CHALMERS
UNIVERSITY OF TECHNOLOGY

Predictive Energy Management of Heavy Vehicle in Uncertain Traffic with Learning-based Optimal Control

Master's thesis in Systems, Control and Mechatronics

Dian Wang, Xinzhu Niu

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's thesis 2025

Predictive Energy Management of Heavy Vehicle in Uncertain Traffic with Learning-based Optimal Control

DIAN WANG
XINZHU NIU

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Division of Systems and Control
Chalmers University of Technology
Gothenburg, Sweden 2025

Predictive Energy Management of Heavy Vehicle in Uncertain Traffic with Learning-based Optimal Control
DIAN WANG
XINZHU NIU

Supervisor: Erik Börve, Volvo Group Trucks Technology
Erik Jonsson Holm, Volvo Group Trucks Technology
Esteban Gelso, Volvo Group Trucks Technology
Examiner: Nikolce Murgovski, Department of Electrical Engineering

Master's Thesis 2025
Department of Electrical Engineering
Division of Systems and Control
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: A Volvo truck, by Volvo Group

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2025

Predictive Energy Management of Heavy Vehicle in Uncertain Traffic with Learning-based Optimal Control
DIAN WANG
XINZHU NIU
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Heavy-duty vehicles (HDVs) contribute to over 30% of CO_2 emissions despite representing only 4% of the vehicles on the road in the European Union. To address this imbalance, eco-driving strategies have been developed to improve the energy efficiency of HDVs and reduce their environmental impact. This thesis focuses on a specific scenario in which an ego HDV follows a leading vehicle (another HDV or a passenger car). By analyzing the longitudinal dynamics of the leading vehicle using its speed measurements and road slope information from the onboard digital map, the goal is to accurately estimate the leading vehicle's future speed trajectory. Combined with a predictive energy management controller onboard the ego vehicle, the system can optimize acceleration and deceleration to minimize braking events and maintain a safe inter-vehicle distance, thereby reducing overall fuel consumption.

To achieve this, a learning-based observer–predictor architecture is proposed. The observer filters noisy speed measurements and fuses them with road slope data to estimate the leading vehicle's maximum power capability in real time. The predictor then uses this estimate, along with future slope information within the horizon, to generate the future speed trajectory of the leading vehicle. This speed prediction is used by the ego vehicle's predictive controller to follow the leading vehicle with reduced air drag, enabling energy-efficient and safe following behavior.

The development and validation of the framework were carried out in MATLAB. Performance was evaluated using both simulation and real-world data from recorded driving cycles. The results demonstrate that the proposed structure can predict the leading vehicle's speed trajectory with good accuracy and has the potential to enable significant improvements in energy efficiency and overall system performance. The main contribution of this thesis is the development of a learning-based observer–predictor structure that can predict the leading vehicle's future speed trajectory to help the ego vehicle plan its own speed in a more energy-efficient manner.

Keywords: predictive energy management, leading vehicle speed prediction, observer and predictor design, eco-driving

Acknowledgements

First and foremost, we would like to express our profound gratitude to our academic supervisor, Nikolce Murgovski. His insightful guidance, constructive feedback, and continuous encouragement have guided this thesis from beginning to end, making each step clearer and much easier.

We are equally appreciative of our industrial supervisors at Volvo—Erik Börve, Erik Jonsson Holm, and Esteban Gelso. Thank you for giving us the wonderful opportunity to conduct our master's thesis project at Volvo, for meeting with us every week without fail, and for generously sharing your insightful perspectives. Outside of our scheduled meetings, you were always ready to answer questions and provide the resources we needed. Your experience in the automotive field has enriched our understanding and contributed immensely to our learning and growth.

Our sincere thanks also go to our colleagues at Volvo—Rafael Klüppel Smijtink, Shenhai Ran, Adam Lagerberg, and Achim Beutner—for taking the time to help us clarify key issues. Your willingness to assist was invaluable.

Finally, we wish to thank our families in China for their unwavering support and encouragement throughout this journey. Without your love and belief in us, completing this thesis would not have been possible.

Thank you all.

Dian Wang & Xinzhu Niu, Gothenburg, June 2025

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

ACC	Adaptive Cruise Control
ANN	Artificial Neural Network
DRL	Deep Reinforcement Learning
GPR	Gaussian Process Regression
GR	Gear Ratio
HDV	Heavy-duty Vehicles
ICE	Internal Combustion Engine
IQR	Inter-Quartile-Range
LP	Linear Programming
LV	Leading Vehicle
ML	Machine Learning
MMSE	Minimum Mean Square Error
MPC	Model Predictive Control
MSE	Mean Square Error
NN	Neural Network
PSD	Power Spectral Density
V2V	Vehicle-to-Vehicle
VAR	Vector Autoregressive

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Indices

i,j	Indices for speed cluster or speed measurement
t	Index for time step
s	Traveled distance along road centerline
ζ	Spatial look-ahead coordinate along the route

Sets

$\mathcal{B}$	Set of admissible values for coefficient vector $\mathbf{b}$
$\mathcal{Z}$	Set of selected measurement points
$\mathcal{D}$	Set of all measurement points

Parameters

Δt	Time discretization step (time interval)
Δv	Speed discretization step (speed interval)
dv	Velocity-bin width used for observations
W	Sliding-window length for online training/estimation
m_L	Mass of the leading vehicle
c_{Ld}	Aerodynamic drag coefficient of the leading vehicle
c_{Lr}	Rolling resistance coefficient of the leading vehicle
A_{Lf}	Frontal area of the leading vehicle
g	Gravitational acceleration (9.81 m/s^2)

ρ_a	Air density
$P_{L\,\max}$	Maximum engine power of the leading vehicle
a_{comfort}	Comfort-constrained acceleration parameter
$t_{h\,\min}$	Minimum time headway required by the controller
$v_{\text{limit}}(\zeta)$	Legal/map speed limit
$v_{\min}(s), v_{\max}(s)$	Road speed limits from digital map
N_c	Number of speed clusters used in observer
H	Prediction horizon length
μ	Mean of measurement noise
σ	Standard deviation of measurement noise
G_i	Gear ratio in gear i
η_i	Drivetrain efficiency in gear i
r_w	Effective wheel radius
ω_e	Engine angular speed
n_e	Engine speed (rpm)

Variables

$v_L(s)$	Speed of the leading vehicle at distance s
$a_L(s)$	Acceleration of the leading vehicle at distance s
$\tilde{v}_L$	Measured (noisy) speed of the leading vehicle
$\tilde{a}_L$	Calculated acceleration of the leading vehicle using speed
$\tilde{f}_{Lf}$	Estimated force-to-mass ratio of the leading vehicle
$\tilde{v}_{Lsel}$	Selected useful speed measurement of the leading vehicle
$\tilde{f}_{Lfsel}$	Selected useful calculated force-to-mass ratio of the leading vehicle
$\tilde{d}_L(s)$	Cumulative distance s
$\tilde{\alpha}_L(s)$	Road slope angle at distance s
$\mathbf{b}$	Coefficient vector for leading vehicle force-to-mass model
φ	Regressor vector for observer linear model
$\phi(v_L)$	Integrated regressor vector over speed interval
$\tilde{f}_{L\text{sup}j}$	Maximum force-to-mass ratio in j-th speed cluster
$\tilde{v}_{L\text{sup}j}$	Corresponding speed value in j-th cluster
$\hat{v}_L(\zeta)$	Predicted future speed trajectory of the leading vehicle
$\hat{a}_L(\zeta)$	Predicted acceleration trajectory of the leading vehicle

$\delta_L(\zeta)$	Overspeed classification result (binary indicator)
F_{Lw}	Wheel traction force of the leading vehicle
$F_{Lw\,\max}$	Maximum wheel traction force of the leading vehicle
F_{Lbrk}	Braking force of the leading vehicle
F_{roll}	Rolling resistance force of the leading vehicle
$F_{L\mathrm{air}}$	Aerodynamic drag acting on the longitudinal dynamics
F_{g}	Gravitational force of the leading vehicle
F_{α}	Composite of gravitational and rolling forces of leading vehicle
E_v	Kinetic energy of the leading vehicle
v_{set}	Ego vehicle set speed reference
$v_{\mathrm{ref}}, t_{\mathrm{ref}}$	Reference speed and time trajectory for MPC

Contents

List of Figures

List of Tables

1

Introduction

This chapter presents the background, related work, problem statement, and scope of the thesis, explaining the motivations and objectives of the research.

1.1 Background

Road freight transportation plays a crucial role in global logistics but is also a major contributor to CO_2 emissions and fuel consumption [1]. Heavy-duty Vehicles, while representing less than 5% of the global vehicle ownership, are responsible for more than 25% of on-road CO_2 emissions globally in 2020 [2]. In the European Union, HDVs represent only 4% of the total fleet but are responsible for 30% of CO_2 emissions [3]. Over time, the projected contribution of HDVs to on-road emissions in the EU will grow from the current one-third to roughly 40% by 2020 and 45% by 2030 [4].

To address this issue and meet sustainability goals, there is a growing need to improve the energy efficiency of HDVs. While much effort on hardware improvements has been made to minimize energy losses, an attractive, hardware-independent alternative is to apply "Eco-driving" technology. It is found that eco-driving is a relatively low-cost and immediate measure to reduce fuel consumption and emissions. The major factors influencing fuel consumption and emissions that a driver has control over are acceleration/deceleration, driving speed, route choice, and idling [5]. Eco-driving strategies typically adjust vehicle speed based on road topography [6]. For instance, slowing down before a steep descent and then disengaging the clutch to "free-roll" throughout the downhill segment converts potential energy at the hilltop into kinetic energy, thereby minimizing brake usage and energy loss [7]. Another key tactic is maintaining a close following distance behind a leading HDV to reduce aerodynamic drag on the ego vehicle. Relevant research has shown that by forming a platoon and optimizing the energy management strategy with techniques such as model predictive control, energy efficiency can be improved by up to 10% in highway driving simulations [8].

Eco-driving strategies that minimize the following distance can reduce aerodynamic drag and improve fuel economy. Still, they also increase the likelihood of rear-end collisions when the lead vehicle brakes abruptly, which requires more frequent braking and potentially increases fuel consumption. Mitigating these safety risks requires ego vehicle access to the leading HDV's future accurate speed plan via reliable V2X

communication. However, accurately obtaining such a speed plan is difficult due to the surrounding traffic conditions, and the behavior of the human driver is inherently unpredictable [9]. Even when a future speed plan exists, V2X links may suffer from latency, packet loss, or complete outages, either due to technical limitations or simply low adoption rates in certain regions.

1.2 Previous Work

By adopting a robust speed prediction method, vehicles can effectively maintain energy efficiency even in scenarios lacking direct communication, as illustrated in Figure 1.1. In such cases, the future speed trajectory of the leading vehicle remains unknown and must be estimated in real-time. Existing research on leading vehicle prediction generally falls into three categories.

The first category assumes the leading vehicle maintains a constant speed throughout the prediction horizon [10]. While simple, this approach is inadequate for heavy-duty vehicles (HDVs) operating in hilly terrains, where vehicle mass and limited engine power result in significant speed variations.

The second category applies data-driven, learning-based approaches such as Vector Autoregressive (VAR) models [11] and Deep Reinforcement Learning (DRL) [12]. These methods rely on large amounts of historical data to infer future trajectories and often achieve higher accuracy in controlled or well-sampled scenarios. However, such methods pose several challenges. First, they demand substantial computational resources and are sensitive to training coverage. In particular, generalization in online settings can be poor. If a specific driver or vehicle type is not present in the training data or simulation domain, prediction performance may degrade significantly. Second, these approaches often lack physical interpretability, making it difficult to validate, verify, or integrate with existing safety or control modules.

