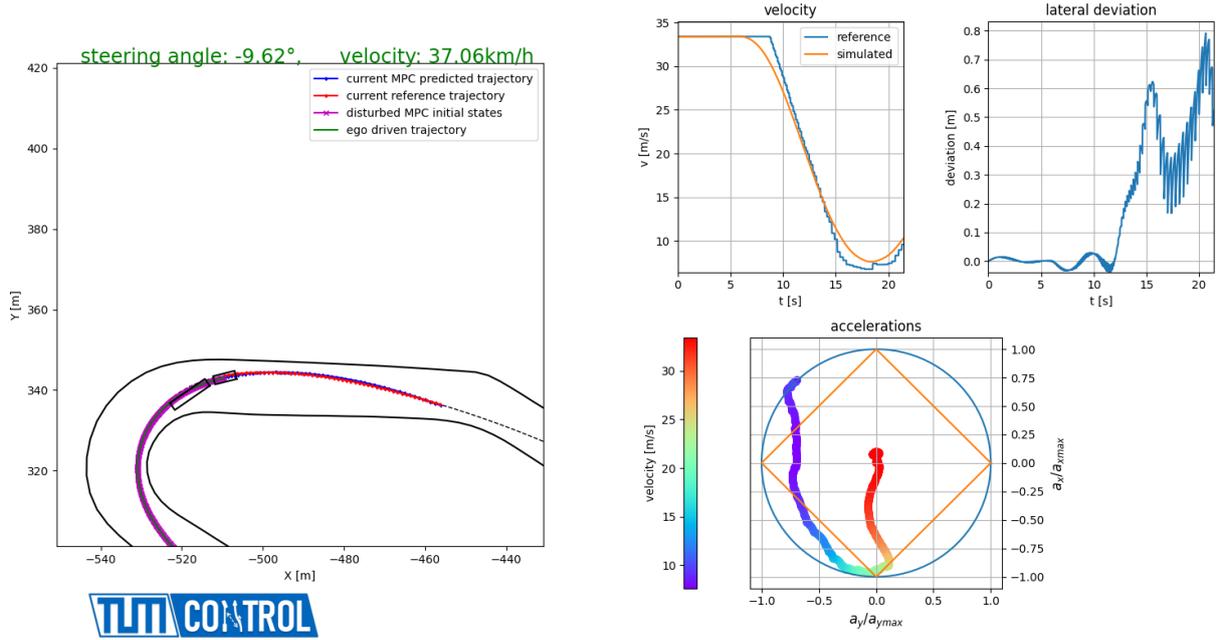


Figure 5.5: TUM-Control developed simulation environment for Semi-truck

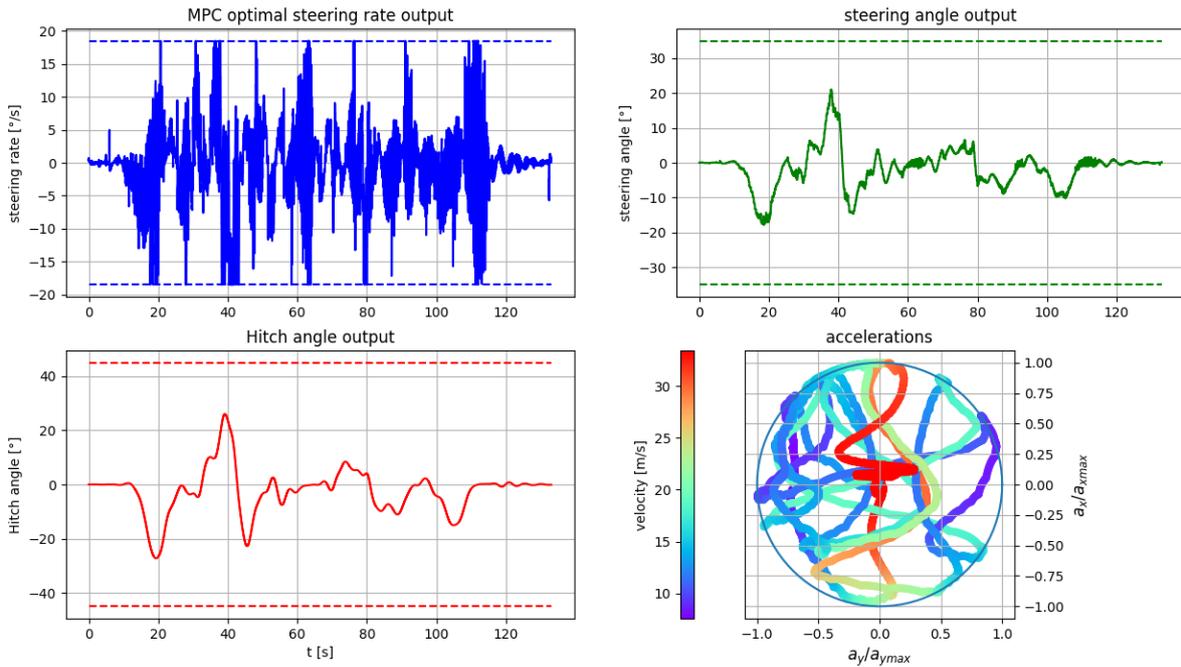


Figure 5.6: Semi-truck's constrained parameters results when Case 1 setting utilized

For all scenarios, the rest of the vehicle parameters in the NMPC are set to match the real values. These scenarios are summarized in Table 5.3. To simulate disturbances acting on the vehicle, we add Gaussian

Table 5.3: Summary of the parameter mismatch scenarios

Scenario No.	$m_2 [kg]$	$I_{z2} [kg.m^2]$	With Disturbance
1	111765	2101506.0	×
2	12418.3	233500.7	×
3	111765	2101506.0	✓
4	12418.3	233500.7	✓

noise to the derivatives of the continuous simulation model (process noise):

$$0 = f_{\text{cont}}^{\text{sim}}(\dot{x}, x, u) + w \quad (5.2)$$

The distribution of the process noise w over time obeys the following standard deviations, chosen according to the disturbance realizations implemented in [53]:

$$\sigma_w = \begin{bmatrix} \sigma_{\dot{x}_{\text{pos}}} \\ \sigma_{\dot{y}_{\text{pos}}} \\ \sigma_{\dot{\psi}} \\ \sigma_{\dot{\theta}} \\ \sigma_{\dot{v}_{\text{lon}}} \\ \sigma_{\dot{v}_{\text{lat}}} \\ \sigma_{\ddot{\psi}} \\ \sigma_{\ddot{\theta}} \\ \sigma_{\dot{\delta}_f} \\ \sigma_{\dot{a}} \end{bmatrix} = \begin{bmatrix} 0.8 \text{ m/s} \\ 0.8 \text{ m/s} \\ 0.1 \text{ rad/s} \\ 0.1 \text{ rad/s} \\ 1.1 \text{ m/s}^2 \\ 0.1 \text{ m/s}^2 \\ 0.05 \text{ rad/s}^2 \\ 0.05 \text{ rad/s}^2 \\ 0.1 \text{ rad/s} \\ 0 \text{ m/s}^3 \end{bmatrix} \quad (5.3)$$

Finally, the state measurement and state estimation process must be modeled with its uncertainties and disturbances. This is realized by adding Gaussian noise v_k (measurement noise) to every result of the prediction model, which is computed by utilizing the standard deviation vector σ_v :

$$x_{\text{disturbed},k+1} = x_{k+1} + v_k \quad (5.4)$$

$$\sigma_v = \begin{bmatrix} \sigma_{x_{\text{pos}}} \\ \sigma_{y_{\text{pos}}} \\ \sigma_{\psi} \\ \sigma_{\theta} \\ \sigma_{v_{\text{lon}}} \\ \sigma_{v_{\text{lat}}} \\ \sigma_{\dot{\psi}} \\ \sigma_{\dot{\theta}} \\ \sigma_{\delta_f} \\ \sigma_a \end{bmatrix} = \begin{bmatrix} 0.05 \text{ m} \\ 0.05 \text{ m} \\ 0.005 \text{ rad} \\ 0.005 \text{ rad} \\ 0.2 \text{ m/s} \\ 0.04 \text{ m/s} \\ 0.02 \text{ rad/s} \\ 0.02 \text{ rad/s} \\ 0.002 \text{ rad} \\ 0 \text{ m/s}^2 \end{bmatrix} \quad (5.5)$$

It is important to note that the defined disturbance is relatively low, and the primary goal is to demonstrate the effect of the existence of disturbance on the performance of the controller and the estimator. To handle larger disturbances, it is recommended to employ robust methods, such as RMPC or SMPC, which are beyond the scope of this study.

The consideration of disturbances results in two distinct vectors for each state:

1. **Measured State:** This state is subject to all disturbances and represents the measured data known to the vehicle. The parameter estimator, as well as the NMPC, only receives the measured state vector.
2. **True State:** This state is only affected by the disturbances imposed on the derivatives, representing the simulated true vehicle behavior. While the true state is not known to the vehicle, it can be used for analysis and assessment purposes, such as open-loop predictions

in the evaluation discussed in Section 5.5. Using the true state is essential in such cases since the prediction of the true vehicle behavior, originating from the true state, needs to be assessed.

In the following sections, unless specified otherwise, references to the state always pertain to the **measured state**.

Figures 5.7 and 5.8 depict the four key metrics, previously reported in Figure 5.4, for all scenarios described in Table 5.3, implemented without any estimation method. Figure 5.7 corresponds to scenarios without disturbances, while Figure 5.8 presents the results for scenarios with system disturbances.

One of the first important observations from these figures is the increase in all key metrics when moving from the absence of disturbances to the presence of disturbances. This trend is particularly noticeable when comparing the no-mismatch cases with and without disturbances.

In Figure 5.7, the degradation in NMPC performance due to parameter mismatch is clearly evident, especially in the average and maximum absolute values of lateral deviation and speed deviation. For example, in Scenario 2, the average and maximum absolute lateral deviation values increased by approximately 4.5 and 8 times, respectively. These results emphasize the critical need for a parameter adaptation mechanism within NMPC to maintain its performance under conditions of parameter mismatch. In this study, this issue is addressed through the implementation of the MHE method, which allows for the continuous adaptation of model parameters.

Figure 5.8 illustrates that the presence of system disturbances introduces unpredictability in the results, particularly in cases with mismatch. Despite this, the degradation in NMPC performance is still observable, especially in terms of lateral deviation, which is the most significant parameter in path-tracking analysis. The disturbances lead to an increase in the magnitude of deviations, demonstrating the challenges in maintaining stable and accurate path tracking under such conditions.

Figure 5.9 consolidates the lateral deviation results from Figures 5.7 and 5.8 for all scenarios into a single comparison, providing a clearer overview of the trends across all test conditions. Additionally, Figure 5.10 presents an example of the lateral deviation from the reference trajectory for Scenario 1, highlighting the deviations in two sections of the track.

5.5 Moving Horizon Estimation (MHE) Evaluation

Now, it is time to evaluate the estimation method presented in the previous chapter across all the assumed scenarios. This evaluation begins with an open-loop performance evaluation, followed by a closed-loop analysis. Additionally, the influence of reducing the available state information is examined in excerpts to gain a deeper understanding of the estimation process.

To determine how to assess the quality of the estimation method, it is necessary to revisit the purpose of the estimation process. Although the primary goal of the MHE method is to estimate the mutable parameters, the main and more significant objective is to assist the NMPC in achieving better prediction quality. Thus, the accuracy of the parameters is of lesser importance, particularly since a correct parameter set might not even exist. Different parameter sets might yield similarly good results. Furthermore, except for cases involving simulated data, a ground truth parameter set might not even be available, making the direct assessment of parameter estimates a poor evaluation method. Therefore, the primary focus should be on the predictions of the dynamics model utilizing the respective estimated parameters. Nevertheless, inspecting the estimated parameter sets can still provide insightful information.

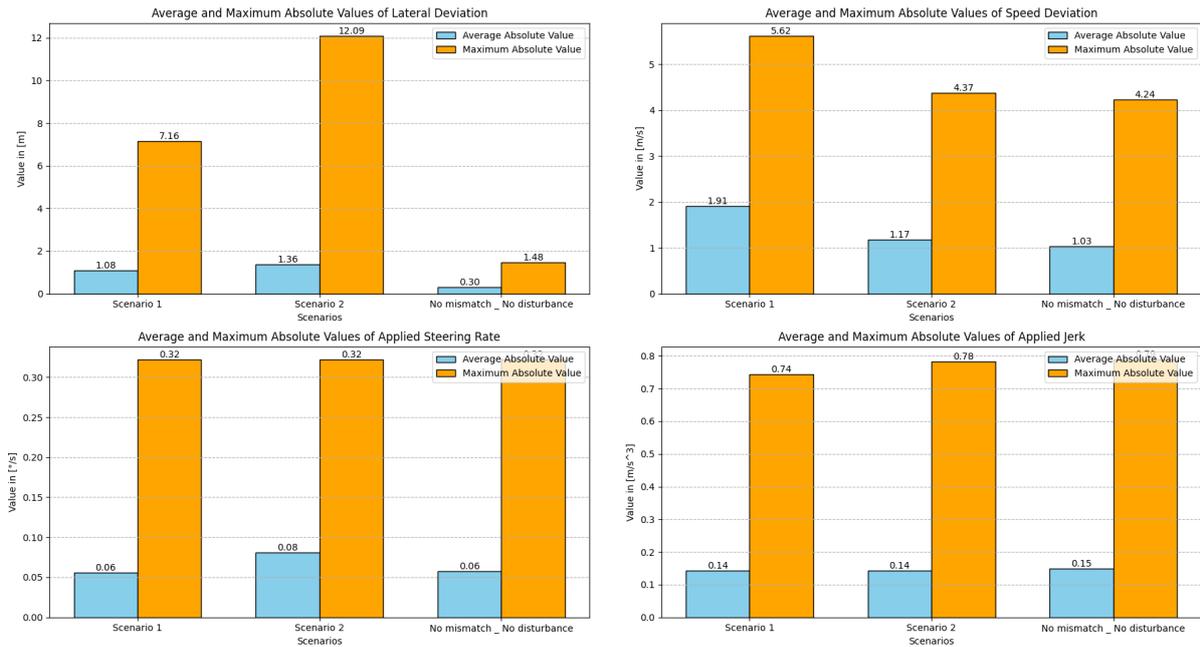


Figure 5.7: NMPC performance results under Scenario 1 and 2

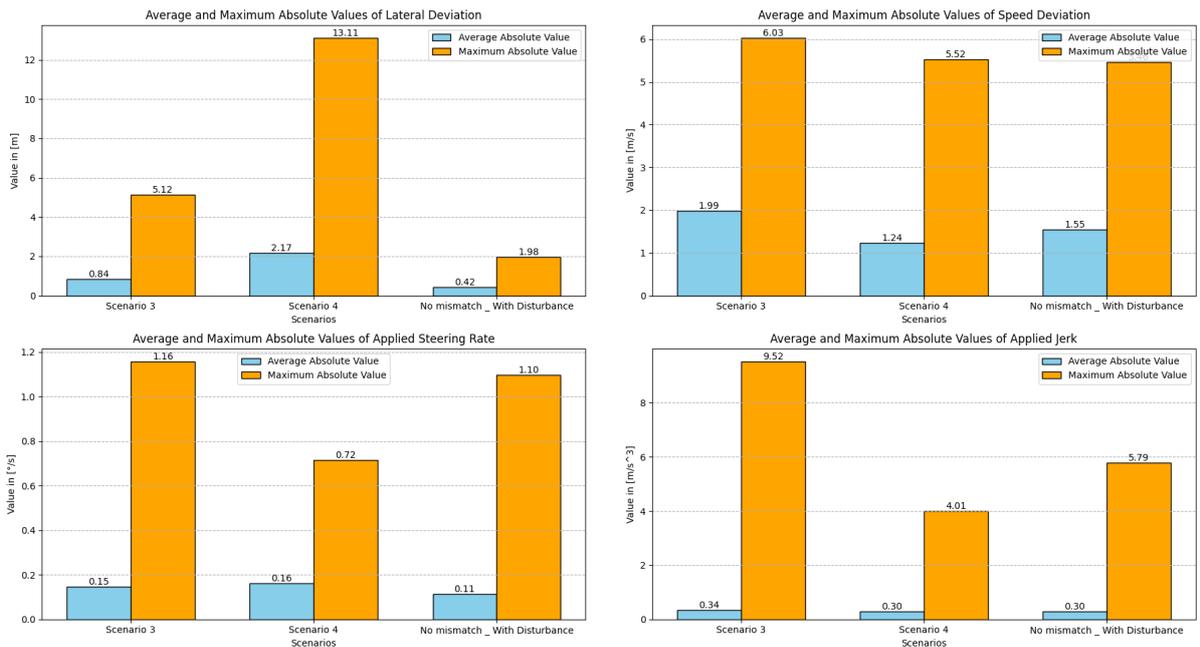


Figure 5.8: NMPC performance results under Scenario 3 and 4

With this in mind, two types of tests are conducted:

- Open-loop tests (Subsection 5.5.1):** These tests are analogous to driving in ghost mode. The estimator is continuously fed data points from a simulated, pre-driven trajectory, which is subject to different environmental changes and disturbances. The estimator returns corresponding parameter estimates, which do not influence the trajectory. These estimates are then analyzed by using them to make open-loop predictions over a duration T (the prediction horizon), originating from the pre-driven trajectory, just as the NMPC would. The key difference is that we do not search for control inputs optimizing the vehicle behavior but instead employ the known control inputs. In the optimal case, the resulting predictions align