The third category consists of physics-based modeling approaches, which predict the leading vehicle's speed profile based on estimated engine power and road gradient [13]. More recent developments in this direction propose stochastic, chance-constrained observers that learn a probability distribution over vehicle acceleration or power capability from sensor measurements [14]. These approaches offer better theoretical guarantees and can be integrated with eco-driving controllers to yield modest energy savings. However, they also face critical limitations. Most notably, they often assume uniform probability over the feasible acceleration set, which does not reflect real driving behavior. Moreover, they are highly sensitive to noise and cannot easily distinguish between reliable and misleading observations. As a result, their predictive performance deteriorates in complex terrain, especially during downhill coasting or variable driver behavior.

This thesis builds upon the physics-based framework by incorporating machine learning modules into both the observer and the predictor. Instead of treating all measurements equally, the proposed system actively evaluates the likelihood that a given observation is informative or "useful" for prediction. This improves robustness against noise and enhances adaptability to context. By learning from real-world

data in an online fashion, the method aims to more accurately model the underlying distribution of driver actions and environmental influences. The resulting speed predictions are then used by the eco-driving controller to support energy-efficient driving, even in the absence of explicit inter-vehicle communication.

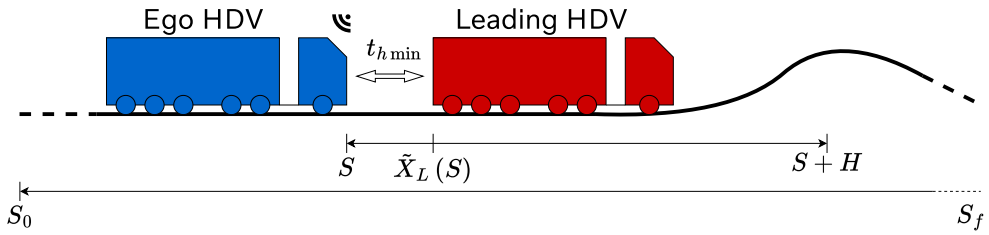

Figure 1.1: Ego HDV follows a leading HDV in hilly terrain (adapted from [13]).

1.3 Problem Statement

The goal of this thesis is to create an integrated, learning-based observer-predictor structure that delivers highly accurate forecasts of a leading Heavy-duty Vehicle's future speed trajectory on hilly roads when no V2V communication is available. The predicted trajectory will be combined with an existing predictive energy management controller to cut fuel consumption while maintaining a safe following distance.

In this structure, the *observer* must infer, in real-time and from noisy measurements, the leading HDV's maximum acceleration capabilities on uphill segments. The resulting estimate is then provided to the *predictor*, which, together with forthcoming road-slope information, generates a reliable forecast of the leading vehicle's future speed profile.

Based on the objectives, the research is structured around the following questions:

- **Data filtering** - Which filtering approach best suppresses sensor measurement noise in the observed vehicle speed signal without distorting its dynamic characteristics?
- **Observer design** - How can machine learning techniques be embedded within a physics-based observer to estimate the leading vehicle's maximum acceleration capability with high accuracy and robustness?
- **Trajectory prediction** - How can the estimated maximum acceleration capability and road-slope preview be fused to predict the leading vehicle's future speed trajectory across the look-ahead horizon?
- **Real-time implementation** - What update logic and computational architecture ensure that the observer-predictor loop runs within the timing constraints of an automotive electronic control unit?
- **Performance and evaluation** - To what extent does the proposed method improve the accuracy and robustness of leading vehicle speed prediction, relative to baseline strategies, in both simulation and real-world driving data?

1.4 Scope and Limitations

This thesis develops an observer–predictor structure that requires only two inputs—measured speed and a preview of upcoming road slope—to estimate a leading vehicle's dynamic properties. Hence, the method is agnostic to vehicle class (passenger car or heavy-duty vehicle); the only operational difference lies in the applicable highway speed limit.

Key assumptions and boundaries are as follows:

- **Sensor noise and residual error:** Measured speed signals always contain noise. While the filtering and learning-enhanced observer aims to suppress disturbances, perfect accuracy is unattainable. Residual estimation error will persist, and in practice, the ACC system's selected "leading" vehicle determines which target is observed and predicted.
- **Terrain focus:** The observer infers the leading vehicle's maximum acceleration capability, which concentrates on uphill segments. Real-world uphill platooning is relatively rare, so real driving-cycle logs will serve for preliminary validation, but controlled simulation data, where slope and traffic conditions can be held constant, will be used for final performance evaluation.
- **Dynamic modeling simplifications:** Only longitudinal dynamics are modeled; lateral motions (e.g., lane changes or cornering) are excluded. Environmental factors such as weather, road-surface variability, and tire–road friction fluctuations are assumed constant. The road-slope preview is drawn from an ideal, onboard digital map and is treated as noise-free.
- **Evaluation constraints:** Due to time and resource limitations, no on-road heavy-duty vehicle field tests will be conducted. All validation will rely on Volvo's simulation platform and existing logged datasets from real truck driving cycles. The available driving-cycle logs were collected from ICE trucks rather than electric trucks, which is expected to have a minor effect on the powertrain modeling.

1.5 Thesis Structure Overview

Following this introduction, **Chapter 2** establishes the theoretical groundwork by summarizing the fundamental concept and theory of this thesis. **Chapter 3** then frames the research problem, describing the observer–predictor system architecture and the objectives it must fulfill. **Chapter 4** explains the proposed methodology in detail, covering the design of the noise-robust observer, the physics-informed predictor, and their real-time integration. **Chapter 5** presents results from simulations and logged driving cycles, assessing accuracy and benchmarking the approach against existing techniques. **Chapter 6** interprets these findings, discusses limitations, and possible failing cases. Finally, **Chapter 7** concludes the thesis, summarizing the main contributions and outlining directions for future research and development.

2

Theoretical Background

This chapter reviews the fundamental theoretical concepts and models related to this thesis, including vehicle longitudinal dynamics, powertrain modeling, data filtering, anomaly removal techniques, machine learning approaches, and convex optimization problems.

2.1 Leading Vehicle Longitudinal Dynamics

This section derives the distance domain model for the longitudinal motion of the leading vehicle by taking into account traction, braking, aerodynamic drag, rolling resistance, and gravitational forces. Specifically, an expression for the acceleration is derived as a function of wheel force, brake force, road slope, and air drag, setting the stage for parameter identification in later chapters.

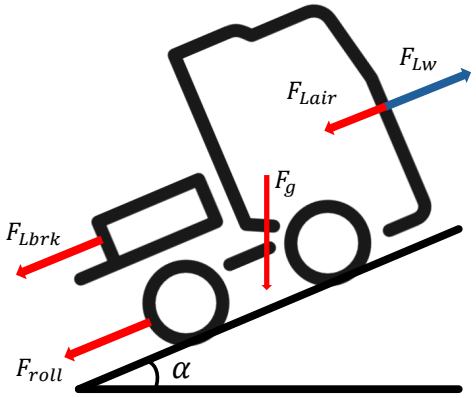

Figure 2.1: The force analysis of the leading HDV's longitudinal dynamics

Only the forces acting along the vehicle's longitudinal axis are considered, as lateral effects are neglected in this thesis. Let s denote the traveled distance along the road center-line, $v_L(s)$ the speed of the leading vehicle (LV), and m_L its mass. Adopting the distance domain (so that $\frac{\mathrm{d}(\cdot)}{\mathrm{d}t} = v_L \frac{\mathrm{d}(\cdot)}{\mathrm{d}s}$), the energy balance reads

$$\frac{\mathrm{d}E_v(s)}{\mathrm{d}s} = F_{Lw}(s) + F_{L\mathrm{brk}}(s) - F_{\alpha}(s) - F_{L\mathrm{air}}(s) \tag{2.1}$$

where the kinetic energy is $E_v(s) = \frac{1}{2} m_L v_L^2(s)$. Each term in Equation (2.1) represents a physical contributor to the longitudinal dynamics and is detailed below.

(i) Wheel force F_{Lw}: This is the net traction delivered to the tire by the power-train after accounting for drivetrain efficiency. It is positive when propelling the LV and zero during free-rolling.

(ii) Brake force $F_{L\mathbf{brk}}$: Produced by the service brakes, it is always opposed to the direction of travel. Following the sign convention in Equation (2.1), $F_{L\mathrm{brk}} < 0$ implies mechanical energy removal.

(iii) Gravitational and rolling forces F_{α}: A road segment of slope angle $\alpha_L(s)$ introduces the composite force

$$F_{\alpha}(s) = m_L \; g\Big[\sin\alpha_L(s) + c_{Lr}\cos\alpha_L(s)\Big] \tag{2.2}$$

where $g = 9.81\,\mathrm{m/s^2}$ and c_{Lr} is the rolling-resistance coefficient, assumed constant.

(iv) Aerodynamic drag $F_{L\mathbf{air}}$: Assuming incompressible flow, the quadratic air drag is

$$F_{L\mathrm{air}}(s) = \frac{1}{2}\,\rho_a\,c_{Ld}\,A_{Lf}\,v_L^2(s) \tag{2.3}$$

with ρ_a the air density, c_{Ld} the drag coefficient, and A_{Lf} the frontal area of the LV.

Differentiating E_v and substituting into Equation (2.1) yields the distance-domain acceleration model

$$a_L(s) = \frac{F_{Lw}(s) + F_{L\mathrm{brk}}(s) - F_{L\mathrm{air}}(s)}{m_L} - g\Big[\sin\alpha_L(s) + c_{Lr}\cos\alpha_L(s)\Big] \tag{2.4}$$

where $a_L(s) = \frac{\mathrm{d}v_L}{\mathrm{d}s}v_L$ is the derivative of speed with respect to distance.

When the LV accelerates at full throttle and the brakes are released $F_{L\mathrm{brk}}(s) = 0$, the wheel force equals its envelope $F_{Lw_{\max}}(v_L)$. Assuming $F_{Lw}(s) = F_{Lw_{\max}}(v_L)$, we define the maximum wheel force as

$$F_{Lw_{\max}}(v_L) = F_{L0} + \frac{P_{L\max}}{v_L(s)} \tag{2.5}$$

where F_{L0} is a constant traction force term used for curve fitting and $P_{L\max}$ the engine's maximum delivered power; both are *unknown* to the ego vehicle and will be identified by the observer. Substituting $F_{Lw_{\max}}$ into Equation (2.4) gives

$$a_L(s) = \frac{F_{Lw_{\max}}(v_L) - \frac{1}{2}\rho_a c_{Ld} A_{Lf} v_L^2(s)}{m_L} - g\Big[\sin\alpha_L(s) + c_{Lr}\cos\alpha_L(s)\Big] \tag{2.6}$$

2.2 Vehicle Model

The leading vehicle is modeled as a point mass with the powertrain illustrated in Figure 2.2. The driving cycle logs and simulation logs in this thesis are collected using an ICE truck. However, since the longitudinal dynamics apply to all vehicles regardless of powertrain type, the analysis is also valid for battery electric vehicles. The only slight difference is the modeling of the ICE and electric motor, which

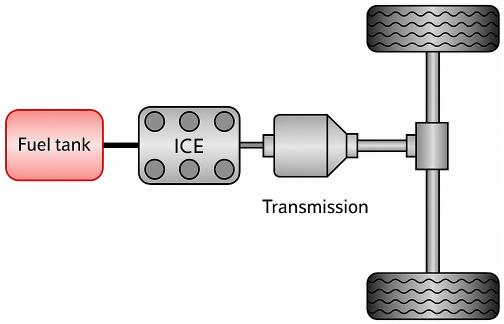

Figure 2.2: The assumed powertrain configuration of the leading vehicle.

will be elaborated on in the next chapter. The maximum wheel force is defined in Equation 2.5, which directly represents the maximum power produced by the powertrain. Therefore, the energy losses occurring in the conversion from the ICE or electric motor to the actual wheel force are not considered in this thesis.