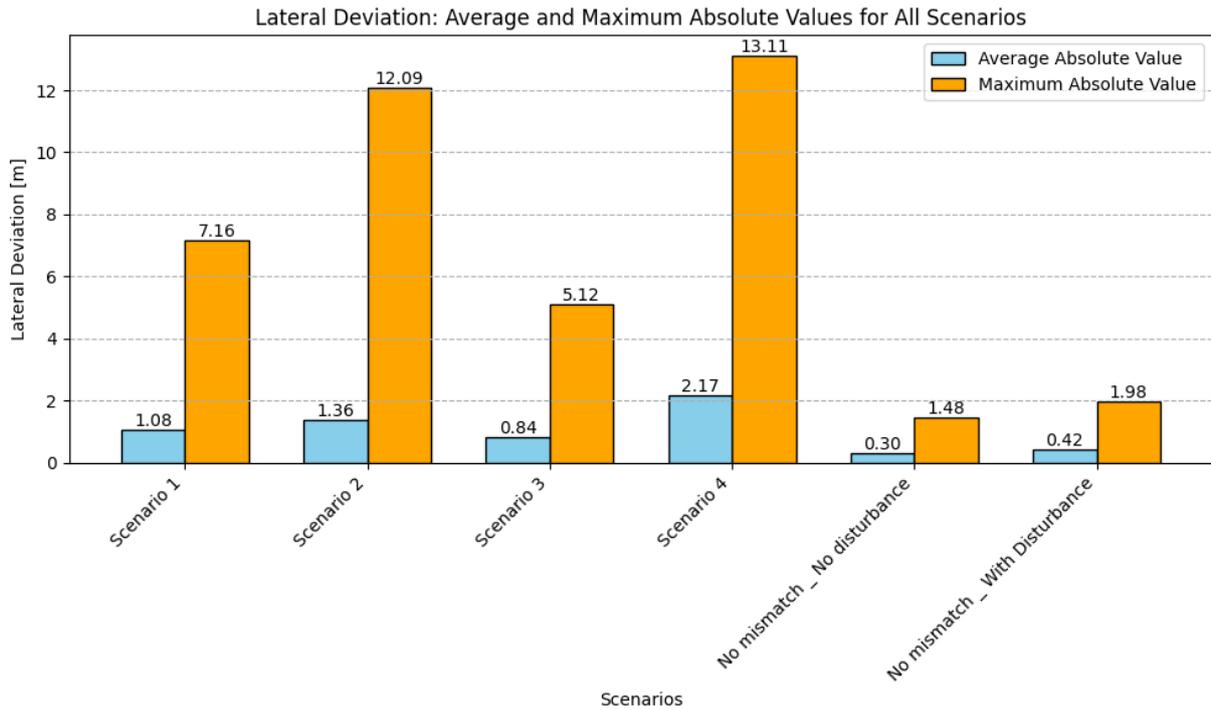


Figure 5.9: Lateral deviation average and maximum absolute values for all scenarios

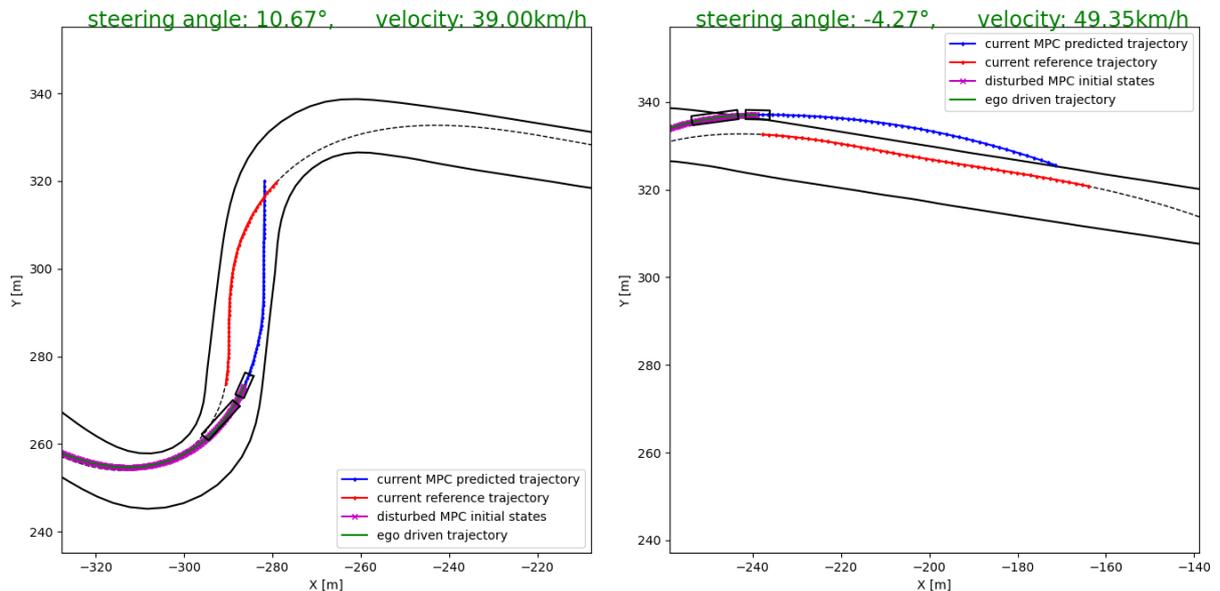


Figure 5.10: Lateral deviation from reference trajectory for scenario 1 / Red line : Reference trajectory in the prediction horizon / Blue line: Predicted trajectory with NMPC

perfectly with the original trajectory. Consequently, the deviation of these predictions from the continuation of the simulated trajectory is assessed.

- Closed-loop tests (Subsection 5.5.2):** These tests involve exposing the developed standard NMPC in this study to parameter mismatch scenarios on a simulated racetrack while being provided with a parameter estimator that adapts the underlying prediction model. The path-tracking performance of the NMPC is then analyzed to evaluate the effectiveness of the estimation method.

These tests, based on the scenarios summarized in Table 5.3, can be executed for estimators with different configurations to contrast their estimation performance. The specific methods for assessing performance depend on the type of test and are thus presented in their respective sections. However, all assessments adhere to the considerations outlined above, emphasizing the focus on the estimation method's contribution to improving prediction quality rather than solely on parameter correctness.

5.5.1 Open-loop tests

This section explains the open-loop testing method, which evaluates parameter estimates derived from a simulated, pre-driven trajectory. The parameter estimator continuously receives data points from pre-driven trajectories based on the scenarios explained in Section 5.4, where no adaptation method is used. The objective is to obtain a dynamically interesting trajectory of states under specific control inputs, disturbances, and environmental changes. This trajectory provides the basis for analyzing the quality of the parameter estimates and their impact on the prediction performance of the developed model.

For each estimated parameter set \hat{p}_k , an open-loop prediction of duration $T = 4\text{s}$ is conducted, starting from the corresponding true state x_k . This involves applying the parameter set \hat{p}_k and a series of control inputs $U_{k:k+S-1}$ to produce the simulated trajectory. The number of open-loop prediction steps, S , is computed as:

$$S = \frac{T}{T_d} = 200 \quad (5.6)$$

where the simulated trajectory is discretized with a sampling interval of $T_d = 20\text{ ms}$. The predicted trajectory is generated with the same discretization time. As a result, each parameter estimate \hat{p}_k is assigned a predicted state series $\tilde{X}_{k+1:k+S}$ containing S elements. This series is compared to the true state series $X_{k+1:k+S}$ from the original trajectory.

The deviation between the two state series is quantified using the Average Relative State Error (ARSE) metric, with a specific focus on the Average Relative Distance Error (ARDE). The error for each step, ϵ_k , is then calculated.

To ensure comparability across parameter estimates, the analysis only considers the first 129 s of the parameter estimates from the simulated trajectory. This approach excludes the last 4 s where data is naturally unavailable for evaluation.

A key advantage of this testing method is its ability to ensure fair comparability in the prediction performance. Every tested estimator is subjected to the same disturbances and provided with identical data points from the same trajectory. For closed-loop tests, this consistency is less straightforward since the trajectory depends on the issued estimations. Consequently, an NMPC equipped with one estimator might enter a specific realm of the trajectory, while another might not. This difference can result in biased performance comparisons. The open-loop method avoids such inconsistencies, making it a more reliable evaluation framework.

Average Relative State Error (ARSE) metric

In order to compute the error ϵ_k from the difference between the predicted state series, $\tilde{X}_{k+1:k+S}$, and the simulated series $X_{k+1:k+S}$ of true states, the well-known error metric, Mean Squared Error (MSE), is formulated as:

$$\epsilon_k = \frac{1}{S} \sum_{\kappa=1}^S \|\tilde{x}_\kappa - x_\kappa\|_2^2 \quad (5.7)$$

where every point of the original trajectory is compared to its corresponding point in the predicted one.

However, this metric, in its plain form, has two main issues:

1. The plain MSE formulation directly compares state variables of different units and different orders of magnitude. The behavior of the error is therefore not well-defined and can become unclear.
2. Deviations at the end of the predicted trajectory are typically much higher than those at the beginning. Since the MSE squares the deviations, it disproportionately emphasizes the final points of the trajectory. This conflicts with the requirement for a metric that compares the entire trajectory more evenly.

To address these issues, every computed norm inside the summation is normalized. Additionally, the square is removed to make the meaning more intuitive and to equalize the contribution of each trajectory point. The revised error metric is defined as:

$$\epsilon_k = \frac{1}{S} \sum_{\kappa=1}^S \frac{\|\tilde{x}_\kappa - x_\kappa\|_2}{\|x_\kappa - x_\kappa\|_2} \quad (5.8)$$

where \tilde{x}_κ and x_κ are the predicted state vector and simulated state vector at each step inside the estimation horizon, and x_κ is the simulated state vector at the beginning of the estimation horizon.

This approach addresses both issues. The normalization makes the state values unitless and neutralizes the effect of differences in magnitude. At the same time, normalizing by the driven simulated state values mitigates the impact of larger deviations at the end of the estimation horizon, ensuring that every part of the trajectory contributes equally to the error.

The error can now intuitively be understood as the deviation in state values per driven simulated state values. To prevent the risk of singularity in the calculation of Equation 5.8, we introduce a minimum normalization vector D_{\min} , which is set to a vector with all elements equal to 1. The final error metric is then defined as:

$$\epsilon_k = \frac{1}{S} \sum_{\kappa=1}^S \frac{\|\tilde{x}_\kappa - x_\kappa\|_2}{\max(\|x_\kappa - x_\kappa\|_2, D_{\min})} \quad (5.9)$$

Since this metric is defined as the average of the differences between predicted and original states relative to the respective driven simulated state values, it is referred to as the Average Relative State Error, or in short ARSE.

In this study, since the primary focus is on trajectory following, a specialized form of ARSE, the ARDE, is emphasized. This metric combines the longitudinal and lateral position states to focus on positional accuracy. The ARDE is computed as follows:

$$\epsilon_{D,k} = \frac{1}{S} \sum_{\kappa=1}^S \frac{\left\| \begin{bmatrix} \tilde{x}_{\text{pos},\kappa} \\ \tilde{y}_{\text{pos},\kappa} \end{bmatrix} - \begin{bmatrix} x_{\text{pos},\kappa} \\ y_{\text{pos},\kappa} \end{bmatrix} \right\|_2}{\max\left(\left\| \begin{bmatrix} x_{\text{pos},\kappa} \\ y_{\text{pos},\kappa} \end{bmatrix} - \begin{bmatrix} x_{\text{pos},\kappa} \\ y_{\text{pos},\kappa} \end{bmatrix} \right\|_2, D_{\min}\right)} \quad (5.10)$$

where, $\tilde{x}_{\text{pos},\kappa}$ and $\tilde{y}_{\text{pos},\kappa}$ are the predicted longitudinal and lateral positions at step κ , $x_{\text{pos},\kappa}$ and $y_{\text{pos},\kappa}$ are the simulated longitudinal and lateral positions at step κ , $x_{\text{pos},\kappa}$ and $y_{\text{pos},\kappa}$ are the longitudinal and lateral positions at the beginning of the prediction horizon.

Main Estimator Configuration

Based on the procedure of the MHE method development, as described in Section 4.3, for the MHE to work effectively and suitably, two key sets of parameters, similar to those in NMPC, need to be defined and tuned. These parameters significantly affect the MHE's performance:

1. **Estimation Horizon (T_{est}):** Similar to the prediction horizon but focused on the vehicle's past behavior, the estimation horizon determines how much historical data is utilized for the estimation process. The length of the estimation horizon plays an important role in balancing computational load and estimation accuracy.
2. **Cost Function Weighting Matrices (Q, Q_{mhe0}):** These matrices determine the relative importance of minimizing deviations from reference values. Q assigns importance to individual state variables, allowing the estimation process to prioritize certain states over others. Q_{mhe0} acts as a penalty term to discourage the estimated states at the initial stage of the estimation horizon from deviating significantly from their prior estimated values, ensuring stability and continuity in the estimation process.

As mentioned in Section 5.3, there are systematic methods available to assign appropriate values to T_{est} , Q , and Q_{mhe0} . However, since the primary focus of this study is on analyzing the effect of using the estimation method on the controller's performance and not on the tuning process itself, these parameters are selected using a trial-and-error approach. This approach is applied during the implementation of the first scenario from Table 5.3, with different parameter set combinations summarized in Table 5.4. The best configuration is selected based on the open-loop performance and the total runtime of the open-loop estimation procedure.

Table 5.4: Tested estimation horizon and cost function weight combinations

Config. No.	T_{est}	Weighting Matrices
1	1 sec	$Q = \text{Diag} [10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4]$ $Q_{\text{mhe0}} = \text{Diag} [10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^{-5}, 10^{-7}]$
2	0.2 sec	$Q = \text{Diag} [10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4]$ $Q_{\text{mhe0}} = \text{Diag} [10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^{-5}, 10^{-7}]$
3	0.5 sec	$Q = \text{Diag} [10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4]$ $Q_{\text{mhe0}} = \text{Diag} [10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^{-5}, 10^{-7}]$
4	0.5 sec	$Q = 2 \times \text{Diag} [10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4]$ $Q_{\text{mhe0}} = \text{Diag} [10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^{-5}, 10^{-7}]$
5	0.5 sec	$Q = \text{Diag} [10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4]$ $Q_{\text{mhe0}} = \text{Diag} [10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^4, 10^{-4}, 10^{-6}]$

Table 5.5: Total open-loop estimation runtime in [s] for all configurations in Table 5.4.

Config. No.	1	2	3	4	5
Runtime [s]	2069	76	599.5	584.5	479

The parameter combinations listed in Table 5.4 were tested using the first scenario, yielding results for the ARDE metric (Figures 5.11 and 5.12) and the total open-loop estimation runtime (Table 5.5). These results are analyzed to identify the most effective configuration for the MHE method. By examining Figure 5.11, which presents the ARDE values for all configurations alongside the default case of Scenario 1 (without employing any estimation method), it is evident that all tested configurations significantly reduce the relative distance error compared to the default case. This reduction highlights the effectiveness of the MHE method in improving prediction accuracy.

To facilitate the selection of the best configuration, Figure 5.12 is used, which depicts the box plot of the ARDE metric. This plot provides a more detailed statistical comparison by illustrating the maximum and

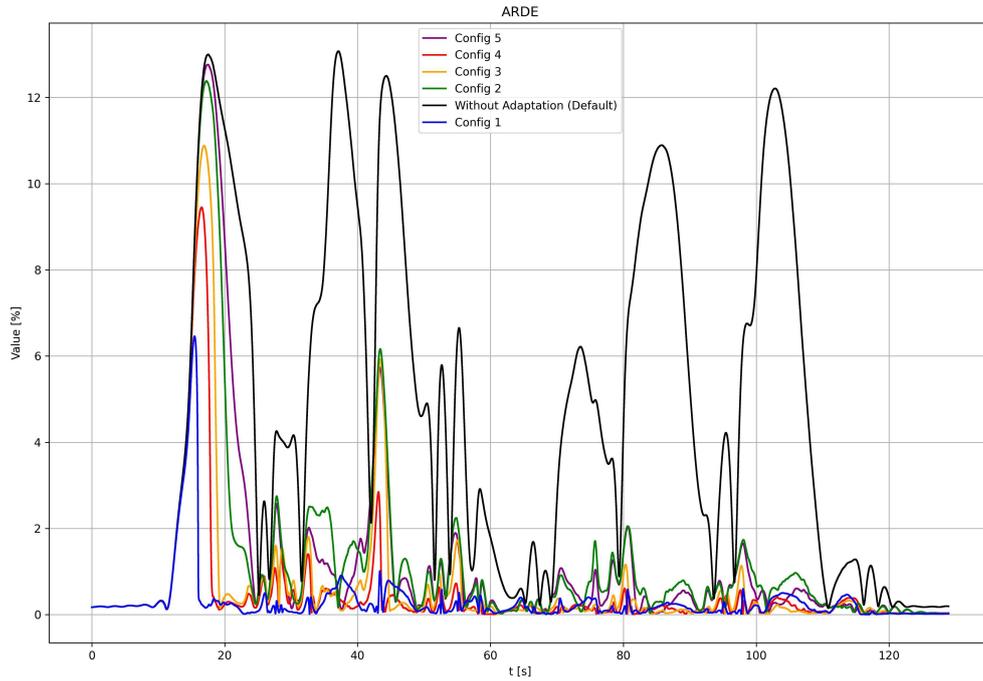


Figure 5.11: ARDE in [%] for all configurations in Table 5.4 and the default parameter set of the first scenario without using MHE.

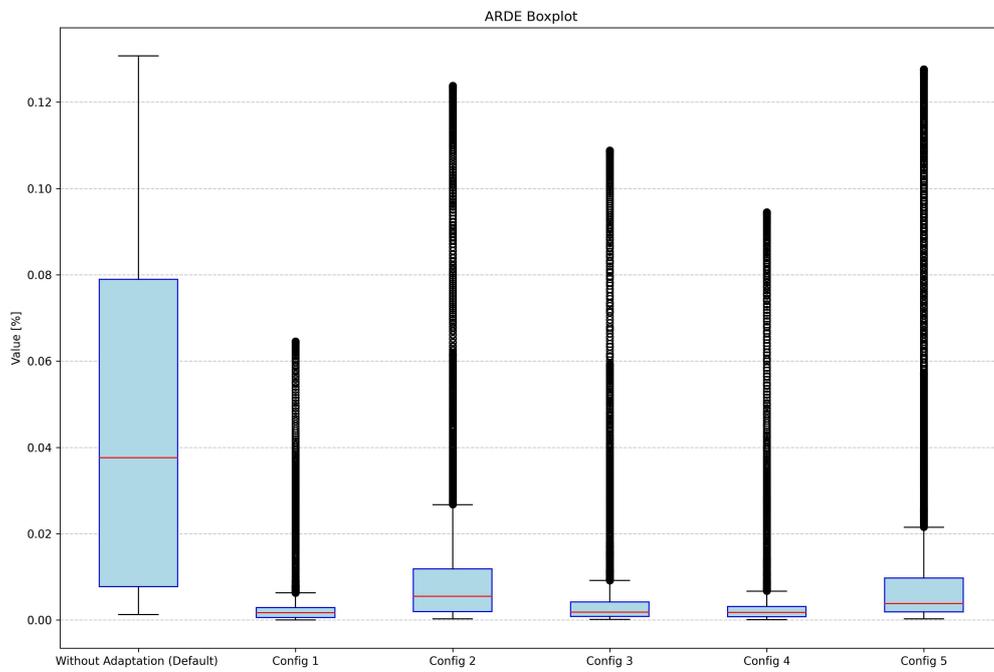


Figure 5.12: Box plot of ARDE results for all configurations in Table 5.4 and the default parameter set of the first scenario without using MHE.

median ARDE values for each configuration over the simulation time span. Upon closer inspection of the box plot, it becomes apparent that two configurations—Configuration 1 and Configuration 4—consistently outperform the others in terms of reducing the maximum and median ARDE values.

Among these two, Configuration 1 achieves the lowest ARDE values, indicating superior performance in reducing the open-loop relative distance error. This enhanced performance can be attributed to its longer estimation horizon, which leverages a more extensive history of state information. By incorporating more historical data, Configuration 1 is able to better capture the underlying system dynamics, leading to improved accuracy in state estimation.

However, when the total runtime values reported in Table 5.5 are considered, Configuration 1 demonstrates a significant computational cost. With a runtime of 2069 seconds, it is 3.5 times slower than Configuration 4. This highlights a key trade-off: while longer estimation horizons improve estimation accuracy, they also increase computational complexity, leading to larger optimization problems that challenge real-time feasibility.

Given these observations, Configuration 4 strikes an optimal balance between accuracy and computational efficiency. Although its ARDE performance is slightly inferior to that of Configuration 1, it still achieves substantial error reduction while maintaining a runtime that is more feasible for real-time applications. Consequently, Configuration 4 is selected as the primary parameter set for the MHE method used throughout the remainder of this study.

Performance Evaluation

Equipped with the ARSE metric, we now evaluate the prediction performance of the MHE-developed estimator using the selected configuration under the four pre-mentioned scenarios. The results for each scenario are depicted in Figures 5.13 to 5.16. All new abbreviations that appear in the subsequent figures are as follows:

- Average Relative Orientation Error (AROE)
- Average Relative Hitch Angle Error (ARHAE)
- Average Relative Longitudinal Velocity Error (ARLOVE)
- Average Relative Lateral Velocity Error (ARLAVE)
- Average Relative Yaw Rate Error (ARYRE)

It is important to note that while other states also have relative error metrics, our primary focus is on trajectory following. Consequently, we are more interested in ARDE values and have opted not to include the results of other state errors in this section.

By examining Figures 5.13 to 5.16, the first noticeable feature is the behavior of the blue curves, which represent the prediction relative errors when using the default scenario parameter sets without employing any estimation method. The spikes in these curves align precisely with the track bends, as shown in Figure 5.3. This alignment suggests that high prediction errors, resulting from incorrect parameter sets, predominantly occur during lateral vehicle maneuvers. Such observations underline the importance of bends or curvy driving maneuvers as valuable sources of information for model adaptation. This is because parameter estimation relies on assessing deviations between the actual series of states driven by the vehicle and the predicted ones based on the current parameter set.

This conclusion becomes even more pronounced in scenarios without system disturbances (scenarios 1 and 2). During the initial phase of these simulations (approximately the first ten seconds), when the track is straight, the prediction errors remain relatively small. Figure 5.17, which focuses on the ARSE of scenario 1 during the first twenty seconds, further substantiates this observation.

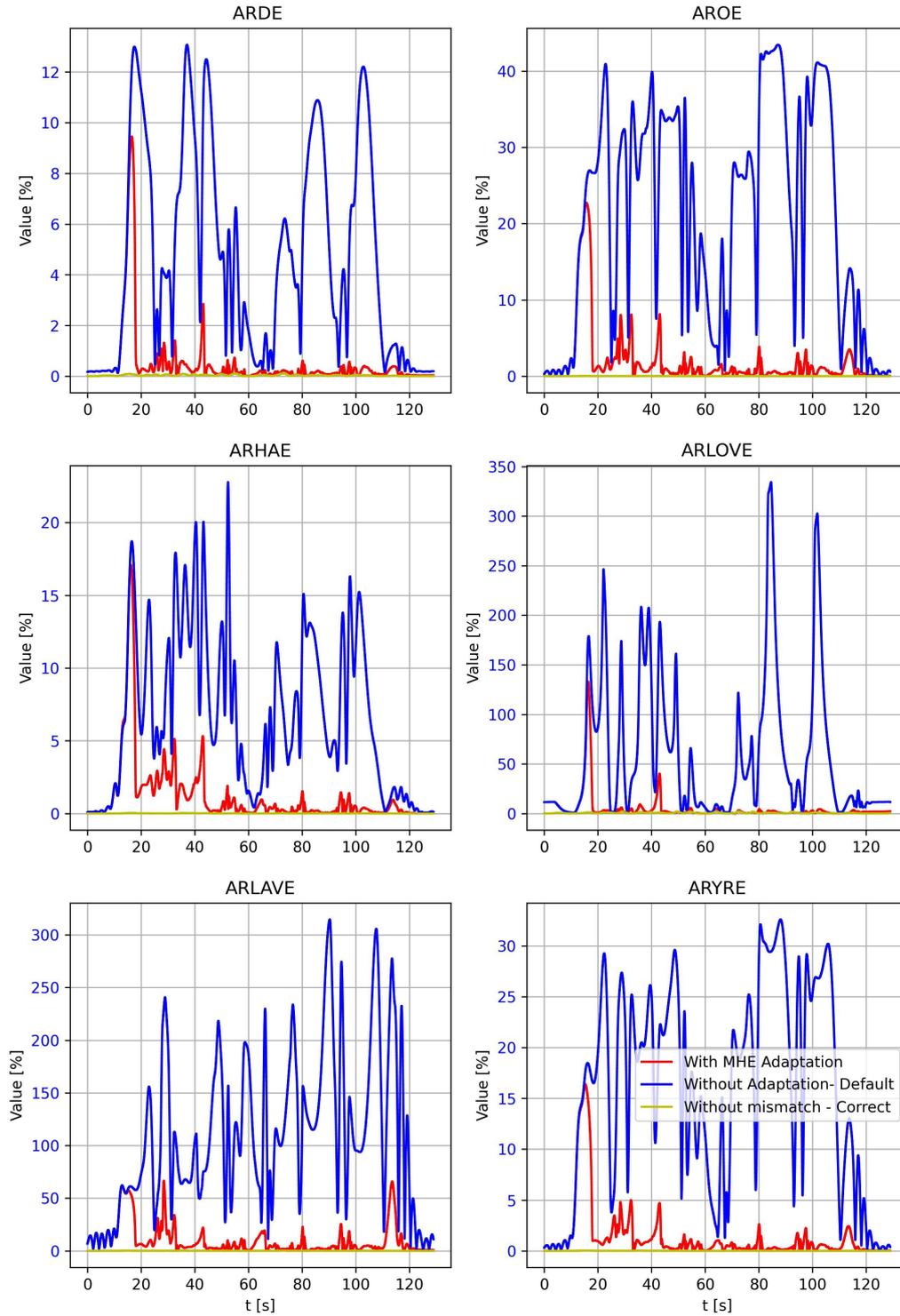


Figure 5.13: ARSE in [%] for Scenario 1/ Red line: Using MHE with selected configuration/ Blue line: With mismatch and No Adaptation (Default) and Yellow line: No mismatch (Correct).

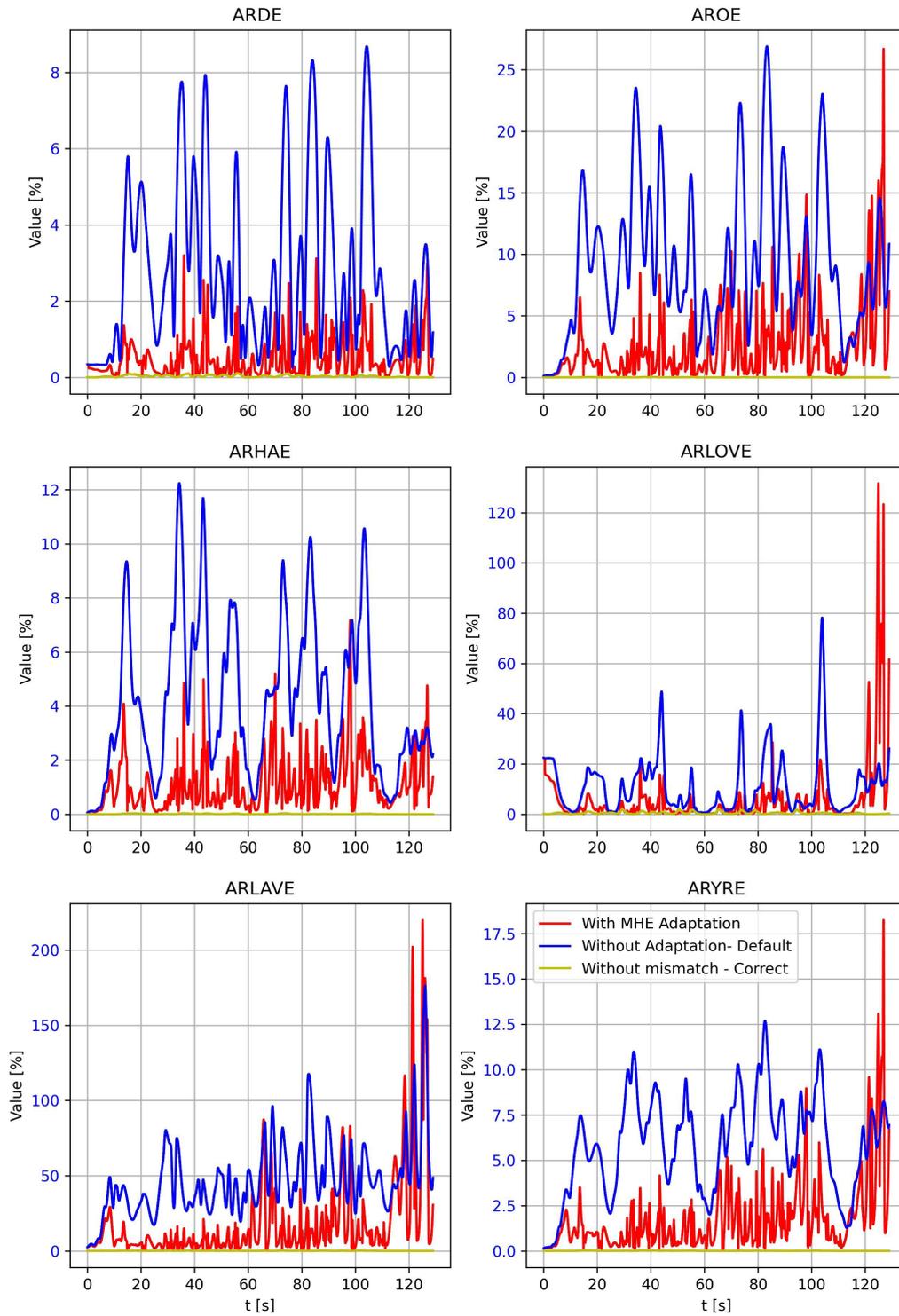


Figure 5.14: ARSE in [%] for Scenario 2/ Red line: Using MHE with selected configuration/ Blue line: With mismatch and No Adaptation (Default) and Yellow line: No mismatch (Correct).

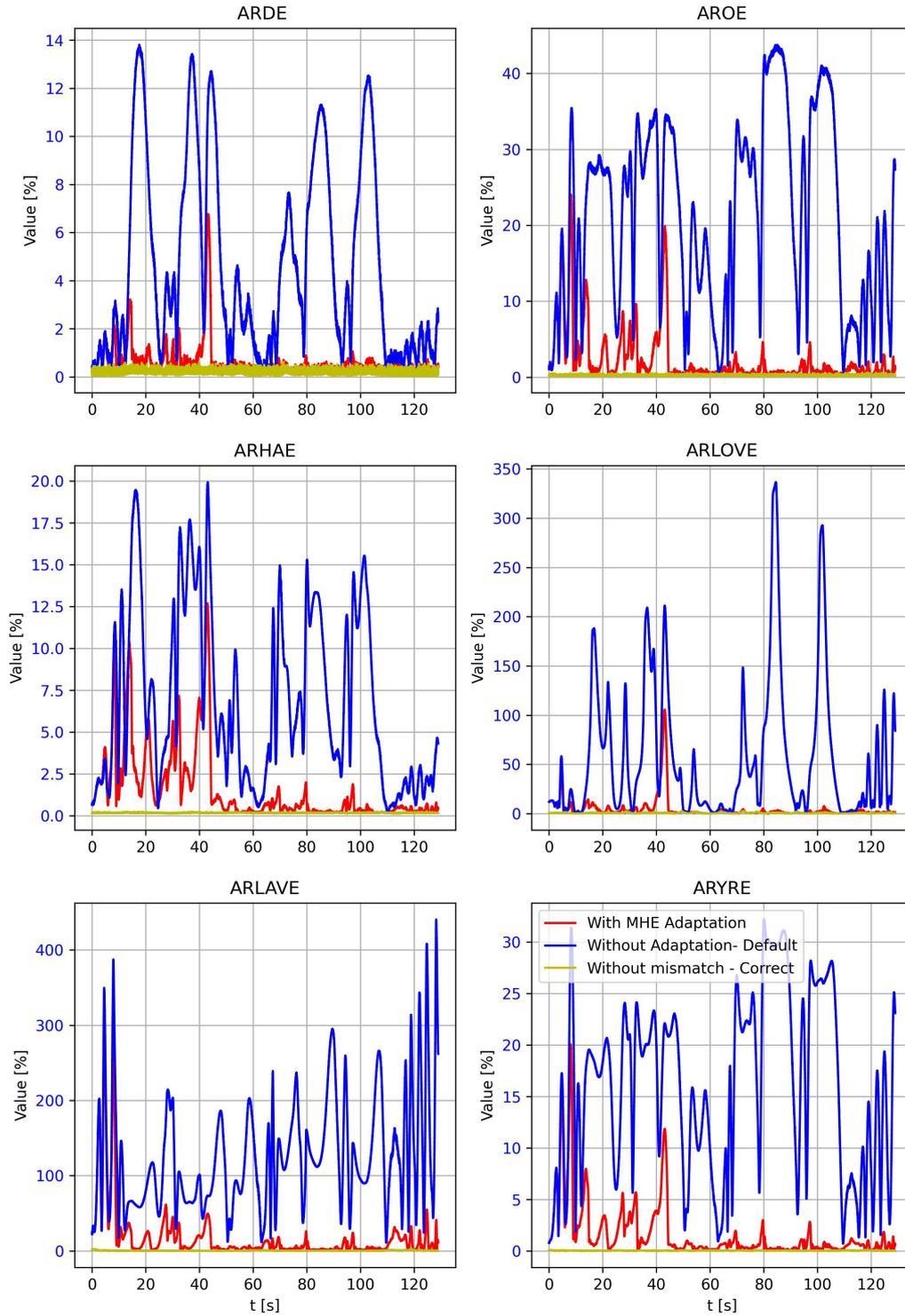


Figure 5.15: ARSE in [%] for Scenario 3/ Red line: Using MHE with selected configuration/ Blue line: With mismatch and No Adaptation (Default) and Yellow line: No mismatch (Correct).