2.3 Model Simplification for HDVs with ICE

Previous research in this field focused on HDV with electric motors, where a constant-torque region at low speed necessitates a nonzero bias term [13], which holds maximum wheel force to follow Equation (2.5) strictly. In this thesis, research would be generic for HDV with all types of engines, but evaluation is implemented on HDVs with ICE, whose torque–speed characteristic lacks such a plateau [15]. This difference motivates simplifying the Vehicle Dynamics model to reduce computational complexity.

Inner approximation of engine torque limit

In this study, the conditions in which the engine operates at maximum power (blue marker in Figure 2.3) and the torque produced under these conditions (red marker in Figure 2.3) are considered. Two models are implemented to fit the ICE torque data:

$$T(\omega) = a + \frac{b}{\omega}, \quad a \geq 0,\, b > 0, \quad (a,b) = \arg\min_{a,b\geq 0} \sum_i \left[T(\omega_i) - (a + \tfrac{b}{\omega_i})\right]^2 \tag{2.7}$$

$$T(\omega) = \frac{b}{\omega}, \quad b > 0, \quad b = \arg\min_{b>0} \sum_i \left[T(\omega_i) - \tfrac{b}{\omega_i}\right]^2 \tag{2.8}$$

Fitting results show that model (2.8) attains residuals statistically indistinguishable from model (2.7), with the estimated bias term a in model (2.7) approaching zero.

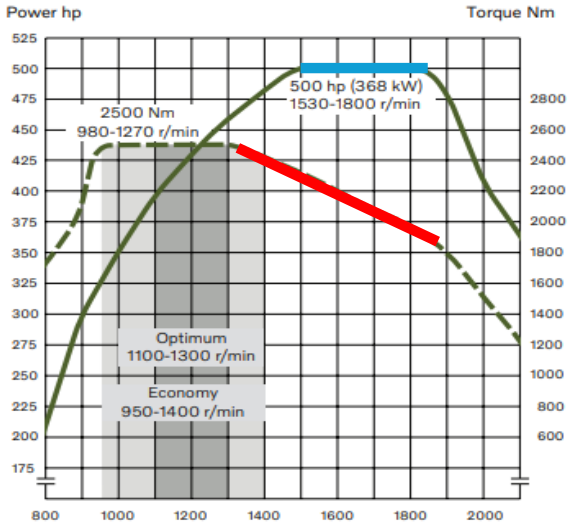

Figure 2.3: Engine specification [24]

Wheel Force–Speed Mapping

Using the fitted torque curve $T_e(\omega_e)$, wheel force for gear i is computed by

$$F_{w,i}(\omega_e) = \frac{T_e(\omega_e)\, G_i\, \eta_i}{r_w} \tag{2.9}$$

and vehicle speed by

$$v_i = \frac{\omega_e}{G_i}\, r_w, \quad \omega_e = \frac{2\pi}{60} n_e \tag{2.10}$$

Figure 2.4 compares the force–speed curves (60 - 120 km/h) obtained from models (2.7) and (2.8) across gears 9–12 of the engine used in simulation.

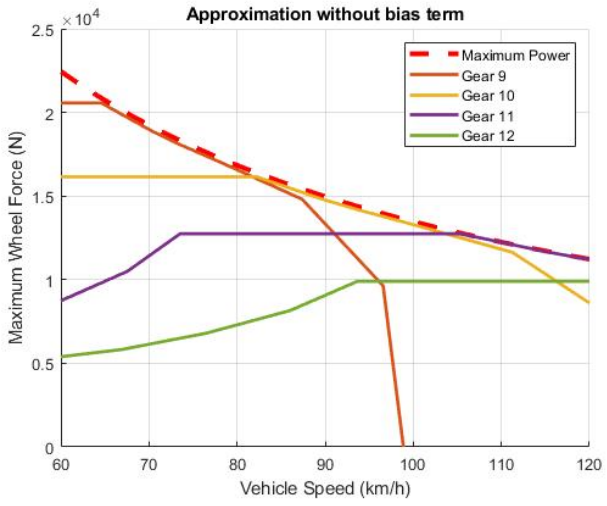

Figure 2.4: Approximation results using simplified model 2.8

In this way, the maximum wheel force can be simplified from 2.5 to:

$$F_{Lw_{\max}}(v_L) = \frac{P_{L\max}}{v_L(s)} \tag{2.11}$$

These observations confirm that the constant term a is unnecessary for the ICE approximation. Since all the available data logs were recorded with ICE trucks, the results of this thesis will be analyzed without the constant term. However, for electric trucks, the constant term can still be included in the modeling. Omitting a reduces the parameter estimation to a single scalar b, which provides the following benefits for subsequent convex optimization: faster solver convergence and reduced computational complexity.

2.4 Data Filtering

Analysis of multiple sensor measurements and corresponding ground truth values from available datasets indicates that the measurement noise follows a zero-mean Gaussian white noise distribution. Noise statistics, including mean and standard deviation, can be reliably estimated. Under these conditions, the Wiener filter provides the optimal linear solution for signal denoising in the sense of minimum mean square error (MMSE) [16].

Wiener Filter for Speed Measurement Denoising

In this thesis, the Wiener filter considers a measured signal $y(t)$ that is modeled as the sum of the true signal $x(t)$ and additive noise $n(t)$:

$$y(t) = x(t) + n(t) \tag{2.12}$$

Where $n(t)$ is zero-mean white Gaussian noise with known power spectral density $S_n(f)$, and $x(t)$ is the underlying noise-free signal.

To design a linear time-invariant filter $H(f)$ that minimizes the expected squared estimation error:

$$\min_{H(f)} \mathbb{E}\left[|x(t) - \hat{x}(t)|^2\right] \tag{2.13}$$

The signal is transformed into the frequency domain via the Fourier transform. Let f denote the frequency variable, and let $Y(f) = \mathcal{F}\{y(t)\}$, $X(f) = \mathcal{F}\{x(t)\}$, and $N(f) = \mathcal{F}\{n(t)\}$ denote the Fourier transforms of the measured signal, true signal, and noise, respectively.

The optimal Wiener filter in the frequency domain is given by:

$$H(f) = \frac{S_x(f)}{S_x(f) + S_n(f)} \tag{2.14}$$

where:

- $S_x(f) = \mathbb{E}[|X(f)|^2]$: Power spectral density (PSD) of the true signal;
- $S_n(f) = \mathbb{E}[|N(f)|^2]$: PSD of the noise.

The filtered estimate $\hat{X}(f) = H(f) \cdot Y(f)$ attenuates frequencies dominated by noise while preserving signal-rich components. Finally, the denoised time-domain signal $\hat{x}(t)$ is obtained by the inverse Fourier transform of $\hat{X}(f)$.

According to classical estimation theory, the Wiener filter achieves the minimum MSE among all linear estimators, provided that the signal and noise are stationary and their spectral characteristics are known.

2.5 Anomaly Removal Techniques

Anomaly detection methods are widely used in the banking, insurance, and stock trading industries to identify fraudulent activities. In the context of leading vehicle speed measurements, noise is introduced due to sensor inaccuracies and other external factors, and this noise persists even after applying a Wiener filter. Consequently, when calculating the maximum acceleration capability of the leading vehicle based on these speed measurements, it is necessary to remove anomalous data points to uncover the true underlying behavior. This requires incorporating anomaly removal techniques into the analysis process. This section summarizes the commonly used anomaly detection approaches and selects a certain method to apply to the leading vehicle speed measurement data.

2.5.1 Parametric Statistical Method

Z-score For approximately Gaussian data, the two-sided Z-score:

$$z_i = \frac{x_i - \mu}{\sigma} \tag{2.15}$$

Where μ and σ are the sample mean and standard deviation, measures the deviation of each observation x_i. Points with $|z_i| > 3$ (or another user-defined threshold) are flagged as anomalies because they sit far away from the majority of the data.

IQR The inter-quartile-range (IQR) [17] rule is widely used for quick anomaly screening, relies only on sample quartiles, and therefore makes no distributional assumptions. For a univariate sample $\{x_i\}$, let Q_1 and Q_3 denote the first and third quartiles; the inter-quartile range is

$$\mathrm{IQR} = Q_3 - Q_1$$

Quartiles divide a sorted dataset into four equal parts; Q_1 marks the 25th percentile, and Q_3 marks the 75th percentile. Observations lying outside the *fence*

$$\left[\, Q_1 - 1.5\,\mathrm{IQR},\ Q_3 + 1.5\,\mathrm{IQR} \,\right]$$

are flagged as mild outliers, while those beyond $Q_1 - 3\,\mathrm{IQR}$ or $Q_3 + 3\,\mathrm{IQR}$ are deemed extreme. Because the calculation involves only sorting and simple arithmetic, it is computationally trivial and well-suited for on-board, real-time anomaly removal in speed-sensor data streams.

2.5.2 Unsupervised Learning Method

K-means clustering K-means [18] is one of the most popular clustering methods, it partitions an unlabeled data set into K clusters by iteratively assigning each point to the nearest centroid (measured with Euclidean distance) and recomputing centroids. In an anomaly-detection setting, points that lie beyond a chosen distance threshold from their closest centroid—or that end up in a very small cluster when $K = 2$ — are treated as outliers. The method is computationally light and parameter-sparse (only K), which has the potential to be used in real-time on-board for leading vehicle maximum force-to-mass ratio computation.

Isolation Forest Isolation Forest [19] builds many random isolation trees that keep splitting a small, randomly chosen subset of the data. Normal points look similar to many others and end up deep in the trees, while anomalies are "few and different" and are isolated after only a handful of splits. The algorithm simply converts the *average path length* $h(x)$ needed to isolate a point into an anomaly score

$$s(x) = 1 - \frac{h(x)}{h_{\max}}$$

so values of $s(x)$ close to 1 indicate likely outliers. Isolation Forest is fast, requires only two settings (subsample size and number of trees), and makes no assumptions about the data distribution, which makes it well suited for real-time sensor streams.

2.5.3 Gaussian Process Regression

Gaussian Process Regression (GPR) is a non-parametric Bayesian method that models a distribution over functions, making it especially effective for noisy time-series data like leading vehicle speed measurements. A key strength of GPR is its ability to quantify predictive uncertainty, providing not just a mean estimate of a signal but also a variance that reflects confidence. This feature is crucial for distinguishing true anomalies from normal fluctuations caused by sparse data or sensor noise.