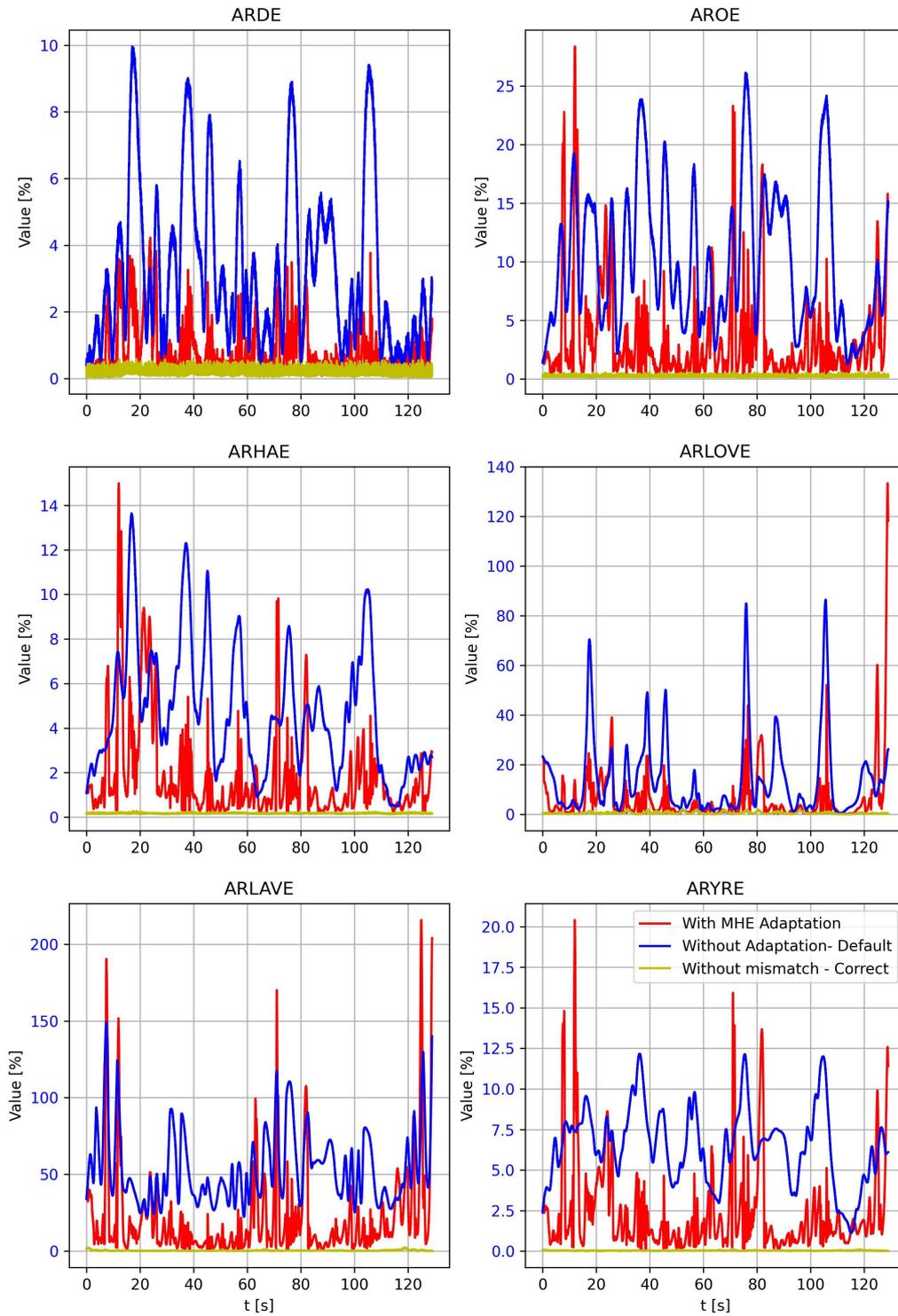


Figure 5.16: ARSE in [%] for Scenario 4/ Red line: Using MHE with selected configuration/ Blue line: With mismatch and No Adaptation (Default) and Yellow line: No mismatch (Correct).

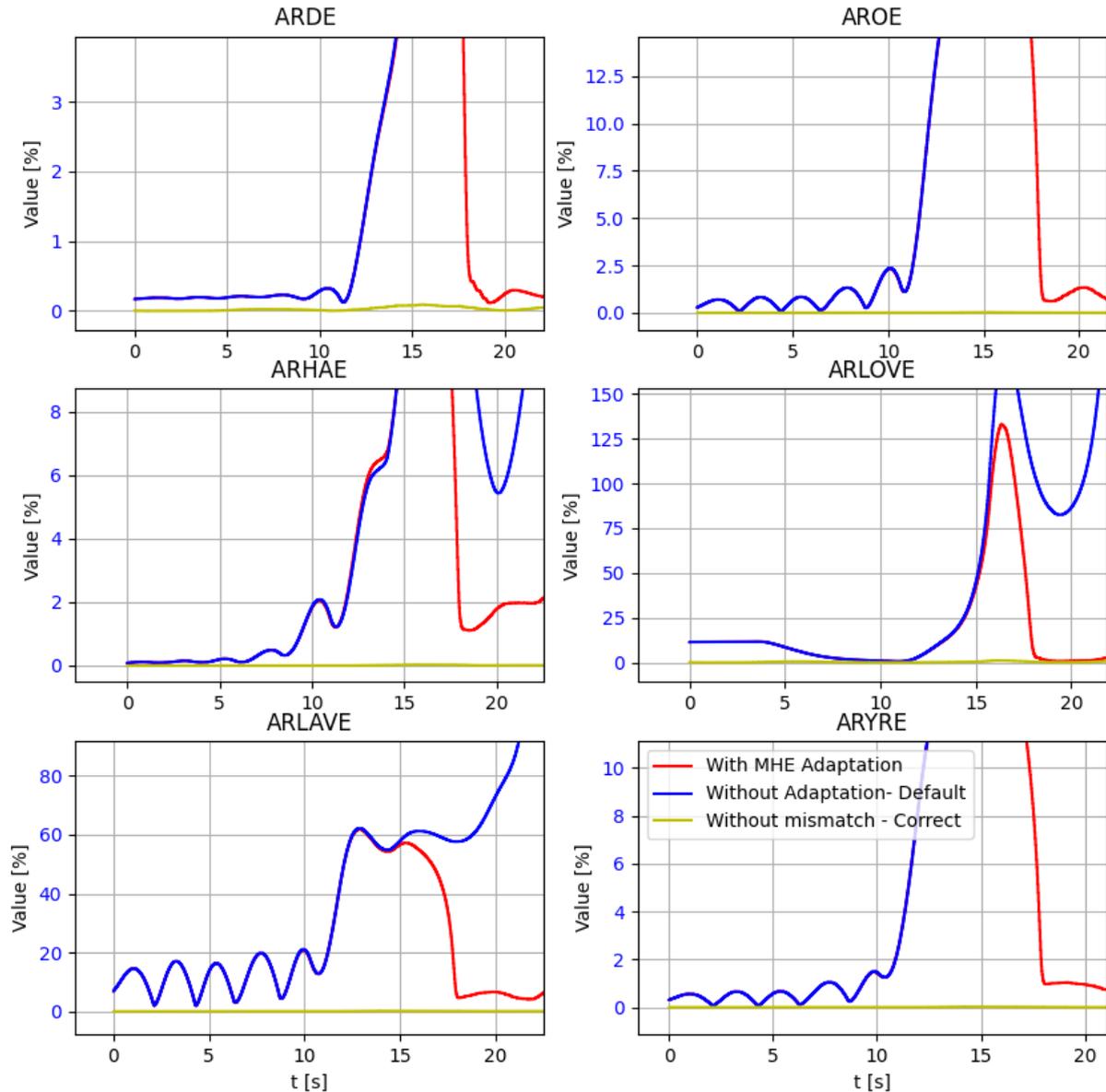


Figure 5.17: ARSE in [%] for Scenario 1/ Red line: Using MHE with selected configuration/ Blue line: With mismatch and No Adaptation (Default) and Yellow line: No mismatch (Correct)- First twenty seconds

A closer analysis of the blue curves reveals that the inclusion of disturbances in the system leads to a noticeable increase in the relative state error values, particularly the ARDE values. This effect is evident when comparing scenario 1 with scenario 3 or scenario 2 with scenario 4. The differences in relative error values across these scenarios are further corroborated by numerical results provided in subsequent sections. These findings highlight the impact of disturbances on the prediction model's accuracy and the challenges they pose to the estimator.

Given that the primary focus of this study is on the path-tracking task, the ARDE results across all scenarios hold particular significance. The selected configuration for the developed estimator demonstrates a substantial improvement in the prediction performance of the model during open-loop simulations. While other relative state errors are also reduced on average, occasional instances of higher relative error values are observed compared to the default simulation of the scenario. These instances may be addressed by further fine-tuning the key parameters of the estimator.

Moreover, it is evident from Figure 5.13 that all relative state errors are significantly reduced in scenario 1, where the estimator was tuned by testing various configurations. This result underscores the critical role of tuning the MHE's key parameters to enhance its effectiveness. Advanced parameter optimization methods could further improve the estimator's performance, ensuring even greater accuracy in prediction tasks.

The analysis thus far strongly suggests that employing the developed estimator to adapt the prediction model offers a significant advantage over using the default parameter set. To quantify this improvement and substantiate the qualitative observations, important statistical indicators—average, median, and maximum—are listed in Table 5.6. However, since their absolute values are difficult to interpret directly, we introduce an improvement metric I , defined as follows:

$$I[\%] = \frac{\epsilon^{\text{Estimator}} - \epsilon^{\text{Default}}}{\epsilon^{\text{Correct}} - \epsilon^{\text{Default}}} \cdot 100 \quad (5.11)$$

where: $-\epsilon^{\text{Estimator}} - \epsilon^{\text{Default}}$ quantifies the improvement gained by using the developed estimator, $-\epsilon^{\text{Correct}} - \epsilon^{\text{Default}}$ represents the maximum achievable improvement by using the correct parameters.

This metric also accommodates values smaller than 0, indicating cases where the estimator performs worse than the default parameter set, as well as values larger than 1, which signify instances where the estimator outperforms even the correct parameters.

As the primary focus of this study is on reducing track-following errors, Table 5.6 exclusively reports the ARDE values and their corresponding improvements. In this table, the "Default Set" refers to simulations conducted using the default parameter set of each scenario without any adaptation, while the "Correct Set" represents simulations executed with the correct parameter set, ensuring no parameter mismatch. The results presented

Table 5.6: Open-loop prediction Assessment - ARDE metric values

Scenario No.	Scenario 1			Scenario 2		
Feature	MHE	Default Set	Correct Set	MHE	Default Set	Correct Set
$\epsilon_{\text{Maximum}} [\%]$	9.45	13.07	0	3.2	8.68	0
$\epsilon_{\text{Median}} [\%]$	0.18	3.76	0	0.35	2.27	0
$\epsilon_{\text{Average}} [\%]$	0.47	4.62	0	0.52	2.83	0
$I_{\text{Maximum}} [\%]$	27.72	0	100	63.12	0	100
$I_{\text{Median}} [\%]$	95.24	0	100	84.72	0	100
$I_{\text{Average}} [\%]$	89.82	0	100	81.62	0	100
Scenario No.	Scenario 3			Scenario 4		
Feature	MHE	Default Set	Correct Set	MHE	Default Set	Correct Set
$\epsilon_{\text{Maximum}} [\%]$	6.76	13.8	0.62	4.22	9.96	0.61
$\epsilon_{\text{Median}} [\%]$	0.31	3.01	0.18	0.49	2.56	0.18
$\epsilon_{\text{Average}} [\%]$	0.52	4.59	0.2	0.78	3.22	0.2
$I_{\text{Maximum}} [\%]$	53.4	0	100	61.36	0	100
$I_{\text{Median}} [\%]$	95.55	0	100	87.01	0	100
$I_{\text{Average}} [\%]$	92.73	0	100	80.32	0	100

in Table 5.6 highlight the significant effectiveness of using adapted prediction models compared to relying on the default parameter set without adaptation. The improvements in median, average, and maximum ARDE values range from 84% to 95%, 80% to 93%, and 27% to 63%, respectively, across different scenarios. These results demonstrate the robustness of the proposed method in enhancing prediction accuracy under

varying conditions. It is crucial to recognize that these enhancements are strongly influenced by the tuned parameters of the MHE. Further optimization of the MHE parameters could potentially yield even greater performance improvements.

As discussed earlier, the presence of disturbances in the system when using the default parameter set degrades the open-loop prediction quality. This degradation is evident in Table 5.6 when comparing the $\epsilon_{\text{Maximum}}$ values in the "Default Set" column between Scenario 1 and Scenario 3 or Scenario 2 and Scenario 4. Specifically, the maximum prediction error increases from 13.07% to 13.8% and from 8.68% to 9.98%, respectively. This increment has happened also for ϵ_{Median} and $\epsilon_{\text{Average}}$. These increases illustrate the detrimental impact of disturbances on prediction performance when no adaptation mechanism is employed.

However, when utilizing the developed adaptation technique, the impact of disturbances does not necessarily lead to further performance degradation. This is due to the fact that, in the presence of disturbances, the estimator starts changing the mutable parameters inside the prediction model sooner, which might help to achieve better overall performance. The behavior of the adapted prediction model in the presence of disturbances depends heavily on the parameter tuning of the MHE. Proper tuning allows the estimator to mitigate the effects of disturbances effectively, thereby maintaining or even improving the prediction performance under such conditions.

Another noteworthy observation is that disturbances introduce relative errors even when the "Correct Set" is used, where no parameter mismatch exists between the controller and the simulator. This phenomenon occurs because disturbances alter the state values between the simulator (plant) and the controller's prediction model, leading to discrepancies even in the absence of parameter mismatches. These findings highlight the inherent challenge of maintaining prediction accuracy in the presence of external disturbances, further emphasizing the importance of the adaptation mechanism in improving system robustness.

Overall, the analysis demonstrates that the developed adaptation technique significantly enhances the prediction accuracy of the model under various conditions, including scenarios with and without disturbances. These results validate the efficacy of the proposed method and highlight areas for further improvement, particularly in parameter tuning and disturbance mitigation.

Figures 5.18 and 5.19 illustrate the open-loop estimated mutable parameters, namely the mass and inertia of the trailer section, across all four scenarios. As previously stated, the primary goal of employing the estimator is to provide the controller, NMPC, with updated parameters to achieve improved prediction quality. Consequently, the estimator does not necessarily need to precisely estimate the real mutable parameters. Instead, any combination of estimated parameters that minimizes the cost function within the estimator can effectively serve the controller. However, it is desirable for the estimator to produce parameter estimations that are generally close to the real values of the mutable parameters and capable of tracking their trends, which is evident from the results shown in these figures.

A closer inspection of the figures reveals that in the presence of disturbances, particularly in Scenarios 3 and 4, the estimation plots, especially those for mass estimation, exhibit more oscillatory behavior and contain more spikes. This behavior can be attributed to the added complexity introduced by disturbances, which likely increases the deviation between the simulated and predicted states and makes the optimization problem within the estimator more challenging.

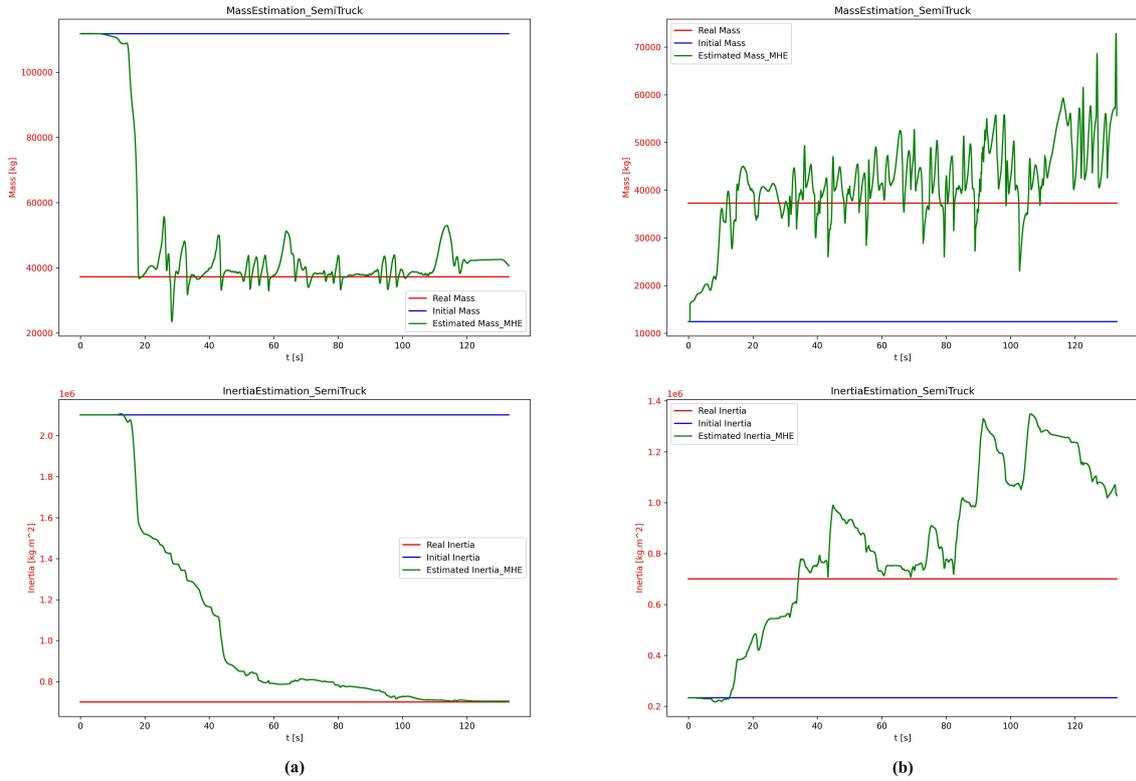


Figure 5.18: Open-loop mutable parameter estimation plots in the absence of disturbance — Scenario 1 (a) and Scenario 2 (b)

Performance Analysis under Reduced State Information

In the subsequent analysis, the prediction quality and stability of the estimators under the presence of incomplete information are investigated. Specifically, this study focuses on withholding the lateral velocity, v_{lat} , as it is generally difficult to measure accurately, and its measurements are often significantly noisy.

To account for the missing v_{lat} information, the estimator's configuration for weighting matrices needs to be modified. The matrices Q and Q_{mhe0} are adjusted as follows:

$$Q = 2 \times \text{Diag}[10^4, 10^4, 10^4, 10^4, 10^4, 0.0, 10^4, 10^4, 10^4, 10^4] \quad (5.12)$$

$$Q_{mhe0} = \text{Diag}[10^4, 10^4, 10^4, 10^4, 10^4, 10^{-4}, 10^4, 10^4, 10^4, 10^4, 10^{-5}, 10^{-7}] \quad (5.13)$$

In the modified configuration, the state weight for the lateral velocity, w_{lat} , is set to zero in Q to exclude this state from the cost function of the estimator. This ensures that the absence of v_{lat} measurements does not directly affect the optimization process. However, the corresponding weight factor for the lateral velocity in Q_{mhe0} is set to 10^{-4} . This adjustment encourages the estimator to utilize its own previous estimation for v_{lat} , thereby reducing the likelihood of excessive oscillations in the lateral velocity estimation. This approach strikes a balance by not relying on unavailable measurements while maintaining stability in the estimation process.

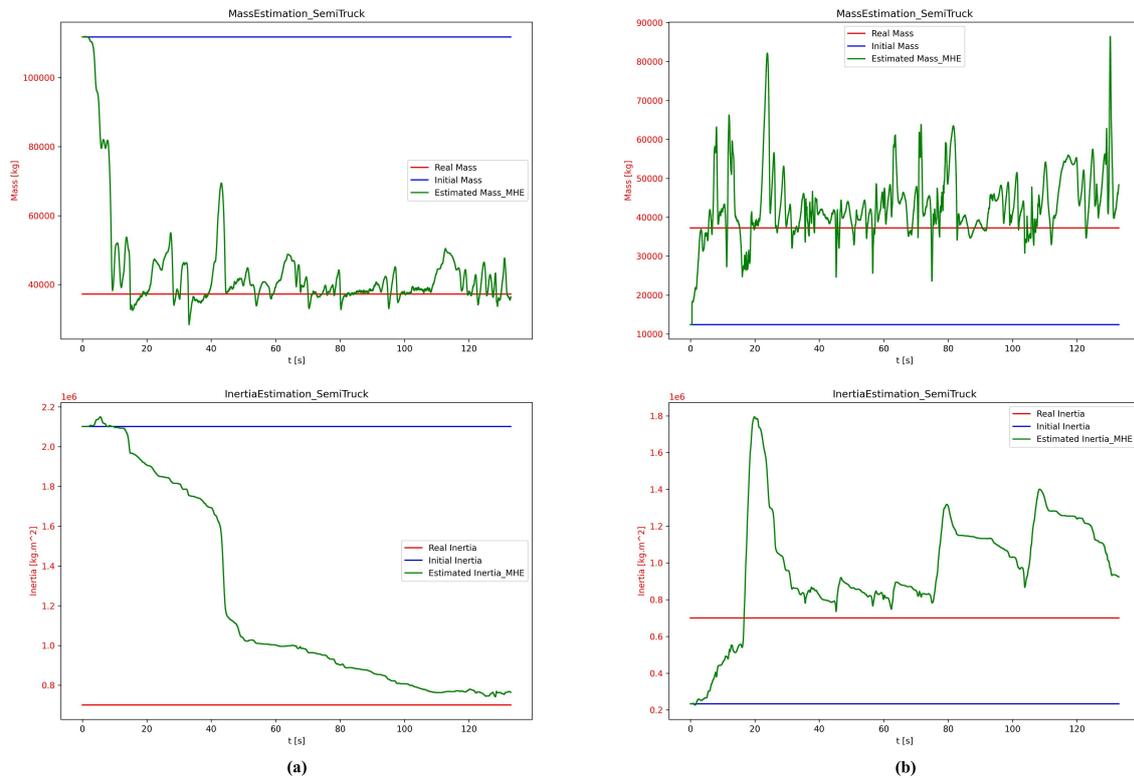


Figure 5.19: Open-loop mutable parameter estimation plots in the presence of disturbance — Scenario 3 (a) and Scenario 4 (b)

As stated earlier, the MHE method is capable of estimating not only the parameters but also the states. In the following figures (Figures 5.20 and 5.21), the estimated lateral velocity for all scenarios in two different situations—when the real lateral velocity values are provided and when this information is missing—is shown against the real lateral velocity over the simulation time.

In each figure:

- The red dotted diagram represents the estimated lateral velocity by the MHE when full knowledge of the real system states is provided. It is evident that the estimated values closely align with the real lateral velocities (shown by the green-colored diagram), demonstrating the effectiveness of the estimator when all information is available.
- The blue lines represent the estimated lateral velocity values in the absence of real information. These blue plots exhibit deviations from the real values and display some unstable behavior at certain points during the simulation. This highlights that the estimator's stability decreases when information is missing, and further fine-tuning of the weighting matrices or the estimation horizon length might be necessary to enhance stability. Nonetheless, even without the real lateral velocity data, the estimator manages to capture the general trend of the real values.

Additionally, by comparing Figures 5.20 and 5.21:

- In Scenarios 3 and 4, where disturbances are present, the estimations exhibit increased oscillations and greater instability compared to Scenarios 1 and 2 (without disturbances).

This suggests that the combination of missing information and the presence of disturbances exacerbates the risk of estimator infeasibility.

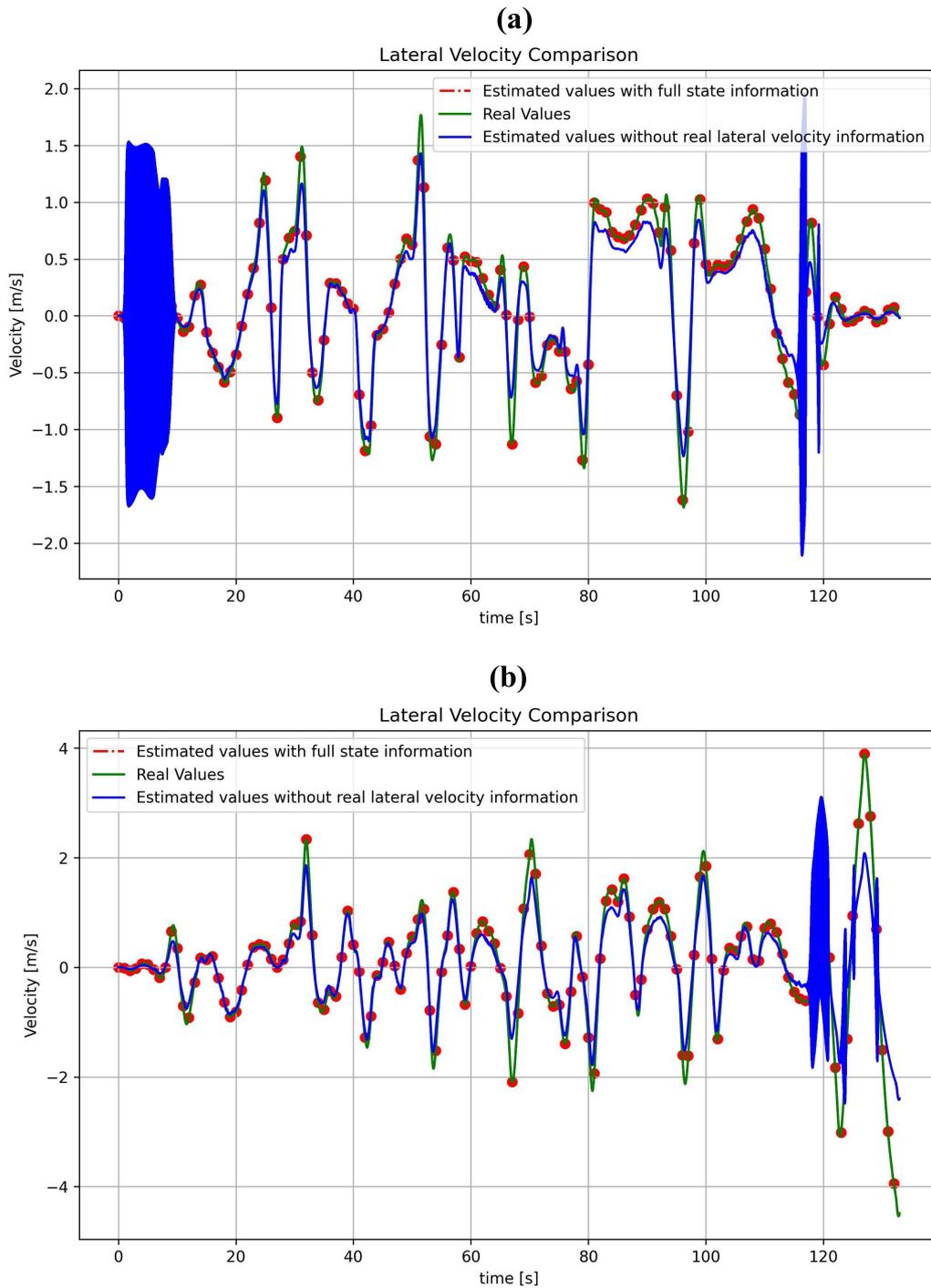


Figure 5.20: Lateral velocity estimations against simulation time in the absence of disturbance - Scenario 1 (a) and Scenario 2 (b)

The results emphasize the importance of providing complete information and ensuring disturbance robustness in the design and tuning of the estimator.

By using the procedure and metric introduced in the previous section, the open-loop prediction performance of the estimator, considering the presence and absence of the real values of lateral velocity (v_{lat}) data in all 60

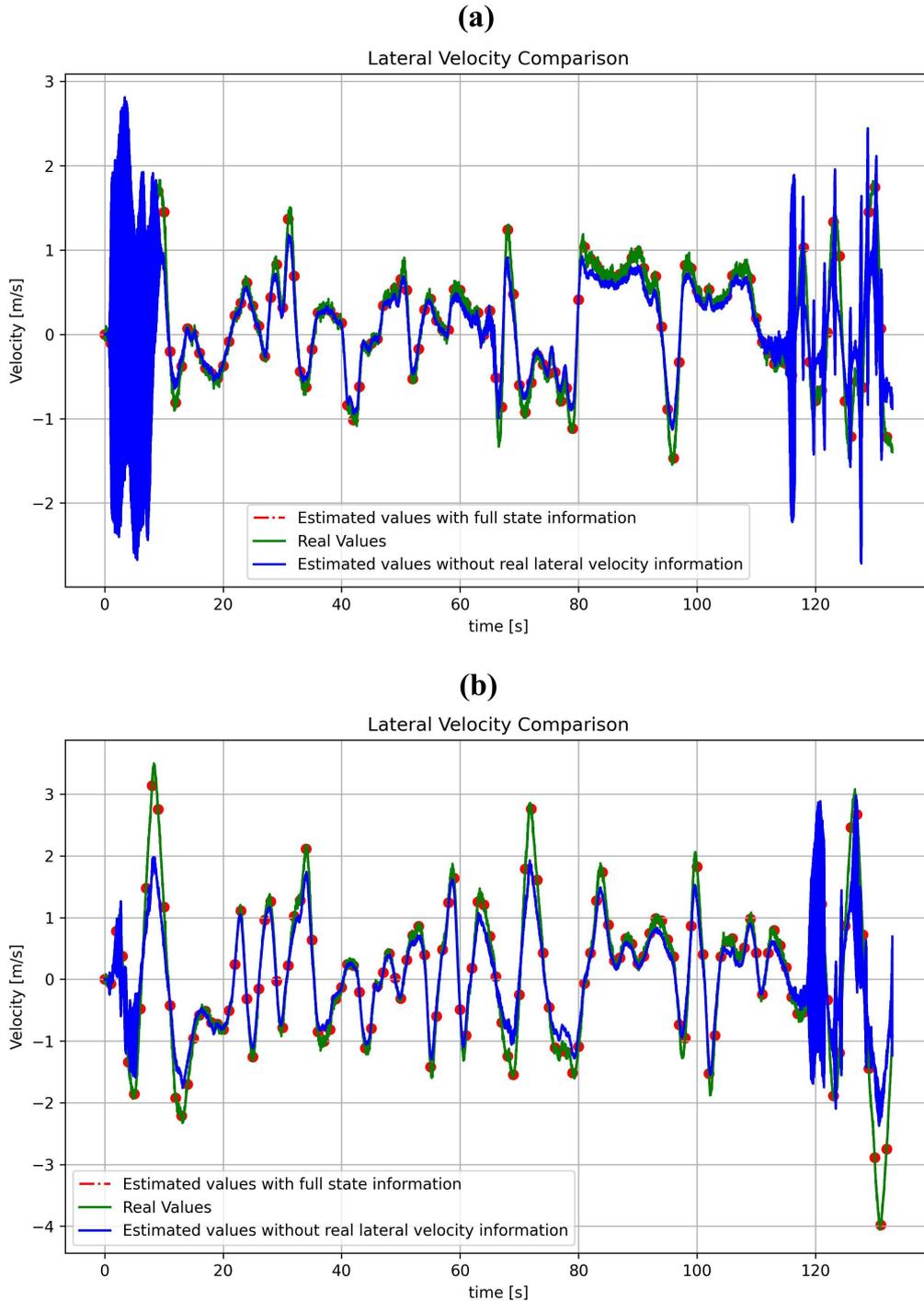


Figure 5.21: Lateral velocity estimations against simulation time in the presence of disturbance - Scenario 3 (a) and Scenario 4 (b)

scenarios, is evaluated. The respective performance improvements (I) for the ARDE metric are depicted in Figure 5.22.

Analyzing the column bars shown in Figure 5.22 reveals that the performance of the adapted prediction model deteriorates when the data for v_{lat} is unavailable. This decline in performance is consistent across all scenarios and statistical indicators, including the average, median, and maximum values. The reason for this behavior lies in the fact that, in the absence of real v_{lat} data, lateral velocity essentially becomes another mutable parameter for the estimator. However, unlike the primary mutable parameters, v_{lat} is

governed by its own dynamic equation, as described by the dynamic model. Consequently, the estimator must simultaneously estimate the two mutable parameters (without any real-time data available for them) and the lateral velocity, significantly increasing the complexity of the optimization problem. This added challenge reduces the estimation performance.

Moreover, the degradation in performance is more severe for the maximum ARDE values compared to the average or median values. This indicates that missing v_{lat} data particularly impacts the model's ability to handle outlier points or extreme deviations effectively.

Despite these challenges, the improvements in other statistical indicators, including the median and average values, remain close to 80% even without real v_{lat} data. This demonstrates the robustness and effectiveness of the estimation method, even in the presence of missing information.

As expected, the absence of lateral velocity (v_{lat}) information also impacts the estimation of mutable parameters. By examining Figure 5.23, which illustrates the open-loop estimated mutable parameters for Scenario 1, it becomes evident that the lack of real v_{lat} data results in more oscillatory and unstable behavior in the predicted parameters, particularly for the mass estimation. This observation aligns with previous conclusions regarding the detrimental effects of missing information on the estimator's performance.

The observed oscillatory behavior and instability emphasize the increased difficulty faced by the estimator in minimizing the cost function when the full set of state information is unavailable.

Nonetheless, even under these conditions, the estimator is still able to follow the general trend of the real mutable parameter values, demonstrating its robustness to some extent.

5.5.2 Closed-loop tests

In this section, we analyze the path-tracking performance of the pre-developed NMPC for the semi-trucks equipped with the developed estimator using the MHE method. The evaluation is conducted under the scenarios described in Section 5.4 (Table 5.3) on the Monteblando racetrack.

A significant difference between these tests and the open-loop tests is that the estimator continuously and in real-time updates the mutable parameters in the prediction model used by the NMPC. This dynamic adjustment directly influences the control commands issued by the NMPC, leading to potential improvements in path-tracking accuracy.

Furthermore, we also measure the additional time introduced by integrating the estimator sequence into the main simulation procedure to assess its impact on real-time feasibility. This evaluation provides insights into the practical applicability of the developed estimator in real-time control scenarios.

Performance Evaluation

The closed-loop analysis primarily focuses on evaluating the lateral deviation from the reference trajectory, as this is where the impacts of parameter mismatches or environmental changes are most apparent. Moreover, lateral deviation is inherently the most safety-critical error in path tracking, making it a crucial performance metric.

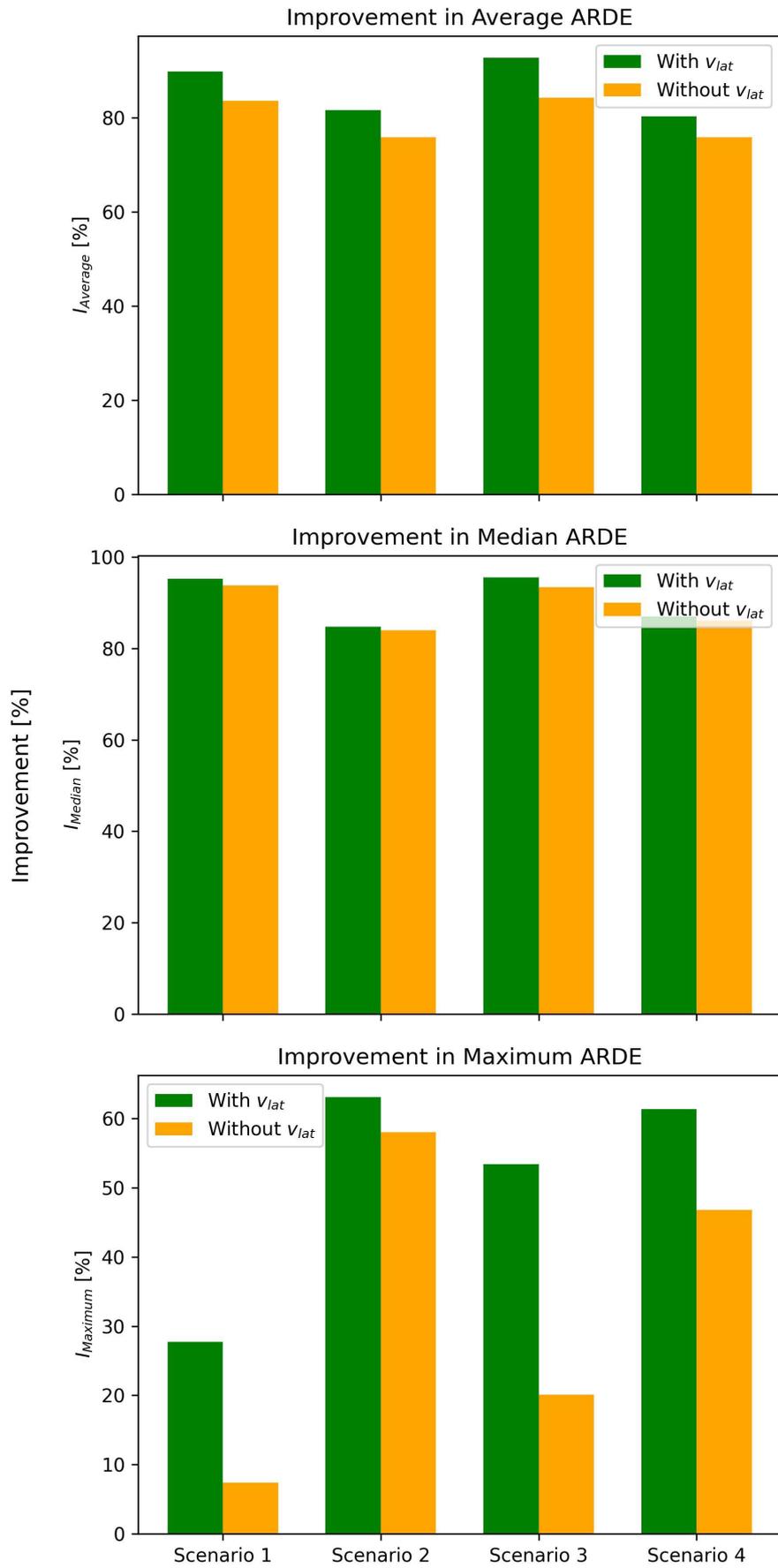


Figure 5.22: Improvements in ARDE for the estimator aware and unaware of v_{lat} .