Let the latent (noise-free) speed signal be $f(x)$. The prior can be written as:

$$f(x) \;\sim\; \mathcal{GP}\Big(m(x),\, k(x, x')\Big)$$

where $m(x) = \mathbb{E}[f(x)]$ (set to zero after de-meaning) and k is a kernel function(for example the squared-exponential) encoding input similarity. Given n observations at times $\mathbf{x} = [x_1, \ldots, x_n]$, the corresponding outputs $\mathbf{f} = [f(x_1), \ldots, f(x_n)]^\top$ follow:

$$\mathbf{f} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}), \quad \mathbf{K}_{ij} = k(x_i, x_j)$$

Assuming Gaussian measurement noise $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$, the observed speeds $\mathbf{y} = \mathbf{f} + \boldsymbol{\varepsilon}$ are distributed as:

$$\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{K} + \sigma_n^2 \mathbf{I})$$

For a new time point x_* define

$$\mathbf{k}_* = [k(x_1, x_*), \ldots, k(x_n, x_*)]^\top, \qquad k_{**} = k(x_*, x_*)$$

The posterior predictive distribution is:

$$\mu_* = \mathbf{k}_*^\top \Big(\mathbf{K} + \sigma_n^2 \mathbf{I}\Big)^{-1} \mathbf{y}, \qquad \sigma_*^2 = k_{**} - \mathbf{k}_*^\top \Big(\mathbf{K} + \sigma_n^2 \mathbf{I}\Big)^{-1} \mathbf{k}_* \tag{2.16}$$

The posterior mean μ_* gives a smooth estimate of the true speed, while the variance σ_*^2 measures uncertainty. Anomalies are flagged based on standardized residuals:

$$r_i = \frac{|\, y_i - \mu_i \,|}{\sigma_i}$$

Values exceeding a defined threshold (e.g. $r_i > 3$) are considered anomalies. This thresholding adapts naturally to the data: since σ_i increases in regions with fewer or more uncertain observations, the method avoids over-flagging edge points or sparse intervals, making it well-suitable for anomaly removal.

2.6 Machine Learning

Machine learning (ML) provides data-driven techniques for learning predictive models from observations without explicit programming. Supervised learning, which infers a mapping from inputs to known outputs, is employed in this thesis using three foundational methods: linear regression for continuous estimation, logistic regression for binary classification, and artificial neural networks for capturing nonlinear relationships.

2.6.1 Linear Regression

Linear regression models a continuous output y as a weighted sum of input features $\mathbf{x} = [x_1, \ldots, x_p]^\top$:

$$\hat{y} = \beta_0 + \sum_{j=1}^{p} \beta_j x_j \tag{2.17}$$

And estimates parameters by minimizing the mean squared error:

$$\boldsymbol{\beta} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \Big(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij}\Big)^2 \tag{2.18}$$

Closed-form solutions exist via the normal equations, making linear regression a simple yet powerful baseline for regression tasks.

2.6.2 Logistic Regression

Logistic regression adapts linear modeling to binary outcomes by linking input features to class probability through the logistic (sigmoid) function:

$$p(\mathbf{x}) = \frac{1}{1 + \exp\Big(-[\beta_0 + \sum_{j=1}^{p} \beta_j x_j]\Big)} \tag{2.19}$$

with β fitted by maximizing the log-likelihood:

$$\boldsymbol{\beta} = \arg\max_{\beta} \sum_{i=1}^{n} \Big[y_i \ln p(\mathbf{x}_i) + (1 - y_i) \ln(1 - p(\mathbf{x}_i)) \Big] \tag{2.20}$$

Logistic regression yields probabilistic outputs and interpretable coefficients, suitable for classification tasks.

2.6.3 Neural Network

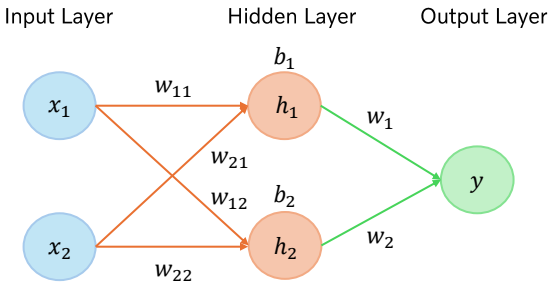

Figure 2.5: Simple sketch of a basic architecture of a neural network

Artificial neural networks (ANNs) are layered, trainable models that learn complex, nonlinear mappings between inputs and outputs by emulating the weighted connections of biological neurons. In a feed-forward network of depth L, an input vector $x \in \mathbb{R}^{d_0}$ is propagated through successive affine transformations and element-wise nonlinearities,

$$x_\ell = \sigma\Big(W_\ell x_{\ell-1} + b_\ell\Big), \qquad \ell = 1, \ldots, L \tag{2.21}$$

where $W_\ell \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$ and $b_\ell \in \mathbb{R}^{d_\ell}$ are the weights and biases, and $\sigma(\cdot)$ is the activation function, which in this thesis is the *tangent sigmoid function*. With at least one hidden layer and a non-polynomial activation, such networks form a universal function approximation.

Learning mechanism Given labeled samples $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$, the trainable parameters of the network are defined as $\theta = \{\theta_\ell\}_{\ell=1}^{L}$, where $\theta_\ell = \{W_\ell, b_\ell\}$ denotes the weights and biases of the ℓ-th layer. These parameters are obtained by minimizing a differentiable loss function, such as cross-entropy for classification. Gradients $\nabla_\theta \mathcal{L}$ are computed efficiently via back-propagation [20] and updated using stochastic gradient descent or an adaptive variant such as Adam; regularization techniques (e.g., Dropout, weight decay, early stopping) are applied to mitigate overfitting.

Probabilistic output For binary decisions, the final layer reduces to a single logistic unit $\hat{p} = \sigma(w^\top x_{L-1} + b)$, providing a probability $\hat{p} \in (0, 1)$ that the input belongs to the positive class. Multi-class tasks employ a softmax function to yield a probability vector that sums to one. A threshold (e.g. $\hat{p} > 0.9$) is chosen in this thesis to convert this soft decision into a crisp label, thereby enabling the removal of anomalies - unwanted data points. Such probabilistic outputs have proved

valuable in vehicular applications ranging from fuel-consumption prediction [21] to driving-power forecasting [22].

Because ANNs capture the nonlinear interplay between vehicle speed, road gradient, and engine state, they outperform linear models such as logistic regression when detecting operating points of special interest—here, those associated with high power demand. Their flexibility and real-time inference speed—assuming a moderate network size (i.e., limited depth L and layer width d_ℓ)—make them attractive for on-board use in the observer–predictor framework developed later in this thesis.

2.7 Convex Optimization Problem

Convex optimization provides the mathematical backbone for estimating the leading vehicle's maximum force–to–mass ratio. A (finite-dimensional) optimization problem is *convex* when both the objective and all inequality constraints are convex functions, while the equality constraints are affine. The standard form of a convex optimization problem reads

$$\begin{aligned} \underset{x\in\mathbb{R}^n}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \le 0, \quad i = 1, \dots, m \\ & Ax = b \end{aligned} \tag{2.22}$$

where

- $f : \mathbb{R}^n \to \mathbb{R}$ is a convex cost function;
- $g_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \dots, m$ are convex inequality functions;
- $Ax = b$ collects all affine equality constraints.

Key properties follow immediately from convexity: any local minimizer is globally optimal, and efficient interior-point or active-set solvers are guaranteed to converge to this optimum.

If f and every g_i are affine, the problem reduces to a *linear programming (LP)*

$$\begin{aligned} \underset{x\in\mathbb{R}^n}{\text{minimize}} \quad & c^{\mathsf{T}}x + d \\ \text{subject to} \quad & Gx \le h \\ & Ax = b \end{aligned} \tag{2.23}$$

whose feasible set is a polyhedron and whose solution algorithms (simplex, barrier methods) enjoy polynomial or even near-linear complexity in practice.

The estimation of the maximum force-to-mass ratio ultimately leads to a linear programming problem because the observer is formulated with a linear objective (least absolute deviation) and polyhedral constraints derived from measured speed limit bounds and HDV's physical properties. Hence, all guarantees of strictly convex optimization—existence of a unique global optimum and availability of fast solvers—apply directly, making the approach both robust and computationally efficient for on-board implementation.

3

System Architecture Overview

This chapter presents the overall system architecture used in this thesis, detailing the module of the observer and predictor of the leading vehicle and the information flow between them.

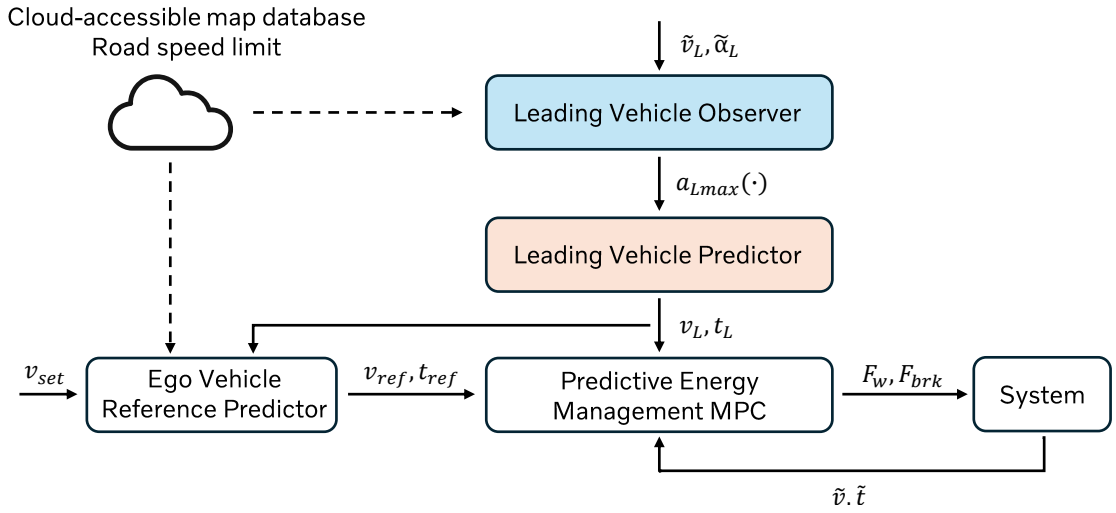

Figure 3.1: System architecture overview [13].

Figure 3.1 sketches the closed-loop structure adopted in this thesis, which is adapted from the eco-driving framework proposed in [13]. The signal flow proceeds from top to bottom and from left to right; solid lines denote on-board measurements or computed signals, while dashed lines indicate information retrieved from an onboard digital map or cloud-accessible map database.

Topographic and traffic information An electronic horizon module supplies future road-slope data $\alpha(s)$ and speed limits $v_{\min}(s), v_{\max}(s)$ for the look-ahead horizon. These previews are broadcast to both the *ego reference predictor* and, via the dashed arrow, to the *leading vehicle observer* block.

Leading vehicle observer A substantial part of this thesis focuses on the leading vehicle observer module. From noisy speed measurements $\tilde{v}_L(s)$ and their filtered derivative $\tilde{a}_L(s)$, together with the previewed road slope, the observer estimates the leading vehicle's *maximum acceleration capability* where the unknown parameters

such as $(P_{L\max}, c_{Ld}A_{Lf})/m_L$ are identified online by solving a convex linear programming problem. Robust data pre-treatment removes measurement spikes and inaccuracies with polynomial fitting and Z-score filtering, or by directly selecting points that operate at maximum power via a pre-trained neural network. The remaining samples are clustered in speed space to keep computational and memory demands low.