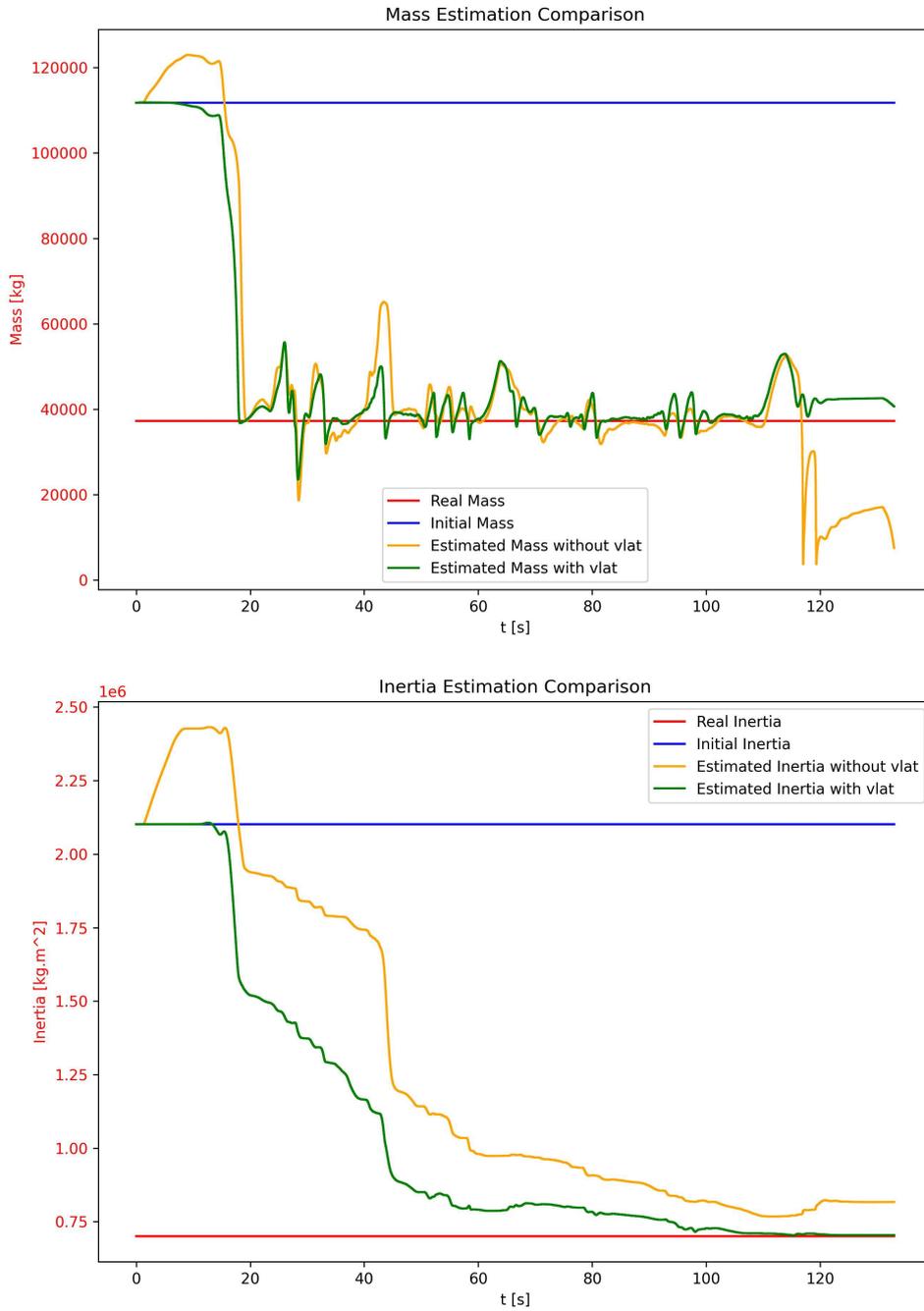


Figure 5.23: Open-loop estimated mutable parameters for scenario 1 with and without v_{lat} .

Figure 5.24 shows the resulting lateral deviation from the reference trajectory line for all scenarios. By examining these plots closely, it can be observed that at the start of each scenario's simulation, the lateral deviation diagrams for the adapted NMPC (green curves) initially overlap with the blue-colored diagrams

(no adaptation employed). This occurs because the estimator is initialized with the parameters described in Table 5.3.

However, after a brief period—particularly when the semi-truck approaches the first turn—the estimator begins adapting the NMPC’s mutable parameters. As a result, the lateral deviation decreases significantly and approaches the red-colored diagrams, which represent the lateral deviation results with no parameter mismatch between the controller and the plant. This demonstrates that the developed estimator effectively enables the NMPC to handle parameter mismatch, regardless of the scenario, over the majority of the simulation duration.

The enhancement provided by the estimator is further emphasized in Figure 5.25, which depicts box plot results of the absolute lateral deviation from the reference trajectory for all scenarios. The box plots clearly demonstrate that incorporating the estimator significantly reduces the maximum lateral deviations. Additionally, the majority of lateral deviation values are markedly decreased, showcasing the estimator’s effectiveness in enhancing the path-tracking performance of the NMPC.

Quantitatively, in the first scenario, the maximum absolute lateral deviation without adaptation exceeds 7 meters. However, with the NMPC utilizing the developed estimator, the deviation is reduced to less than 2 meters. Similarly, substantial reductions are observed in the other scenarios: in scenario 2, the deviation decreases from 12 meters to less than 2 meters; in scenario 3, it drops from over 5 meters to approximately 1.8 meters; and in scenario 4, it is reduced from more than 25 meters to nearly 2 meters. These results highlight the robustness of the estimator in minimizing lateral deviations across diverse parameter mismatch conditions.

To have a better understanding of the results, we can further utilize the statistical indicators displayed in Figure 5.25 to quantify the improvement in path-tracking performance, similar to the approach used in the open-loop analysis. The path-tracking improvement indicator, $I_{\text{path-tracking}}$, is defined as follows:

$$I_{\text{path-tracking}}[\%] = \frac{\Delta_{\text{Estimator}} - \Delta_{\text{Default}}}{\Delta_{\text{Correct}} - \Delta_{\text{Default}}} \cdot 100 \quad (5.14)$$

where Δ represents the respective statistical indicator of the absolute lateral deviation (e.g., average, median, or maximum). This metric allows us to evaluate the effectiveness of the developed estimator in improving path-tracking performance by comparing it to the default parameter set and the correct case with no parameter mismatch.

Using the previously defined equation and the data from Figure 5.25, Figure 5.26 is derived. This figure demonstrates that the estimator reduces the average lateral deviation by at least 99%, the maximum lateral deviation by at least 95%, and the median by at least 103%. Improvement values exceeding 100% suggest that the NMPC equipped with the estimator can outperform the standard NMPC. This enhancement is particularly pronounced in scenarios 3 and 4, where light disturbances are present.

Two possible reasons may explain this observation. First, as discussed earlier, the MHE method solves a nonlinear optimization problem. This process can result in an estimated parameter set that enables the controller to issue more accurate control commands, allowing the vehicle to follow the desired trajectory slightly better than in the ideal case without parameter mismatch. Second, while disturbances increase instability and oscillatory behavior in the estimator, they also affect the adaptation process timing. In scenarios without disturbances, the simulations start with a few seconds of straight roads where deviation errors are minimal, indicating that the controller is already performing well. Consequently, the estimator does not perceive an immediate need to adjust the mutable parameters, delaying adaptation until the semi-truck encounters turns or scenarios with rising deviation errors. Conversely, in scenarios with disturbances, errors are present from the beginning of the simulation. This prompts the estimator to adjust the mutable parameters

earlier, which explains why the path-tracking improvement indicators in scenarios 3 and 4 are higher than their corresponding values in scenarios 1 and 2.

However, it is important to note that disturbances and missing state information can introduce instability in the estimator. As a result, the estimator does not guarantee improved controller performance under all disturbance conditions.

By reviewing Figures 5.24 to 5.26, it is notable that the lateral deviations influenced by parameter estimation closely align with the deviations observed under no mismatch conditions. This occurs because the no-mismatch condition acts as a near-ideal case where the controller and the plant have closely aligned dynamic models, resulting in minimal deviation. The primary goal of the adaptation technique is to enhance the prediction quality of the controller. Consequently, the estimator adapts the NMPC to achieve performance as close as possible to the no-mismatch condition, sometimes resulting in improvements beyond expectations.

Figures 5.27 and 5.28 illustrate the estimated mutable parameters across all closed-loop simulation scenarios. As previously discussed, the estimator is not required to estimate the exact values of all mutable parameters at each time step. However, since the controller achieves near-ideal performance when the estimated parameters are close to the real values (i.e., no mismatch condition), it is expected that the estimator follows the general trend of the real mutable parameters.

This behavior is clearly evident in these figures, where the estimator adjusts the estimated parameters (represented by the green curves) to gradually approach their real values (shown by the red lines) over time, starting from each scenario's default mutable parameter set (depicted by the blue line). Additionally, by comparing Figure 5.27 with Figure 5.28, it is evident that scenarios 3 and 4 display more oscillatory patterns with higher peaks and dips. This behavior is a direct result of the disturbances present in these scenarios, which further emphasize that disturbances contribute to the increased instability of the estimator.

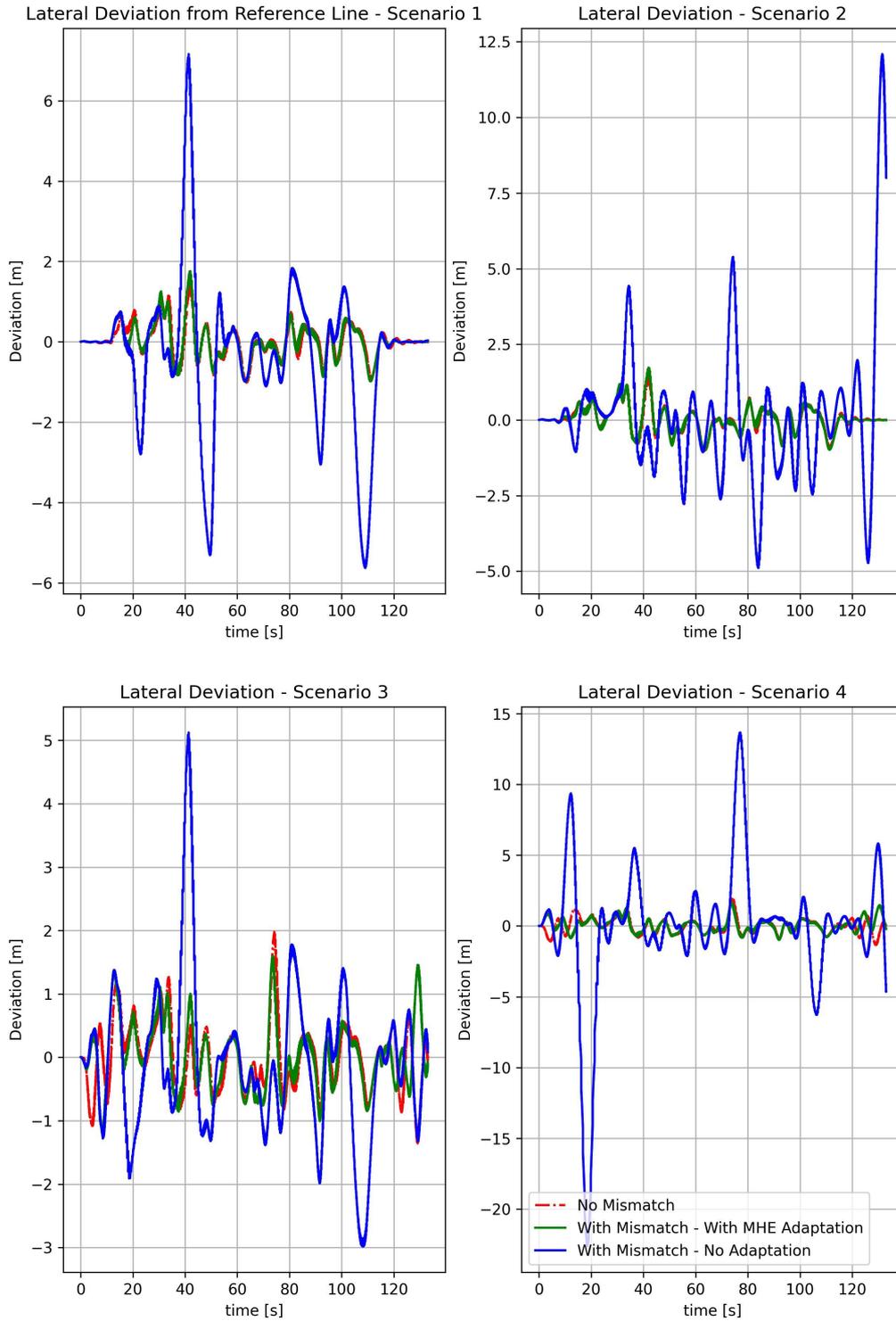


Figure 5.24: Lateral deviation from reference trajectory line in all scenarios.

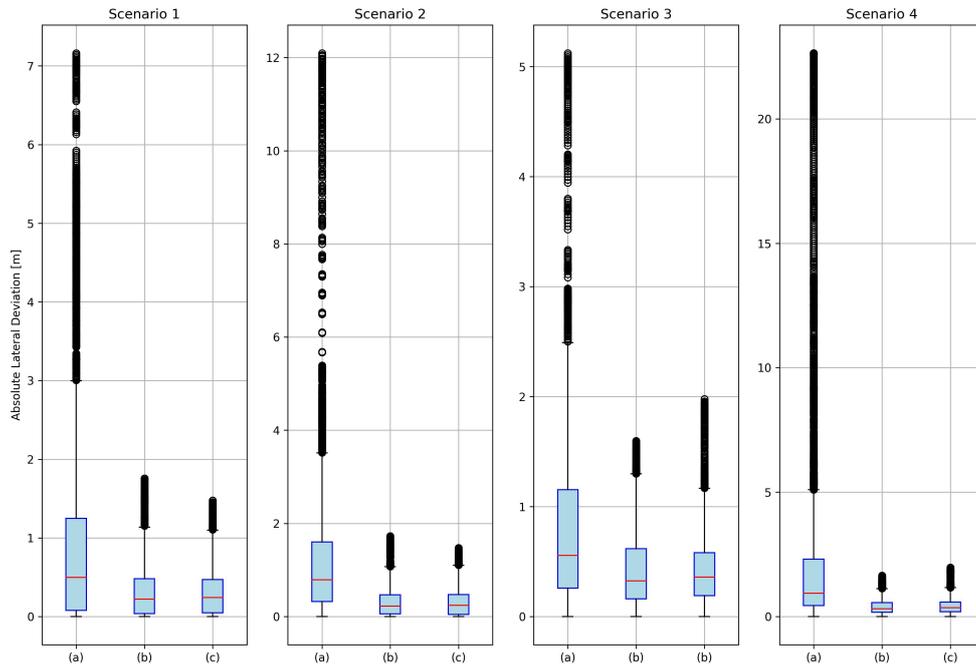


Figure 5.25: Box plots of absolute lateral deviation from reference trajectory line in all scenarios. (a)With mismatch - No adaptation, (b)With mismatch - With Adaptation and (c)No mismatch.

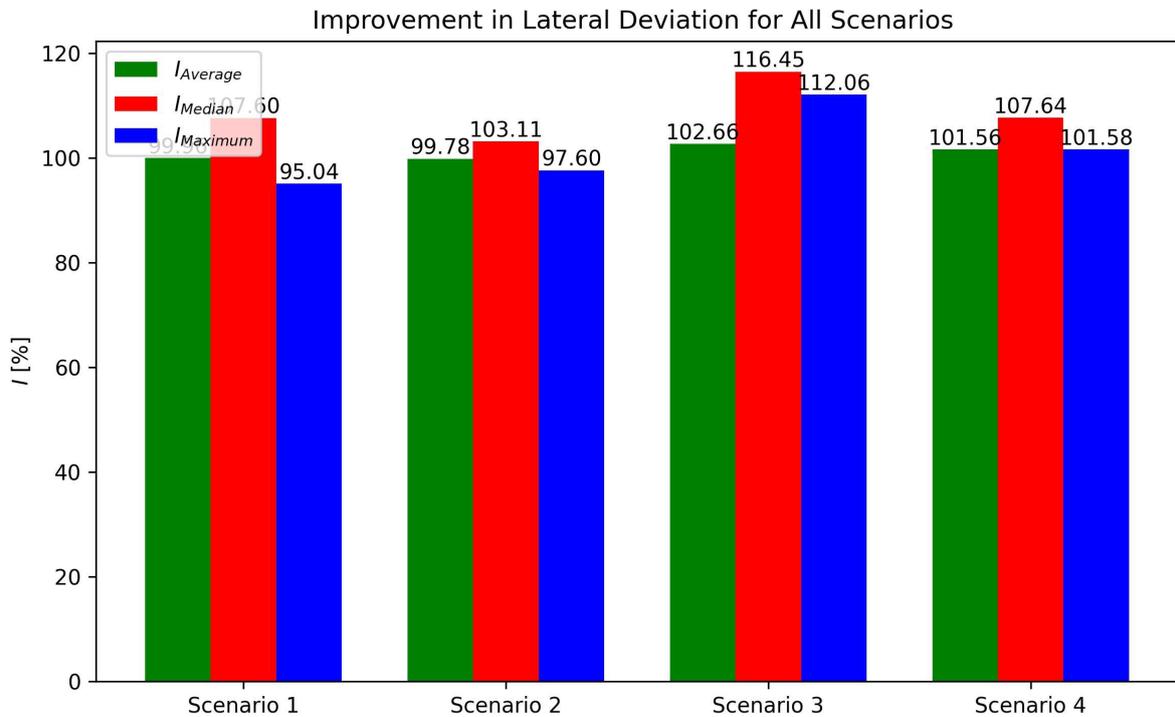


Figure 5.26: Improvement in absolute lateral deviation from the reference trajectory line in all scenarios.

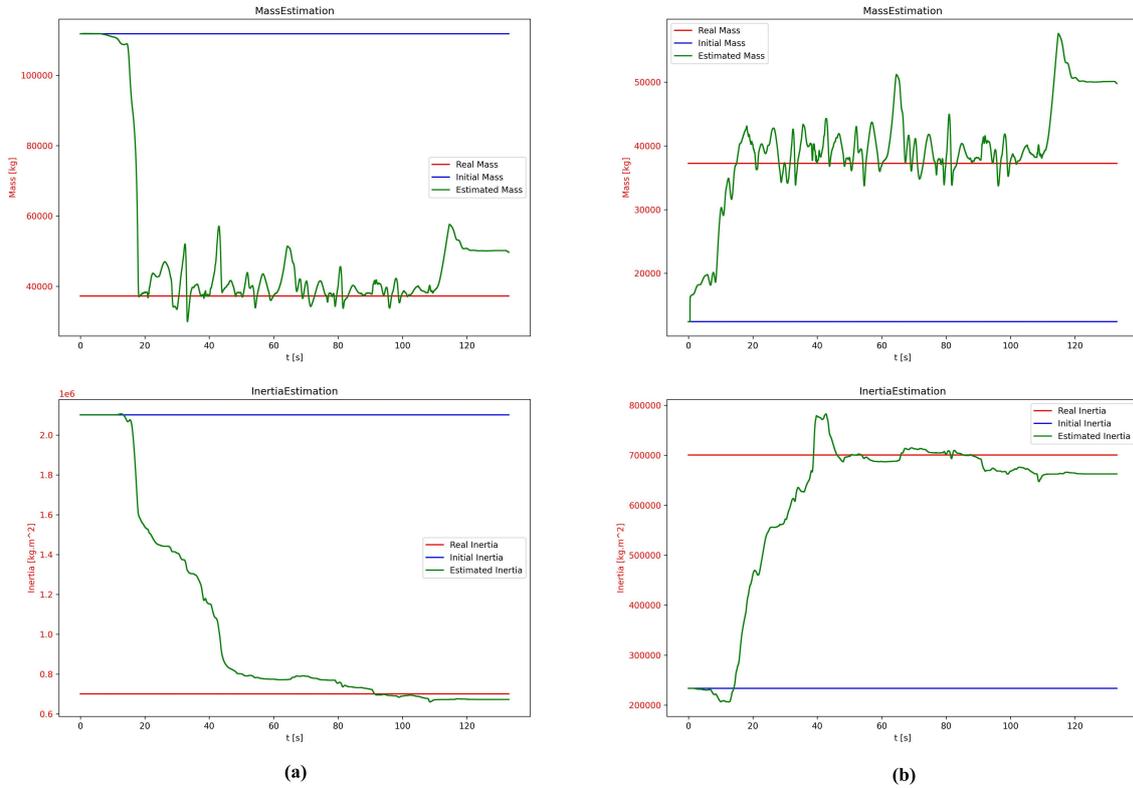


Figure 5.27: Closed-loop mutable parameter estimation plots in the absence of disturbance / Scenario 1 (a) and Scenario 2 (b) / Blue : Default parameters, Red : Correct parameters and Green : Estimated parameters

Simulation Time

This subsection reviews the simulation times across all scenarios, both with and without estimator utilization. Specifically, the simulation time of closed-loop tests is analyzed to evaluate the additional computational burden introduced by integrating the estimator sequence into the main simulation procedure and assess its impact on real-time feasibility. The experiments yielding the simulation times in this subsection were conducted on an Intel i7-6500U 3.10GHz CPU with 8 GB DDR4 RAM.

The total time required for each simulation scenario, along with the average time per iteration (expressed in seconds), is presented in Table 5.7. A detailed analysis of the table reveals the following observations:

1. Mismatch Impact on Simulation Time: In all scenarios, transitioning from the Correct Set (no mismatch) to the Default Set (with mismatch but no adaptation) increases simulation times. This is because, under mismatched conditions, the optimization problem within the NMPC solver becomes more complex, requiring additional time to converge to a solution that satisfies the dynamic equations of the prediction model and all constraints. For example, in Scenario 3, the total simulation time increases by approximately 9%, from 631.5 seconds to 688.1 seconds.

2. Impact of Disturbances Without Adaptation: By comparing Scenario 1 with Scenario 3 or Scenario 2 with Scenario 4, it is evident that the presence of disturbances increases simulation times when no adaptation technique is applied. For instance, moving from Scenario 1 to Scenario 3 (using the Default Set), the total simulation time increases by about 10%. This is due to the added complexity disturbances introduce into the optimization problem, which makes convergence more time-consuming.

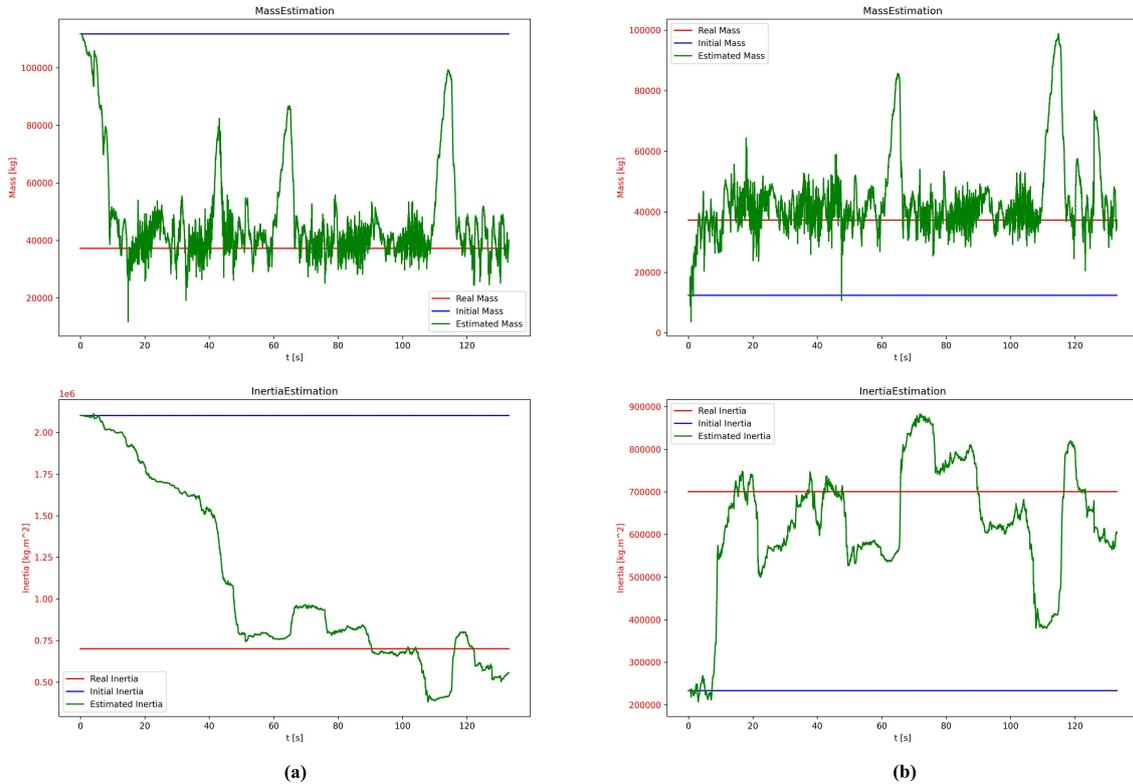


Figure 5.28: Closed-loop mutable parameter estimation plots in the presence of disturbance / Scenario 3 (a) and Scenario 4 (b)/ Blue : Default parameters, Red : Correct parameters and Green : Estimated parameters

3. Impact of Estimator Utilization: Across all scenarios, incorporating the estimator significantly increases simulation times. For instance, in Scenario 1, the total simulation time increases from 622.4 seconds to 1154.4 seconds, an increase of over 84%. This increase is expected, as the estimator adds its computational burden to the simulation. These computations include creating and updating buffer data, preparing data inside the buffer, constructing the cost function, and solving the optimization problem—the latter being the most time-intensive task. Addressing this challenge to maintain real-time feasibility requires trade-offs, such as:

- Reducing the estimation horizon, though this may increase sensitivity to disturbances.
- Simplifying the cost function by reducing its terms, albeit at the expense of accuracy.
- Using more powerful hardware with faster processors and larger memory, which increases the cost.

4. Estimator Performance with Disturbances: An interesting observation emerges when comparing the total simulation times for scenarios with adaptation in the absence or presence of disturbances. Moving from Scenario 1 to Scenario 3 or from Scenario 2 to Scenario 4, the total simulation time decreases. Although disturbances generally make the optimization problem more time-consuming to solve, the adapted parameters provided by the estimator (which are closer to the Correct Set) help the solver converge faster. For example, in Scenario 3 (with the estimator), the total time is reduced to 1136.3 seconds compared to 1154.4 seconds in Scenario 1. This reduction suggests that the benefit of using adapted parameters outweighs the time

increase caused by disturbances in this study. However, this behavior may not hold under heavier disturbance conditions.

Table 5.7: Closed-loop simulation times

Scenario No.	Scenario 1			Scenario 2		
Feature	MHE	Default Set	Correct Set	MHE	Default Set	Correct Set
Total simulation time [s]	1154.4	622.4	604.6	1148.9	644.9	604.6
Average time per iteration [s]	0.174	0.094	0.091	0.173	0.097	0.091
Scenario No.	Scenario 3			Scenario 4		
Feature	MHE	Default Set	Correct Set	MHE	Default Set	Correct Set
Total simulation time [s]	1136.3	688.1	631.5	1112.0	679.2	631.5
Average time per iteration [s]	0.171	0.103	0.095	0.168	0.102	0.095

6 Conclusion and Outlook

This work investigates the online adaptation of an NMPC prediction model, developed for a semi-truck, to address the parameter mismatch challenge between the controller and the plant. The objective of this adaptability is to enhance the controller's ability to issue better control commands by improving the prediction quality.

The proposed approach employs a nonlinear dynamic single-track model, presented in state-space formulation, as the basis for both prediction and estimation models. This study begins by validating the chosen dynamic model and subsequently develops an NMPC with appropriate constraints for motion tracking purposes. The primary focus, however, is on developing the adaptation process, which leverages the concept of MHE to make the proposed method adaptable to dynamically changing environmental conditions while remaining feasible for real-time application.

Furthermore, the presented adaptation method is not limited to semi-trucks or the mutable parameters considered in this study—namely, the mass and inertia of the trailer section. It can also be applied to other vehicles and parameter mismatches, provided certain conditions are met. These include the availability of a sufficiently accurate dynamic model to serve as the prediction and estimation model and the ability to directly measure or estimate the full state of the system with adequate accuracy. When these conditions are fulfilled, implementing an efficient model adaptation process requires only the identification of a suitable set of configuration parameters in the estimator.

The performance and impact of the model adaptation are primarily assessed using the simulation environment TUM-Control [53]. The entire implementation for this study is carried out in Python. Model formulation, as well as the development of cost functions and the solving of optimization problems, are achieved using CasADi [54], ACADOS tools [55], and NumPy [56].

The prediction performance of the adapted models and the path-tracking performance of an NMPC controller equipped with the adaptive prediction model are thoroughly analyzed under four different scenarios. These scenarios include parameter mismatch conditions and the presence of light disturbances in the system, combined with a highly dynamic trajectory. The resulting performance is benchmarked against (i) the non-adaptive use of the parameter set, which serves as the initial guess for model adaptation, and (ii) the use of a parameter set that is instantly adapted to the one currently employed in the simulation, representing a near-ideal situation with no mismatch. Additionally, the study explores the effects of missing state information, specifically lateral velocity, on the performance of the adaptation process and the controller.

The proposed nonlinear dynamic single-track model for semi-trucks is validated using the scenario defined in [29]. Despite a maximum deviation of 8% in the hitch angle results between the developed model and the reference values—attributable to differences in the lateral tire force model and potential errors in the reference data acquisition process—the developed prediction model performs well within the defined scenario. The validated prediction model, with appropriately defined constraints on state and control input variables, is further utilized to design an NMPC strategy for semi-trucks. This standard NMPC is tested for a path-tracking mission on the Montebianco track, where it successfully controls the vehicle along the track while satisfying all constraints, provided there is no parameter mismatch.

The proposed estimator is tested across four distinct scenarios with parameter mismatches and achieves an average improvement of 85% in open-loop prediction performance in the absence of disturbances and 86% in the presence of light disturbances, when compared to the non-adaptive parameter set. Path-tracking performance, in terms of lateral deviation, is similarly enhanced, with improvements averaging no less than 99% across all scenarios, regardless of the presence or absence of light disturbances. Although the estimator exhibits minor performance reductions and increased instability when lateral velocity information is unavailable, it continues to operate successfully even without access to these measurements. Moreover, the NMPC equipped with the adaptive model demonstrates behavior comparable to, and in some cases even superior to, the NMPC that is fully aware of the simulation parameters, highlighting the robustness of the model adaptation.

However, employing the NMPC equipped with the proposed estimator increases the total simulation time by up to 85%, which poses a challenge to real-time feasibility. To address this, several potential solutions are suggested, including reducing the estimation horizon, simplifying the cost function, and utilizing more powerful hardware with faster processors and larger memory. Nevertheless, further research is required to explore strategies for reducing simulation time and ensuring real-time feasibility in practical applications.

Moreover, this study employs some simplified assumptions, such as using a linear tire formulation in the prediction model and relying on tested data from EDGAR for defining the reference trajectory and NMPC constraints, due to the unavailability of reliable experimental data for semi-trucks. A crucial direction for future development in this topic would involve incorporating more complex and precise tire models, as well as defining constraints and reference trajectories based on experimental data from actual semi-truck systems.

Finally, an essential future endeavor is to test the proposed adaptive NMPC with real semi-truck data. A more extensive study, particularly one that includes closed-loop testing with a real vehicle, is necessary to observe the controller's behavior in real operational conditions and to conclusively assess its performance. This aspect is critical, as the tested scenarios in this study are largely hypothetical cases and require validation under real-world conditions.