Leading vehicle predictor The other key focus of this thesis is the leading vehicle predictor module. Using the freshly estimated $a_{L\max}(\cdot)$, it constructs the leading vehicle's future speed profile $(v_L(\zeta|s), t_L(\zeta|s))$ over the horizon $\zeta \in [s, s+H]$, based on the minimum of the legal speed limit and a comfort-bounded acceleration profile along the slope preview. An online learning approach is integrated to handle real-world uncertainties and deviations from ideal engine power used, enhancing the accuracy and robustness of the prediction.

Model Predictive Controller (MPC) The MPC receives three sets of information: the road-aware ego reference $(v_{\text{ref}}, t_{\text{ref}})$, the leading vehicle's trajectory (v_L, t_L), and the ego vehicle's current state $(\tilde{v}, \tilde{t})$. At every preset time or distance interval, it solves a receding-horizon optimal control problem that minimizes a weighted sum of fuel rate, ride comfort, and slack variables while satisfying:

- longitudinal dynamics
- road-speed limits
- actuator bounds $F_{w\min} \le F_w \le F_{w\max}$ and $F_{\text{brk}} \le 0$
- a safety headway constraint $t(\zeta|s) \ge t_L(\zeta|s) + t_{h\min}$

The optimizer outputs wheel traction F_w and brake force F_{brk}, which are applied to the powertrain ("System" block). Measured ego speed $\tilde{v}$ and time $\tilde{t}$ close the feedback loop.

Information flow and hierarchy Two prediction loops run in parallel: one centered on the unknown leading vehicle (observer + predictor) and one centered on the ego vehicle (reference predictor). The MPC sits at the junction of these loops, blending their outputs to generate a fuel-optimal yet safe speed plan. By estimating the leading vehicle's maximum acceleration capability rather than assuming constant speed, the architecture anticipates speed losses on hills, allowing the ego vehicle to pre-adjust its velocity, avoid unnecessary braking, and exploit aerodynamic drag reduction — thereby achieving the desired eco-driving strategy.

This architecture forms the baseline on which the subsequent chapters build. Chapter 4 details the design of the learning-based leading vehicle observer and predictor, and Chapter 5 presents their accuracies and potential impact on energy efficiency.

4

Learning-based Observer-Predictor Structure Design

This chapter introduces a learning-based observer–predictor structure designed to estimate and forecast the speed trajectory of a leading vehicle using onboard sensors and a look-ahead digital map. The architecture combines a modular observer for real-time parameter estimation with an adaptive predictor that accounts for road conditions and driving behavior. By integrating data-driven filters and online learning techniques, the system improves accuracy and adaptability in dynamic highway environments. The overall design emphasizes computational efficiency and suitability for real-world deployment.

4.1 High-level Architecture Overview

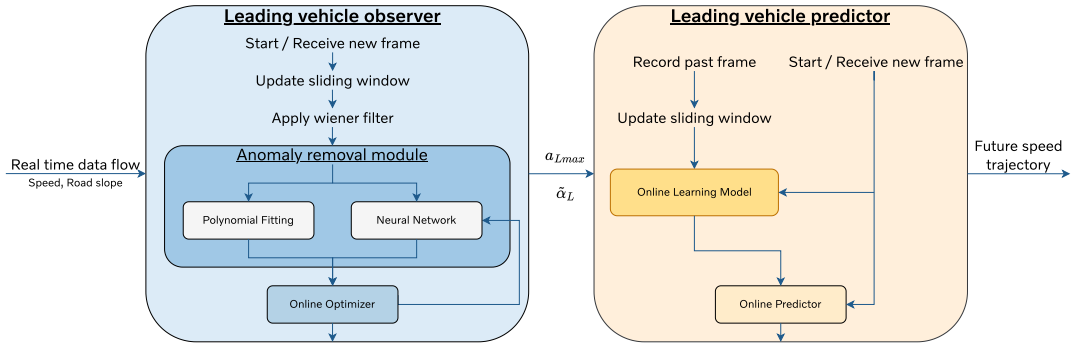

Figure 4.1: High-level architecture of the learning-based observer-predictor structure

Figure 4.1 summarizes the real-time **Learning-based Observer-Predictor Structure**. The system processes real-time data collected by the onboard sensors, which measure the speed of the leading vehicle and receive slope preview information from the onboard computer. These two inputs are fed into the system.

First, the data is passed to the leading vehicle observer. Within the observer, the data is stored in a data buffer, which keeps track of the leading vehicle's object ID.

If the ID of the leading vehicle changes, the newly received data will not be stored in the same buffer. This ensures that the buffer only retains information relevant to estimating the current leading vehicle.

A sliding window mechanism is employed to manage the amount of data stored in the buffer. As more data accumulates, solving the polynomial fitting problem becomes more time-consuming and requires more memory. The sliding window controls the buffer size, ensuring that the system processes a manageable and relevant subset of the data.

Once the data buffer contains sufficient continuous measurements from the same leading vehicle, the data is passed to the **Polynomial Fitting Filter**. The **Neural Network Filter** depends on the output coefficient **b** from the Polyfit Filter to perform its prediction. Therefore, the Polyfit Filter, which uses a statistical and robust approach, runs before the neural network, which typically provides more precise results. These two filters operate sequentially.

Each filter produces suitable "speed, force-to-mass ratio" pairs, which are then passed to the **online optimizer**. The term "force-to-mass ratio" will be introduced in more detail in Chapter 4.2. The optimizer formulates a convex optimization problem based on these inputs and solves it to determine the final parameter **b**. This parameter reflects the maximum acceleration capability of the leading vehicle. The result is then forwarded to the leading vehicle predictor, where the future speed trajectory of the leading vehicle is estimated and generated.

The leading vehicle predictor module receives the estimated maximum acceleration capacity and real-time kinematic data of the leading vehicle from the observer. It initially computes a baseline future speed trajectory under the assumption that the leading vehicle operates at its maximum acceleration. To address deviations caused by real-world factors, such as downhill segments or variable driver behavior, historical data are continuously recorded and used to update an online regression model. This model learns to correct acceleration estimates over time, enabling the generation of a refined and more reliable prediction of the leading vehicle's future speed trajectory.

4.2 Leading Vehicle Observer Design

The detailed design of the learning-based leading vehicle observer is illustrated in Figure 4.2. From the longitudinal modeling in Section 2.1, the *maximum-power* acceleration of the leading vehicle is:

$$a_{L\,\max}(s) = \frac{F_{Lw_{\max}}(v_L) - \frac{1}{2}\rho_a c_{Ld} A_{Lf} v_L^2(s)}{m_L} - g\Big[\sin\alpha_L(s) + c_{Lr}\cos\alpha_L(s)\Big] \tag{4.1}$$

$v_L(s)$ is obtained from vehicle sensor readings, and the road-slope preview $\alpha_L(s)$ is known from the onboard digital map. Moreover, since the main application focuses on highway driving scenarios under normal weather conditions, the rolling resistance term c_{Lr} is considered constant. Therefore, the bracketed gravity–rolling term is treated as a *known* disturbance. Removing it from the current equation yields the

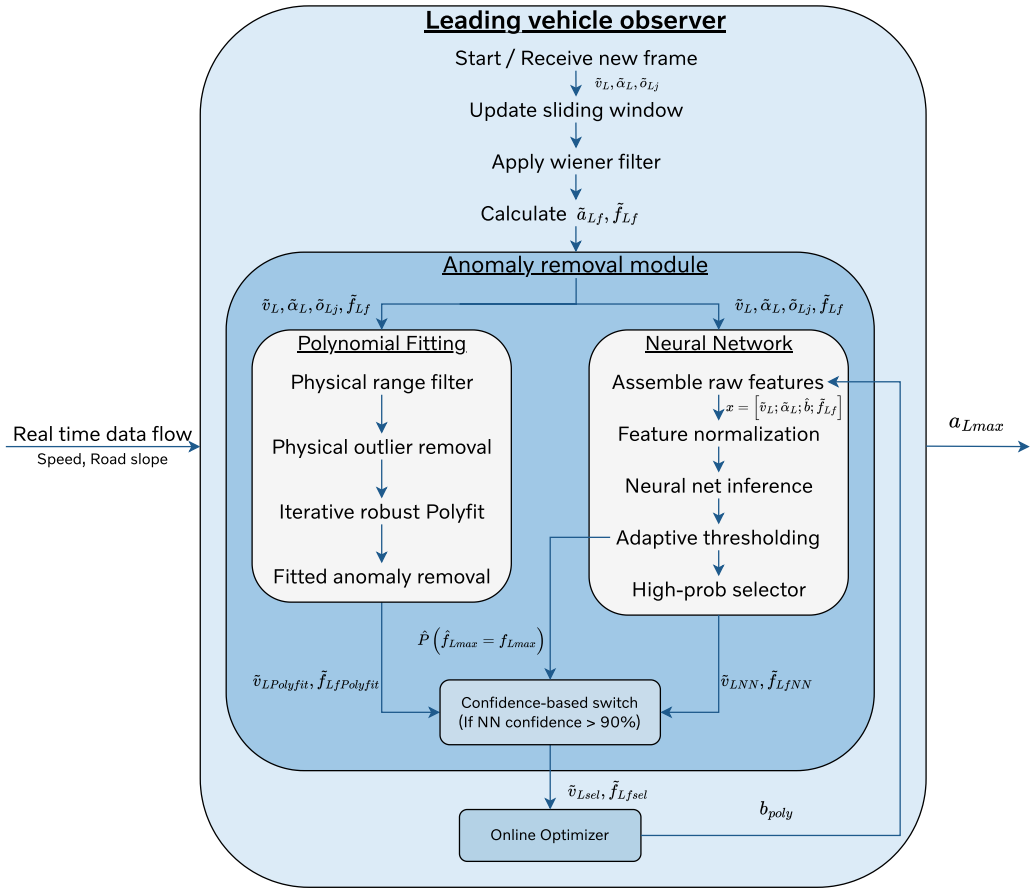

Figure 4.2: The overview of the learning-based observer-predictor structure

maximum force-to-mass ratio:

$$f_{Lmax}(b, v_L) = \frac{\frac{P_{L\max}}{v_L(s)} - \frac{1}{2}\,\rho_a\,c_{Ld}\,A_{Lf}\,v_L^2(s)}{m_L} = \boldsymbol{\varphi}^\top(v_L)\,\mathbf{b} \tag{4.2}$$

with

$$\boldsymbol{\varphi}(v_L) = \begin{bmatrix} v_L^{-1}(s) \\ -v_L^2(s) \end{bmatrix} \tag{4.3}$$

and

$$\mathbf{b} = \begin{bmatrix} \frac{P_{L\max}}{m_L} \\ \frac{\rho_a\,c_{Ld}\,A_{Lf}}{2\,m_L} \end{bmatrix} \tag{4.4}$$

Hence, all measurements enter linearly through the regressor $\boldsymbol{\varphi}(v_L)$, while the unknown vehicle parameters are stacked in $\mathbf{b}$.

The system will get leading vehicle speed measurements and the future road slope profile. The dataset is described as follows:

$$\mathcal{D} = \left\{ \left(\tilde{v}_L^{(1)}, \tilde{\alpha}_L^{(1)}\right), \ldots, \left(\tilde{v}_L^{(i)}, \tilde{\alpha}_L^{(i)}\right) \right\} \tag{4.5}$$

Where the speed sensor is corrupted by Gaussian noise:

$$\tilde{v}_L = v_L + e, \quad e \sim \mathcal{N}(0, \sigma^2) \tag{4.6}$$

Thus, the leading vehicle's maximum acceleration becomes:

$$a_{L\max}(\tilde{v}_L) = f_{L\max}(b, \tilde{v}_L) - g\Big(\sin\tilde{\alpha}_L(s) + c_{Lr}\cos\tilde{\alpha}_L(s)\Big) \tag{4.7}$$

The filtered force-to-mass ratio used by the observer is:

$$\tilde{f}_{Lf}(s) = \tilde{a}_{Lf}(s) + g\Big(\sin\tilde{\alpha}_L(s) + c_{Lr}\cos\tilde{\alpha}_L(s)\Big) \tag{4.8}$$

Using the filtered force-to-mass ratio $\tilde{f}_{Lf}$, calculated from the leading vehicle's speed measurements and future slope preview, the information is passed to both the Polynomial fitting filter and the neural network-based filter. These two filters each identify the point at which the truck is likely operating at maximum power. The selected data is then fed into the optimization problem to estimate the leading vehicle's maximum acceleration capability.

4.2.1 Wiener Filter Design

Let $\tilde{v}_L(t)$ denote the raw speed measurements of the lead vehicle, and $v_L(t)$ the corresponding ground truth values (obtained from high-resolution reference sensors or post-processed data). To construct a Wiener filter tailored to this application, the noise characteristics are estimated using the difference:

$$e(t) = \tilde{v}_L(t) - v_L(t) \tag{4.9}$$

This estimation error is assumed to primarily reflect the additive sensor noise.

Power Spectral Density Estimation

The noise power spectral density $S_n(f)$ is computed using Welch's method [25]:

$$S_n(f) = \mathcal{P}\left[e(t)\right] = \mathcal{P}\left[\tilde{v}_L(t) - v_L(t)\right] \tag{4.10}$$

Similarly, the PSD of the noisy signal is estimated as:

$$S_{\tilde{v}_L}(f) = \mathcal{P}\left[\tilde{v}_L(t)\right] \tag{4.11}$$

Assuming signal and noise are uncorrelated, the clean signal's PSD is approximated as:

$$S_{v_L}(f) \approx S_{\tilde{v}_L}(f) - S_n(f) \tag{4.12}$$

Filter Computation and Application

The Wiener filter in the frequency domain is:

$$H(f) = \frac{S_{v_L}(f)}{S_{v_L}(f) + S_n(f)} \tag{4.13}$$

This frequency-domain filter is applied pointwise to the FFT of the noisy signal:

$$\hat{V}_L(f) = H(f) \cdot \tilde{V}_L(f)$$

Where $\tilde{V}_L(f) = \mathcal{F}\{\tilde{v}_L(t)\}$, and the filtered signal is recovered by inverse FFT:

$$\tilde{v}_{Lf}(t) = \mathcal{F}^{-1}\left[\hat{V}_L(f)\right]$$

Implementation Details

In practice, the FFT is computed using a segment length of 1024 samples with 50% overlap and a Hanning window. The data is sampled at $10Hz$, and the resulting filter $H(f)$ is applied to the full speed signal. The output $\tilde{v}_{Lf}(t)$ exhibits significantly reduced high-frequency noise while preserving the overall shape and temporal structure of the original trajectory.

This tailored Wiener filter implementation enables reliable speed signal denoising in the presence of stationary Gaussian noise, which is critical for accurate observer design in the following components.

4.2.2 Polynomial Fitting Filter

As described in Section 2.5, the polynomial fitting filter employs a polynomial fitting approach combined with the Z-score method to eliminate anomalous data points. It takes as input the filtered leading vehicle speed measurements and the calculated force-to-mass ratio $\tilde{f}_{Lf}$. Within the filter, these inputs first pass through a physical range filter. Although the speed measurements have already been processed using a Wiener filter to suppress Gaussian noise, some nonlinear components remain and can result in unrealistic $\tilde{f}_{Lf}$ values. These residual nonlinearities may arise from

sensor imperfections, unmodeled transient dynamics, or the nonlinear nature of the force-to-mass computation itself, which can amplify small variations when dividing or multiplying filtered signals. To address this, the filter imposes bounds on the acceptable range of $\tilde{f}_{Lf}$ based on physical constraints, such as vehicle mass and engine power, which must conform to known reasonable limits. This establishes a physically meaningful range for the force-to-mass ratio. Additionally, during the later stages of the optimization process, only speed values between 25 km/h and 85 km/h are considered, providing further constraints on the dataset. Consequently, the newly computed "speed–force-to-mass ratio" pairs are filtered to ensure data robustness.

Once the data pairs have been further filtered to ensure physical feasibility, an iterative polynomial fitting process is performed. In this step, a third-order polynomial $p(v)$ is defined to model the relationship between vehicle speed and the force-to-mass ratio. The system repeatedly fits $p(v)$ to the filtered data, computes the residuals, and excludes any data points whose residuals exceed one standard deviation from the mean. This process continues for up to five iterations or until no additional anomalies are identified. After the loop concludes, all data points flagged as anomalies are permanently removed from the dataset. The resulting cleaned dataset, consisting of the final selected data pairs $\tilde{v}_{L\text{Polyfit}}$ and $\tilde{f}_{Lf\text{Polyfit}}$, is then passed to the online optimization routine.

4.2.3 Neural-Network-based Filter

The neural-network-based filter follows a different approach from the polynomial fitting filter, which takes a more data-driven path. Rather than removing all anomalies from the dataset, the neural network directly selects data points that it deems most likely to represent moments when the leading vehicle is operating at maximum power.

The network is trained offline using a relatively small ego vehicle dataset, as the ego vehicle records whether its engine is delivering maximum torque—information not available for the leading vehicle. This labeled dataset serves as the basis for model training.

Input features For every time step, a five-dimensional feature vector is built:

$$\mathbf{x} = \left[\tilde{v}_L^{\mathrm{n}},\ \tilde{\alpha}_L^{\mathrm{n}},\ \hat{b}_1^{\mathrm{n}},\ \hat{b}_2^{\mathrm{n}},\ \tilde{f}_{Lf}^{\mathrm{n}}\right]^\top \tag{4.14}$$

where

- $\tilde{v}_L^{\mathrm{n}} = \dfrac{\tilde{v}_L - \mu_v}{\sigma_v}$ is the normalized leading vehicle speed measurement.
- $\tilde{\alpha}_L^{\mathrm{n}} = \dfrac{\tilde{\alpha}_L - \mu_\alpha}{\sigma_\alpha}$ is the normalized road slope.
- $\hat{b}_1^{\mathrm{n}} = \dfrac{\hat{b}_1 - \mu_{b_1}}{\sigma_{b_1}}$ and $\hat{b}_2^{\mathrm{n}} = \dfrac{\log_{10}\hat{b}_2 - \mu_{b_2}}{\sigma_{b_2}}$ are the normalized elements of the preliminary $\hat{\mathbf{b}}$ vector from the polynomial fitting filter. The second component

spans several orders of magnitude, hence the base-10 log is taken before normalization.

- $\tilde{f}_{Lf}^{\mathrm{n}} = \dfrac{\tilde{f}_{Lf} - \mu_f}{\sigma_f}$ is a physics-inspired helper feature, with

$$\tilde{f}_{Lf} = \frac{\hat{b}_1}{v_L} - \hat{b}_2 v_L^2 \tag{4.15}$$

All statistics (μ and σ) are computed from the training partition only, and reused at inference time.

Network architecture A compact fully-connected multilayer perceptron is employed:

Layer	Size	Activation
Input	5	–
Hidden 1	15	`tansig`
Hidden 2	15	`tansig`
Hidden 3	15	`tansig`
Hidden 4	15	`tansig`
Hidden 5	15	`tansig`
Hidden 6	15	`tansig`
Output	1	`logsig`

With five inputs and the layer widths above, the model contains

$$N_{\mathrm{param}} = \sum_{l=1}^{7} \Big(n_l \left(n_{l-1} + 1 \right) \Big) = 1306$$

trainable weights and biases—small enough to avoid overfitting yet expressive enough to capture the operating envelope.

Training data Five real driving logs are used (two *Euro* cycles and three *Brazil* cycles), covering trucks of $35t$–$75t$. In total, roughly $N \approx 90000$ labeled samples were obtained. The data are randomly split into 70% training and 30% testing. Each sample is labeled.

$$y = \begin{cases} 1, & \text{if } P_{\mathrm{eng}} \geq 0.9 P_{\max} \\ 0, & \text{otherwise} \end{cases}$$

where P_{eng} is the measured engine power of the *ego* vehicle, available offline. Because the positive class is very sparse, we initially bias the output neuron with $\log(p/(1-p))$ (p = positive ratio in the training set) and add an ℓ_2 regularizer ($\lambda = 0.1$).

Optimization Training uses the scaled conjugate-gradient algorithm (`trainscg`) for at most 1500 epochs. The loss is the usual binary cross-entropy

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \Big[y_i \log p_i + (1 - y_i) \log(1 - p_i) \Big].$$

From probability to decision Let $p^{(k)} \in (0,1)$ be the network output at sample k. The set of valid indices detected at the threshold τ is denoted by $\mathcal{I}_\tau$, and its size by $N(\tau)$:

$$\mathcal{I}_\tau \;=\; \left\{k \mid p^{(k)} \geq \tau\right\} \quad \text{with} \quad N(\tau) = |\mathcal{I}_\tau|$$

A very conservative starting threshold $\tau_0 = 1$ is adopted and gradually relaxed in fixed steps.

$$\tau_{j+1} = \tau_j - \Delta\tau, \qquad \Delta\tau = 10^{-5}$$

until at least

$$N(\tau_j) \;\geq\; N_{\min} = 5$$

High-confidence samples have been found. The resulting threshold $\tau_\star$ is then frozen and the corresponding indices $\mathcal{I}_{\tau_\star}$ define the final set of "maximum-power" observations

$$\left(\tilde{v}_{L\,\mathrm{NN}},\; \tilde{f}_{Lf\,\mathrm{NN}}\right)$$

This adaptive rule guarantees a minimum of five reliable data pairs while keeping the threshold as high as possible, thus maintaining a strict quality filter with negligible onboard computational cost.

In summary, the neural network filter brings actual engine-status knowledge into the observer with only $\sim$ 1.3 k parameters and negligible on-board cost, yet it significantly reduces the number of wrongly accepted samples compared with the purely statistical baseline.

4.2.4 Optimization Problem Formulation

After obtaining the selected "speed–force-to-mass ratio" pairs, each representing a data point where the leading vehicle is considered to be operating at maximum power, the coefficient vector $\mathbf{b}$, which contains the unknown parameters of the leading vehicle, can be identified. This is achieved by minimizing the integral of the force to mass ratio over all data points within the interval $[x_L(s_0), x_L(s)]$, under the assumption that the leading vehicle is delivering a force consistent with $\tilde{f}_{Lf}(\zeta)$ for all $\zeta \in [x_L(s_0), x_L(s)]$. In other words, the optimization problem finds the solution that minimizes the area under the fitted curve $\boldsymbol{\varphi}^\top\left(\tilde{v}_{Lf}\right)\mathbf{b}$, while ensuring it remains above all the selected "speed–force-to-mass ratio" pairs. The estimation problem can be formulated as a linear programming (LP) problem:

$$\begin{aligned} \min_{\mathbf{b}} \quad & \int_{x_L(s_0)}^{x_L(s)} \boldsymbol{\varphi}^\top\left(\tilde{v}_{Lf}(\zeta)\right)\mathbf{b}\, d\zeta \\ \text{s.t.} \quad & \boldsymbol{\varphi}^\top\left(\tilde{v}_{Lf}(\zeta)\right)\mathbf{b} \geq \tilde{f}_{Lf}(\zeta), \quad \forall\, \zeta \in [x_L(s_0), x_L(s)] \\ & \mathbf{b} \in \mathcal{B} \end{aligned} \tag{4.16}$$

Here, $\tilde{v}_{Lf}(\zeta)$ denotes the measured and filtered leading vehicle speed, and $\tilde{f}_{Lf}(\zeta)$ represents the calculated force-to-mass ratio, both obtained from either the polynomial fitting filter or the neural network filter. The set $\mathcal{B}$ defines a convex polytope representing the admissible values of the coefficient vector $\mathbf{b}$. The formulated linear programming problem (Equation 4.16) is designed to be both computationally and memory efficient, enabling real-time implementation on onboard systems.