List of Figures

Figure 2.1:	Semi-truck simplified schematic	4
Figure 2.2:	Components of vehicle combination.....	5
Figure 3.1:	Semi-truck dynamic single track model	13
Figure 3.2:	MHE and MPC frameworks	18
Figure 4.1:	Entire motion control architecture and three undertaken steps	21
Figure 4.2:	Model inputs reported in [29] for validation process.....	21
Figure 4.3:	NMPC procedure	22
Figure 4.4:	Visual representative of nonlinear inequality constraints.....	26
Figure 4.5:	MHE procedure.....	28
Figure 5.1:	Validation results - reference values from [29]	36
Figure 5.2:	TUM-Control simulation environment - example for EDGAR	37
Figure 5.3:	Monteblanco racetrack with reference trajectory and velocity profile	38
Figure 5.4:	Performance results of developed NMPC with different combinations in the Table 5.2	40
Figure 5.5:	TUM-Control developed simulation environment for Semi-truck	41
Figure 5.6:	Semi-truck's constrained parameters results when Case 1 setting utilized	41
Figure 5.7:	NMPC performance results under Scenario 1 and 2.....	44
Figure 5.8:	NMPC performance results under Scenario 3 and 4.....	44
Figure 5.9:	Lateral deviation average and maximum absolute values for all scenarios	45
Figure 5.10:	Lateral deviation from reference trajectory for scenario 1 / Red line : Reference trajectory in the prediction horizon / Blue line: Predicted trajectory with NMPC.....	45
Figure 5.11:	ARDE in [%] for all configurations in Table 5.4 and the default parameter set of the first scenario without using MHE.	49
Figure 5.12:	Box plot of ARDE results for all configurations in Table 5.4 and the default parameter set of the first scenario without using MHE.	49
Figure 5.13:	ARSE in [%] for Scenario 1/ Red line: Using MHE with selected configuration/ Blue line: With mismatch and No Adaptation (Default) and Yellow line: No mismatch (Correct).	51
Figure 5.14:	ARSE in [%] for Scenario 2/ Red line: Using MHE with selected configuration/ Blue line: With mismatch and No Adaptation (Default) and Yellow line: No mismatch (Correct).	52
Figure 5.15:	ARSE in [%] for Scenario 3/ Red line: Using MHE with selected configuration/ Blue line: With mismatch and No Adaptation (Default) and Yellow line: No mismatch (Correct).	53
Figure 5.16:	ARSE in [%] for Scenario 4/ Red line: Using MHE with selected configuration/ Blue line: With mismatch and No Adaptation (Default) and Yellow line: No mismatch (Correct).	54
Figure 5.17:	ARSE in [%] for Scenario 1/ Red line: Using MHE with selected configuration/ Blue line: With mismatch and No Adaptation (Default) and Yellow line: No mismatch (Correct)- First twenty seconds	55

Figure 5.18:	Open-loop mutable parameter estimation plots in the absence of disturbance — Scenario 1 (a) and Scenario 2 (b).....	58
Figure 5.19:	Open-loop mutable parameter estimation plots in the presence of disturbance — Scenario 3 (a) and Scenario 4 (b).....	59
Figure 5.20:	Lateral velocity estimations against simulation time in the absence of disturbance - Scenario 1 (a) and Scenario 2 (b).....	60
Figure 5.21:	Lateral velocity estimations against simulation time in the presence of disturbance - Scenario 3 (a) and Scenario 4 (b).....	61
Figure 5.22:	Improvements in ARDE for the estimator aware and unaware of v_{lat}	63
Figure 5.23:	Open-loop estimated mutable parameters for scenario 1 with and without v_{lat}	64
Figure 5.24:	Lateral deviation from reference trajectory line in all scenarios.	67
Figure 5.25:	Box plots of absolute lateral deviation from reference trajectory line in all scenarios. (a)With mismatch - No adaptation, (b)With mismatch - With Adaptation and (c)No mismatch.	68
Figure 5.26:	Improvement in absolute lateral deviation from the reference trajectory line in all scenarios.	68
Figure 5.27:	Closed-loop mutable parameter estimation plots in the absence of disturbance / Scenario 1 (a) and Scenario 2 (b) / Blue : Default parameters, Red : Correct parameters and Green : Estimated parameters.....	69
Figure 5.28:	Closed-loop mutable parameter estimation plots in the presence of disturbance / Scenario 3 (a) and Scenario 4 (b)/ Blue : Default parameters, Red : Correct parameters and Green : Estimated parameters.....	70

List of Tables

Table 2.1:	Literature review of semi-truck’s different vehicle models.....	6
Table 3.1:	Kinematic vs Dynamic vehicle model comparison[44, 45].....	11
Table 3.2:	Vehicle and Tire parameters[12, 21].....	15
Table 4.1:	Estimation model fields, defined to be compliant with the ACADOS tool settings	31
Table 5.1:	Maximum deviation (Max. Dev.) values regarding the results in Figure 5.1	37
Table 5.2:	Tested prediction horizon and cost function weight combinations	39
Table 5.3:	Summary of the parameter mismatch scenarios.....	41
Table 5.4:	Tested estimation horizon and cost function weight combinations	48
Table 5.5:	Total open-loop estimation runtime in [s] for all configurations in Table 5.4.	48
Table 5.6:	Open-loop prediction Assessment - ARDE metric values	56
Table 5.7:	Closed-loop simulation times	71

Bibliography

- [1] A. Mačiulis, A. V. Vasiliauskas and G. Jakubauskas, "The impact of transport on the competitiveness of national economy," *Transport*, vol. 24, no. 2, pp. 93–99, 2009, DOI: 10.3846/1648-4142.2009.24.93-99.
- [2] G. M. I. Inc., "Semi-trailer Market Size," *Global Market Insights*, vol. -, no. 1, p. 260, 2024, DOI: https://www.gminsights.com/industry-analysis/semi-trailer-market?utm_source=globnewswire.com&utm_medium=referral&utm_campaign=paid_globnewswire.
- [3] A. Rahimi and Y. He, "A Review of Essential Technologies for Autonomous and Semi-autonomous Articulated Heavy Vehicles," *Progress in Canadian Mechanical Engineering*, vol. 3, no. 1, pp. -, 2020, DOI: 10.32393/csme.2020.1203.
- [4] E. Yurtsever, J. Lambert, A. Carballo and K. Takeda, "A Survey of Autonomous Driving: Common Practices and Emerging Technologies," *IEEE Access*, vol. 8, no. 1, pp. 58443–58469, 2020, DOI: 10.1109/ACCESS.2020.2983149.
- [5] P. Karle, T. Betz, M. Bosk, F. Fent, N. Gehrke, et al., "EDGAR: An Autonomous Driving Research Platform – From Feature Development to Real-World Application," *Arxiv*, vol. -, no. 1, pp. -, 2024, DOI: 10.1109/OJITS.2022.1234567.
- [6] Y. Dai and S.-G. Lee, "Perception, Planning and Control for Self-Driving System Based on On-board Sensors," *Advances in Mechanical Engineering*, vol. 12, no. 9, pp. -, 2020, DOI: 10.1177/1687814020956494.
- [7] P. Stano, U. Montanaro, D. Tavernini, M. Tufo, G. Fiengo, et al., "Model predictive path tracking control for automated road vehicles: A review," *Annual Reviews in Control*, vol. 55, no. -, pp. 194–236, 2023, DOI: 10.1016/j.arcontrol.2022.11.001.
- [8] M. M. Morato, J. E. Normey-Rico and O. Sename, "Model predictive control design for linear parameter varying systems: A survey," *Annual Reviews in Control*, vol. 49, no. -, pp. 1367–5788, 2020, DOI: 10.1016/j.arcontrol.2020.04.016.
- [9] J.-P. Rodrigue, *The Geography of Transport Systems*, 5th Edition, London, Routledge, p. 480, 2020, ISBN: 9780429346323. DOI: 10.4324/9780429346323. Available: <https://doi.org/10.4324/9780429346323>.
- [10] E. Commission, D.-G. for Mobility and Transport, *White paper on transport – Roadmap to a single European transport area – Towards a competitive and resource efficient transport system*, Publications Office of the European Union, 2011, DOI: doi/10.2832/30955.
- [11] G. Giuliano, *The challenges of freight transport in cities*, pp. 11–34, 2023, ISBN: 9781800370166. DOI: 10.4337/9781800370173.00008.
- [12] A. Gustafsson and E. Olsson, "Electrically Propelled Semi-trailer: Dimensioning and Packaging of an Electrical Powertrain in a Semi-trailer," Master's Thesis, CHALMERS UNIVERSITY OF TECHNOLOGY, 2023.
- [13] National Highway Traffic Safety Administration. "Large Trucks Traffic Safety Facts," 2025. [Online]. Available: <https://www.nhtsa.gov> [visited on 01/05/2025].

- [14] B. Moradi and M. Mehrandezh, "A Non-linear MPC Local Planner for Tractor-Trailer Vehicles in Forward and Backward Maneuvering," 2022. arXiv: 2212.11427 [cs.LG]. Available: <https://arxiv.org/abs/2212.11427>.
- [15] J. Betz, H. Zheng, A. Liniger, U. Rosolia, P. Karle, et al., "Autonomous Vehicles on the Edge: A Survey on Autonomous Vehicle Racing," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 3, pp. 458–488, 2022, DOI: 10.1109/OJITS.2022.3181510.
- [16] S. V. Raković. "Robust Model Predictive Control", in: *Encyclopedia of Systems and Control*. Ed. by J. Baillieul and T. Samad. Cham: Springer International Publishing, 2021, pp. 1965–1975. ISBN: 978-3-030-44184-5. DOI: 10.1007/978-3-030-44184-5_2. Available: https://doi.org/10.1007/978-3-030-44184-5_2.
- [17] E. González, J. Sanchis, S. García-Nieto and J. Salcedo, "A Comparative Study of Stochastic Model Predictive Controllers," *Electronics*, vol. 9, no. 12, 2020, DOI: 10.3390/electronics9122078. Available: <https://www.mdpi.com/2079-9292/9/12/2078>.
- [18] L. Hewing, K. P. Wabersich, M. Menner and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, no. 1, pp. 269–296, 2020.
- [19] M. Yue, X. Hou and W. Hou, "Composite path tracking control for tractor-trailer vehicles via constrained model predictive control and direct adaptive fuzzy techniques," *Journal of Dynamic Systems, Measurement, and Control*, vol. 139, no. 11, p. 111008, 2017.
- [20] T. Wu and J. Y. Hung, "Path following for a tractor-trailer system using model predictive control," pp. 1–5, 2017, DOI: 10.1109/SECON.2017.7925337.
- [21] L. Liu, B. Wang and Y. He, "Research on path-tracking control of articulated vehicle with a trailer based on advanced model prediction control strategy," *lateral*, vol. 20, p. 1, 2021.
- [22] J. Yang, S. Kim and K. Huh, "Development of a unified lane-keeping and collision avoidance system for semi-trailer truck," *IEEE Access*, vol. 8, pp. 149751–149763, 2020.
- [23] C. Hynén Ulfsjö and T. Westny. "Modeling and Lateral Control of Tractor-Trailer Vehicles during Aggressive Maneuvers," 2020.
- [24] M. Van de Molengraft-Luijten, I. J. Besselink, R. Verschuren and H. Nijmeijer, "Analysis of the lateral dynamic behaviour of articulated commercial vehicles," *Vehicle system dynamics*, vol. 50, no. sup1, pp. 169–189, 2012.
- [25] M. M. Islam, X. Ding and Y. He, "A closed-loop dynamic simulation-based design method for articulated heavy vehicles with active trailer steering systems," *Vehicle system dynamics*, vol. 50, no. 5, pp. 675–697, 2012.
- [26] K. Gasik, "Comparison of LQR and LQR-Mrac for Linear Tractor-Trailer Model," MA thesis, California Polytechnic State University, 2019.
- [27] P. Nilsson and K. Tagesson, "Single-track models of an A-double heavy vehicle combination," Chalmers University of Technology, 2014.
- [28] J. Diestra and S. Skouras, "Trajectory planning for automated driving of articulated heavy vehicles," 2019.
- [29] A. Latif, N. Chalhoub and V. Pilipchuk, "Control of the nonlinear dynamics of a truck and trailer combination," *Nonlinear Dyn*, vol. 99, no. -, pp. 2505–2526, 2020, DOI: 10.1007/s11071-019-05452-1.
- [30] N. A. Spielberg, M. Brown, N. R. Kapania, J. C. Kegelman and J. C. Gerdes, "Neural network vehicle models for high-performance automated driving," *Science robotics*, vol. 4, no. 28, eaaw1975, 2019.

- [31] J. Wang and Y. Zhang, "A Tutorial on Gaussian Process Learning-based Model Predictive Control," *arXiv preprint arXiv:2404.03689*, 2024.
- [32] E. Arcari, M. V. Minniti, A. Scampicchio, A. Carron, F. Farshidian, et al., "Bayesian multi-task learning mpc for robotic mobile manipulation," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3222–3229, 2023.
- [33] M. Lin, Z. Sun, R. Hu and Y. Xia, "Safety enhancement for nonlinear systems via learning-based model predictive control with Gaussian process regression," *Neurocomputing*, vol. 591, p. 127706, 2024.
- [34] I. Clavera, A. Nagabandi, R. S. Fearing, P. Abbeel, S. Levine, et al., "Learning to Adapt: Meta-Learning for Model-Based Control," *CoRR*, vol. abs/1803.11347, 2018. arXiv: 1803.11347. Available: <http://arxiv.org/abs/1803.11347>.
- [35] Y. Jin, Y. Li, B. He, X. Yang and L. Zheng, "Mass estimation of tractor-semitrailer systems: An approach of dynamics and data fusion-driven in real environments," *Measurement*, vol. 238, p. 115367, 2024.
- [36] A. H. Korayem, A. Khajepour and B. Fidan, "A review on vehicle-trailer state and parameter estimation," *IEEE Transactions on intelligent transportation systems*, vol. 23, no. 7, pp. 5993–6010, 2021.
- [37] L. Ge, Y. Zhao, S. Zhong, Z. Shan, F. Ma, et al., "Motion Control of Autonomous Vehicles Based on Offset Free Model Predictive Control Methods," *Journal of Dynamic Systems, Measurement, and Control*, vol. 144, no. 11, p. 111003, 2022, DOI: 10.1115/1.4055166. eprint: https://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/144/11/111003/6911733/ds_144_11_111003.pdf. Available: <https://doi.org/10.1115/1.4055166>.
- [38] L. Ljung, "Prediction error estimation methods," *Circuits, Systems and Signal Processing*, vol. 21, no. 1, pp. 11–21, 2002, DOI: 10.1007/BF01211648. Available: <https://doi.org/10.1007/BF01211648>.
- [39] A. B. Martinsen, A. M. Lekkas and S. Gros, "Combining system identification with reinforcement learning-based MPC," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 8130–8135, 2020, DOI: <https://doi.org/10.1016/j.ifacol.2020.12.2294>. Available: <https://www.sciencedirect.com/science/article/pii/S2405896320329542>.
- [40] A. Tuveri, C. S. Nakama, J. Matias, H. E. Holck, J. Jäschke, et al., "A regularized Moving Horizon Estimator for combined state and parameter estimation in a bioprocess experimental application," *Computers and Chemical Engineering*, vol. 172, p. 108183, 2023, DOI: <https://doi.org/10.1016/j.compchemeng.2023.108183>. Available: <https://www.sciencedirect.com/science/article/pii/S0098135423000522>.
- [41] E. L. Haseltine and J. B. Rawlings, "Critical evaluation of extended Kalman filtering and moving-horizon estimation," *Industrial & engineering chemistry research*, vol. 44, no. 8, pp. 2451–2460, 2005.
- [42] L. Zhang, J. Xie and S. Dubljevic, "Tracking model predictive control and moving horizon estimation design of distributed parameter pipeline systems," *Computers & Chemical Engineering*, vol. 178, p. 108381, 2023.
- [43] J. P. Maree, S. Gros and H. T. Walnum, "Adaptive control and identification for heating demand-response in buildings," in *2021 European Control Conference (ECC)*, 2021, pp. 1931–1936.
- [44] S. LaValle, *Planning Algorithms*, Cambridge University Press, 2006, ISBN: 9781139455176. Available: <https://books.google.de/books?id=-PwLBAAAQBAJ>.
- [45] R. Rajamani, *Vehicle Dynamics and Control*, (Mechanical Engineering Series), Springer US, 2011, ISBN: 9781461414322. Available: <https://books.google.de/books?id=eoy19aWAjBgC>.
- [46] E. Fernandez-Camacho and C. Bordons-Alba, *Model predictive control in the process industry*, Springer, 1995.

- [47] J. B. Rawlings, D. Q. Mayne and M. Diehl, *Model Predictive Control: Theory and Design*, Nob Hill Publishing, 2017.
- [48] F. Allgöwer and A. Zheng, *Nonlinear Model Predictive Control: Theory and Applications*, (Lecture Notes in Control and Information Sciences). vol. 26, Springer, 2012, ISBN: 978-3-540-20917-7. DOI: 10.1007/978-3-540-24751-3.
- [49] P. Kühn, M. Diehl, T. Kraus, J. P. Schlöder and H. G. Bock, "A real-time algorithm for moving horizon state and parameter estimation," *Computers and Chemical Engineering*, vol. 35, no. 1, pp. 71–83, 2011, DOI: <https://doi.org/10.1016/j.compchemeng.2010.07.012>. Available: <https://www.sciencedirect.com/science/article/pii/S0098135410002607>.
- [50] C. McGurk, H. Ahmed, M. Foo, A. Pike, Q. Lu, et al., "A comparative analysis of moving average filter and Kalman filter for large diesel engine test cell back-pressure control," *International Journal of Engine Research*, vol. 24, no. 7, pp. 3186–3196, 2023, DOI: 10.1177/14680874221145749. Available: <https://doi.org/10.1177/14680874221145749>.
- [51] K. Atkinson, W. Han and D. Stewart, *Numerical Solution of Ordinary Differential Equations*, (Pure and Applied Mathematics: A Wiley Series of Texts, Monographs and Tracts), Wiley, 2011, ISBN: 9781118164525. Available: <https://books.google.de/books?id=QzjGgLIKCYQC>.
- [52] M. Valipour and L. A. Ricardez-Sandoval, "Assessing the Impact of EKF as the Arrival Cost in the Moving Horizon Estimation under Nonlinear Model Predictive Control," *Industrial & Engineering Chemistry Research*, vol. 60, no. 7, pp. 2994–3012, 2021, DOI: 10.1021/acs.iecr.0c06095. Available: <https://doi.org/10.1021/acs.iecr.0c06095>.
- [53] B. Zarrouki. "TUM CONTROL," <https://github.com/bzarr/TUM-CONTROL>.
- [54] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings and M. Diehl, "CasADi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019, DOI: 10.1007/s12532-018-0139-4. Available: <https://doi.org/10.1007/s12532-018-0139-4>.
- [55] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, et al., "acados—a modular open-source framework for fast embedded optimal control," *Mathematical Programming Computation*, vol. 14, no. 1, pp. 147–183, 2022, DOI: 10.1007/s12532-021-00208-8. Available: <https://doi.org/10.1007/s12532-021-00208-8>.
- [56] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, et al., "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, 2020, DOI: 10.1038/s41586-020-2649-2. Available: <https://doi.org/10.1038/s41586-020-2649-2>.
- [57] E. Bøhn, S. Gros, S. Moe and T. A. Johansen, "Reinforcement Learning of the Prediction Horizon in Model Predictive Control," *IFAC-PapersOnLine*, vol. 54, no. 6, pp. 314–320, 2021, DOI: <https://doi.org/10.1016/j.ifacol.2021.08.563>. Available: <https://www.sciencedirect.com/science/article/pii/S2405896321013380>.
- [58] M. Alhajeri and M. Soroush, "Tuning Guidelines for Model-Predictive Control," *Industrial & Engineering Chemistry Research*, vol. 59, no. 10, pp. 4177–4191, 2020, DOI: 10.1021/acs.iecr.9b05931. Available: <https://doi.org/10.1021/acs.iecr.9b05931>.
- [59] B. Zarrouki, M. Spanakakis and J. Betz. "A Safe Reinforcement Learning driven Weights-varying Model Predictive Control for Autonomous Vehicle Motion Control," 2024. arXiv: 2402.02624 [cs . RO]. Available: <https://arxiv.org/abs/2402.02624>.