4.2.4.1 Discard irrelevant data points

As stated previously in Section 4.2.1, the speed measurements are passed through a Wiener filter to suppress Gaussian noise. The high computational efficiency allows the filter to reduce noise while preserving the underlying signal pattern. After applying the filter, the data can be further refined using the following steps:

Highway-driving scenario constraint Since the majority of the project's application will be focused on the high-way driving scenario, a safe assumption is that the vehicle's speed will be mainly limited in a certain range $[v_{L\,\min}, v_{L\,\max}]$, with $v_{L\,\min} > 0$, and typically $v_{L\,\min} = 25km/h,\ v_{L\,\max} = 85km/h$. If the speed drops under the $v_{L\,\min}$, it is considered as the outside disturbances that cause the massive speed drop instead of the limited leading vehicle's maximum power capability.

Range of interest $\mathcal{B}$ The force-to-mass ratio defined in this project essentially represents the power capability of the leading vehicle. And due to the design of HDV's engine and the limitation applied to the mass value. Together with the aerodynamic drag and the frontal area of the leading HDV, all these values have to fall into a reasonable range, and as a result, the convex polytope $\mathcal{B}$ will pose a further constraint on the values of the force-to-mass ratio:

$$\varphi^{\top}\left(\tilde{v}_{Lf}\right)\mathbf{b}_{\min} \leq f_{L\,\max}\left(\mathbf{b}, v_{Lf}\right) \leq \varphi^{\top}\left(\tilde{v}_{Lf}\right)\mathbf{b}_{\max} \tag{4.17}$$

where:

$$\begin{aligned} \mathbf{b}_{\min} &= \left[\frac{P_{L\,\min}}{m_{L\,\max}} \quad \frac{\rho_a c_{Ld} A_{Lf}}{2m_{L\,\max}}\right]^{\top}, \\ \mathbf{b}_{\max} &= \left[\frac{P_{L\,\max}}{m_{L\,\min}} \quad \frac{\rho_a c_{Ld} A_{Lf}}{2m_{L\,\min}}\right]^{\top} \end{aligned} \tag{4.18}$$

Which defines the upper and lower bounds of the final fitted coefficient vector $\mathbf{b}$. Data points outside these limits are considered outliers and are discarded. An illustration of the range of interest is shown in Figure 4.3.

Road slope value constraint Data discarding based on road slope ensures that only measurements in which the lead vehicle can sustain its set cruising speed are retained; when a vehicle encounters an incline, the engine must overcome a gravitational component.

$$F_g = mg\Big(\sin\alpha + c_{Lr}\cos\alpha\Big)$$

where m is vehicle mass, g is gravitational acceleration and c_{Lr} is the rolling-resistance coefficient. The traction force required at set speed v_{set} is

$$F_{\text{req}} = \phi^{T}\Big(v_{\text{set}}\Big)\, b$$

where $\phi(v_{\text{set}}) \in \mathbb{R}^n$ is the vector of speed-dependent resistance coefficients and b the vehicle resistance parameter. Defining the worst-case traction demand with parameter $b_{\min}$ (corresponding to minimum engine power and maximum vehicle mass), the minimum critical slope $\alpha_{L\,\min}$ is obtained by solving

$$\phi^{T}\Big(v_{\text{set}}\Big)\, b_{\min} - g\Big(\sin\alpha_{L\,\min} + c_{Lr}\cos\alpha_{L\,\min}\Big) = 0$$

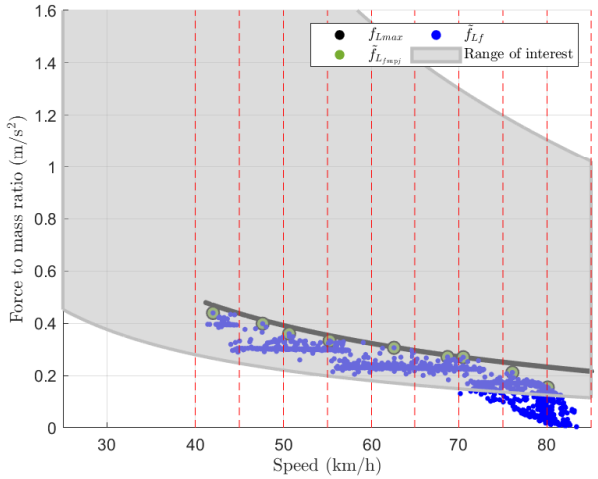

Figure 4.3: Filtered speed measurements within the 25 km/h to 85 km/h range are shown as blue dotted points. The black curve shows the fitted $\tilde{f}_{L\max}$. The range of interest $\mathcal{B}$ is illustrated as the gray area; data points outside this region are considered outliers and are discarded. The red dashed lines denote the segmentation of speed intervals, and the green filled circles indicate the maximum measured force-to-mass ratio within each interval.

Which yields the closed-form

$$\alpha_{L\min} = \sin^{-1}\left(\frac{\frac{\phi^T(v_{\text{set}})\, b_{\min}}{g} - c_{Lr}\sqrt{1 - \left(\frac{\phi^T(v_{\text{set}})\, b_{\min}}{g}\right)^2 + c_{Lr}^2}}{1 + c_{Lr}^2}\right)$$

Measurements with local slope $\alpha_i > \alpha_{L\min}$ are discarded to prevent speed deficits due to insufficient engine power from biasing subsequent analysis.

4.2.4.2 Change of independent variable

To significantly reduce computational complexity, the independent variable is changed from distance ζ to speed v_L. Consequently, the linear programming (LP) problem is equivalently reformulated as:

$$\begin{aligned} \min_{\mathbf{b}} \quad & \int_{v_{L\min}}^{v_{L\max}} \boldsymbol{\varphi}^\top(v_L)\mathbf{b}\, dv_L \\ \text{s.t.} \quad & \boldsymbol{\varphi}^\top(v_L)\mathbf{b} \geq F_L(v_L), \quad \forall v_L \in [v_{L\min}, v_{L\max}] \\ & \mathbf{b}_{\min} \leq \mathbf{b} \leq \mathbf{b}_{\max} \end{aligned} \tag{4.19}$$

where

$$\begin{aligned} & \zeta \in \mathcal{Z} \subseteq [x_L(s_0), x_L(s)] \\ & F_L(v_L) = \left\{ \tilde{f}_{Lf}(\zeta) \mid \tilde{v}_{Lf}(\zeta) \in [v_L, v_L + dv], \zeta \in \mathcal{Z} \right\} \end{aligned} \tag{4.20}$$

which represents the measurements of the filtered force-to-mass ratio within a small speed interval $[v_L, v_L + dv]$. The integral of the objective function can be evaluated

analytically, resulting in:

$$\int_{v_{L\,\min}}^{v_{L\,\max}} \boldsymbol{\varphi}^\top(v_L)\ \mathbf{b}\ dv_L = \boldsymbol{\phi}^\top\ \mathbf{b} \tag{4.21}$$

with

$$\boldsymbol{\phi} = \left[\ln\left(\frac{v_{L\,\max}}{v_{L\,\min}}\right),\quad -\frac{v_{L\,\max}^3 - v_{L\,\min}^3}{3}\right]^\top \tag{4.22}$$

Next, as observed in Figure 4.3, the continuous speed interval $[v_{L\,\min}, v_{L\,\max}]$ is discretized into N_c clusters, each with a width of Δv. The set of measurements associated with the j-th speed cluster is therefore given by:

$$F_{Lj} = \left\{\tilde{f}_{Lf}(\zeta) \mid \tilde{v}_{Lf}(\zeta) \in [v_{L\,\min} + (j-1)\Delta v, v_{L\,\min} + j\Delta v], \zeta \in \mathcal{Z}\right\} \tag{4.23}$$

Moreover, instead of imposing constraints on all measurement data pairs, it is sufficient to ensure that the maximum force-to-mass ratio points in each speed cluster satisfy the constraints, since the LP problem aims to identify the maximum force-to-mass ratio of the leading vehicle. This information is particularly useful for subsequent speed trajectory prediction in the upcoming leading vehicle predictor. Therefore, the speed range is partitioned into clusters, and in each cluster, only the supremum is retained:

$$\tilde{f}_{Lf\,\max j} = \max\left\{\tilde{f}_{Lf}(\zeta)\ :\ \tilde{v}_{Lf}(\zeta) \in \text{cluster } j\right\} \tag{4.24}$$

with the corresponding speed denoted as $\tilde{v}_{Lf\,\max j}$. The maximum force-to-mass ratio points can be identified in Figure 4.3, shown as large green points. The LP problem can then be further reduced to:

$$\begin{aligned} \min_{\mathbf{b}} \quad & \boldsymbol{\phi}^\top \mathbf{b} \\ \text{s.t.} \quad & \boldsymbol{\varphi}^\top\left(\tilde{v}_{Lf\,\max j}\right)\mathbf{b} \geq \tilde{f}_{Lf\,\max j}, \quad j = 1, \ldots, N_c \\ & \mathbf{b} \in \mathcal{B} \end{aligned} \tag{4.25}$$

where $\boldsymbol{\phi}$ is the analytical integration of $\boldsymbol{\varphi}$ over the interval $[v_{L\,\min}, v_{L\,\max}]$, and N_c is the number of clusters. The final LP problem only requires storing N_c pairs of $\tilde{v}_{Lf\,\max j}, \tilde{f}_{Lf\,\max j}$, which is significantly fewer than what is required in the formulation from Equation 4.19. This greatly reduces the computational and memory demands, making the algorithm suitable for real-time onboard prediction.

4.2.4.3 Additional constraints on the LP problem

The formulations in Equation (4.25) apply two types of limits: (i) the *cluster–supremum* requirement and (ii) the *convex polytope* $\mathcal{B}$. To ensure that the fitted curve remains physically realistic across the whole speed range, two further shape constraints are imposed on the first and second derivatives of the model.

Bounded first derivative: The slope of the curve:

$$f'(v) = -\frac{\mathbf{b}_1}{v^2} - 2\,\mathbf{b}_2\,v$$

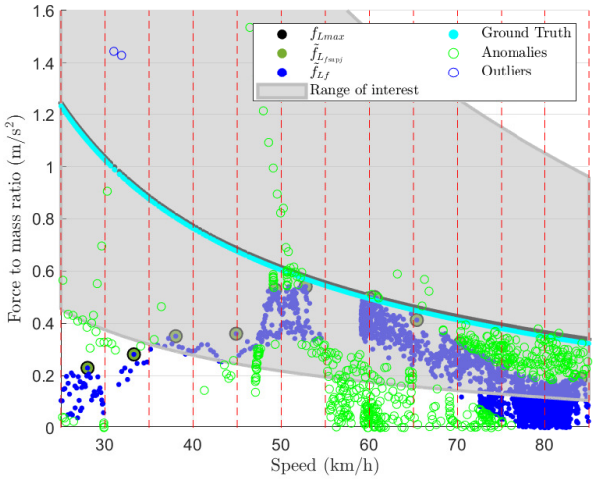

Figure 4.4: The output of the optimization problem is the fitted black curve, which represents the estimated maximum force-to-mass ratio. This curve is defined by the coefficient vector $\mathbf{b}$, calculated using the LP problem described in Equation 4.25. The true maximum power capability of the leading vehicle is represented by the blue curve, which closely follows the black curve. The green hollow points indicate anomalies that were removed after applying the polynomial fitting filter.

quantifies how fast the available force decays with speed. By substituting the extreme admissible parameters $\left(\mathbf{b}_{1\,\max}, \mathbf{b}_{2\,\max}\right)$ and $\left(\mathbf{b}_{1\,\min}, \mathbf{b}_{2\,\min}\right)$. The linear inequality constraint on the first derivative can be obtained as follows:

$$f'_{\text{low}}(v) \leq f'(v) \leq f'_{\text{up}}(v) \quad \forall v \in [v_{L\,\min}, v_{L\,\max}] \tag{4.26}$$

where

$$\begin{aligned} f'_{\text{low}}(v) &= -\frac{\mathbf{b}_{1\,\max}}{v^2} - 2\mathbf{b}_{2\,\max} v \\ f'_{\text{up}}(v) &= -\frac{\mathbf{b}_{1\,\min}}{v^2} - 2\mathbf{b}_{2\,\min} v \end{aligned} \tag{4.27}$$

The constraint prevents the fitted $\tilde{f}_{L\,\max}$ curve from dropping too slowly or too steeply, thus guaranteeing a reasonable deceleration profile.

Convexity (non-negative second derivative): The second derivative of the fitted function is given by

$$f''(v) = \frac{2b_1}{v^3} - 2b_2 \tag{4.28}$$

which results from differentiating the model $f(v) = \frac{b_1}{v} - b_2 v^2$. Physically, convexity ($f''(v) \geq 0$) means that as the vehicle speed increases, the available acceleration decreases at an accelerating rate.

$$-b_1 + b_2\, v_i^3 \;\leq\; 0 \tag{4.29}$$

If convexity is not enforced, the fitted curve may exhibit non-physical behavior, such as bending downward, implying that aerodynamic drag decreases at higher speeds, which contradicts basic vehicle physics. More generally, without such constraints,

the fitted function could deviate arbitrarily and produce unrealistic trends. To ensure physical feasibility and model stability, the condition $f''(v) \geq 0$ is imposed across the full speed interval.

Complete four-constraint LP problem Collecting the new rows (4.26)–(4.29) together with the existing constraints lead to the final linear programming problem:

$$\begin{aligned} & \min_{\mathbf{b}} \quad \boldsymbol{\phi}^\top \mathbf{b} \\ \text{s.t.} \quad & \boldsymbol{\varphi}^\top\left(\tilde{v}_{Lf\max j}\right)\mathbf{b} \geq \tilde{f}_{Lf\max j}, \quad j = 1, \ldots, N_c \\ & \mathbf{b}_{\min} \leq \mathbf{b} \leq \mathbf{b}_{\max} \\ & f'_{\text{low}}(v) \leq f'(v) \leq f'_{\text{up}}(v), \quad \forall v \in [v_{L\min}, v_{L\max}] \\ & -b_1 + b_2\, v^3 \;\leq\; 0 \end{aligned} \tag{4.30}$$

4.3 Leading Vehicle Predictor Design

This section introduces a modular framework for predicting the future speed of the leading vehicle. Based on a model-driven baseline method, two learning-based enhancements are proposed to improve adaptability and robustness in the presence of uncertainty, such as driver variability and road slope-induced behavior.

4.3.1 Baseline Method

The baseline predictor follows the general principle established in prior work on non-communicative eco-driving scenarios, where the leading vehicle's future plan is unknown. In [13], the leading HDV is assumed to aim for maintaining the legal speed limit $v_{\max}$, but may fail to do so on uphill segments due to limited engine power. Consequently, a model-based predictor is used to estimate future speed based on the available power capability and road gradient. The predictor selects the minimum between setting cruise control speed and the integrated acceleration profile, using the estimated a_{Lmax} as input. This setup has shown effective fuel savings when used in model predictive control frameworks, without relying on explicit communication between vehicles.

4.3.1.1 Overview of Existing Approach

Since the leading HDV's future driving plan is unknown, the baseline model assumes that the vehicle attempts to maintain a predefined cruise control speed, denoted as $v_{\max}$. This value typically corresponds to the road speed limit or a slightly lower setting defined by the driver. In the absence of active braking, the vehicle is expected to maintain or attempt to reach this cruising speed whenever feasible.

When facing uphill segments, the HDV may be unable to maintain $v_{\max}$ due to limited engine power. In such cases, the vehicle is assumed to accelerate at the maximum available engine rate a_{Lmax}.

With estimation from observer $\hat{b}$, speed measurement $\tilde{v}_L(s)$, slope information $\tilde{\alpha}_L(s)$ and distance record $\tilde{d}_L(s)$, a_{Lmax} can be calculated based on equation:

$$a_{\mathrm{Lmax}}(s) = \frac{F_{Lw_{\max}}(\tilde{v}_L) - \frac{1}{2}\rho_a c_{Ld} A_{Lf} v_L^2(s)}{m_L} - g\Big[\sin\tilde{\alpha}_L(s) + c_{Lr}\cos\tilde{\alpha}_L(s)\Big] \tag{4.31}$$

The predicted speed is then computed as the minimum of:

- the cruise control speed $v_{\max}(\zeta \mid s)$
- the speed derived by integrating the available acceleration from the last observed speed.

Formally, the speed prediction is given by:

$$\begin{aligned}\hat{v}_L(\zeta \mid s) = \min\Bigg\{ & v_{\max}(\zeta \mid s), \\ & \tilde{v}_L(s) + \int_{\zeta_L(s)}^{\zeta} \min\left\{ \frac{a_{\text{comfort}}}{v_L(p \mid s)}, \frac{a_{\text{Lmax}}(s)}{v_L(p \mid s)} \right\} \mathrm{d}p \Bigg\}, \\ & \forall\, \zeta \in \left(\tilde{x}_L(s),\, s + H\right]\end{aligned} \tag{4.32}$$

Where:

- $\hat{v}_L(\zeta \mid s)$: predicted speed of the leading vehicle at position ζ, given observation time s;
- $\tilde{v}_L(s)$: observed speed at the last known position $\tilde{x}_L$;
- $\zeta_L(s)$: current position of the leading vehicle;
- a_{comfort}: a predefined constant limiting acceleration for comfort; (in this model, it is fixed constant);
- $a_{\text{Lmax}}(s)$: maximum vehicle acceleration available at position s;
- H: prediction horizon in spatial coordinates.

4.3.1.2 Workflow

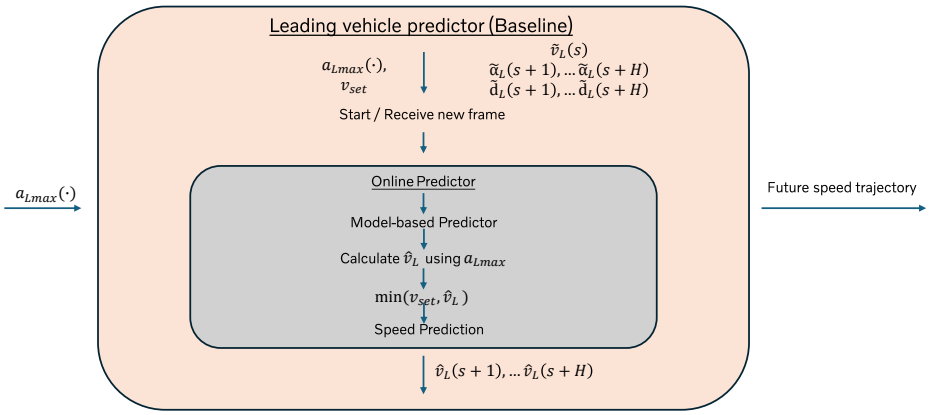

Figure 4.5: The overview of the Baseline Predictor Structure: forecasting leading-vehicle speed by integrating maximum engine acceleration, constant comfort acceleration, and fixed cruise/speed-limit constraint.

4.3.1.3 Limitations

The baseline predictor assumes that the leading vehicle always operates at maximum engine power when not constrained by road limits. However, in real-world driving, especially during downhill coasting or partial throttle, the vehicle often deviates from this assumption, leading to over-conservative speed predictions.

Additionally, the cruise speed $v_{\max}$ used as a limit in the model is a static driver-defined value or a system setting (usually close to the speed limit), which does not adapt to context or driving behavior. Moreover, the comfort acceleration a_{comfort} is implemented as a constant parameter and, in practice, has little effect in constraining the final prediction due to the dominance of the engine limit. The model also lacks the ability to account for driver-specific variability or contextual factors such as traffic or braking, which limits its generalization in varied environments.

Improvement To overcome these limitations, two online learning-based enhancements were introduced:

1. **Overspeed classification module:** Dynamically adjusts the speed limit using data-driven decisions.
2. **Acceleration regression module:** Learns the vehicle's behavior in real time.

4.3.2 Integrated with Online Overspeed Classification

4.3.2.1 Motivation

While the baseline method limits predicted speed using a fixed value v_{set}, logged data indicates that the leading vehicle may exceed this speed, especially when transitioning from uphill to downhill. This occurs due to momentum and gravity and is highly dependent on vehicle mass. These overspeed events are difficult to capture with rule-based models.

Empirical investigation of the feature distribution—specifically vehicle speed, road slope, and lookahead horizon—reveals complex nonlinear patterns and high variability in overspeed occurrences. These characteristics motivate the use of a machine learning-based classification method. A logistic regression model is selected due to its interpretability and suitability for online inference, enabling the predictor to adaptively detect and respond to overspeed cases with improved flexibility and accuracy.

4.3.2.2 Workflow

1) Future speed prediction The baseline speed update rule is modified to account for overspeed classification:

$$\begin{aligned}\hat{v}_L(\zeta \mid s) = \min\Big\{ & \hat{v}_{\max}(\zeta \mid s), \\ & \tilde{v}_L(s) + \int_{\zeta_L(s)}^{\zeta} \min\Big\{ \frac{a_{\text{comfort}}}{v_L(p \mid s)}, \frac{a_{\text{Lmax}}(s)}{v_L(p \mid s)} \Big\} \, \mathrm{d}p \Big\}, \\ & \forall\, \zeta \in \Big(\tilde{x}_L(s),\, s + H\Big]\end{aligned} \tag{4.33}$$

$$\hat{v}_{\max}(\zeta \mid s) = \begin{cases} v_{\text{limit}}(\zeta \mid s), & \text{if classified as overspeed } \hat{\delta} = 1 \\ v_{\text{set}}(\zeta \mid s), & \text{otherwise } \hat{\delta} = 0 \end{cases} \tag{4.34}$$

Where:

- $\hat{v}_{\max}(\zeta \mid s)$: future possible maximum speed (to be predicted);
- $v_{\text{set}}(\zeta \mid s)$: fixed cruise control speed;
- $v_{\text{limit}}(\zeta \mid s)$: legal road speed limit;
- $\hat{\delta}$: binary decision variable.

2) Online Classifier Training To identify cases of overspeed, a logistic regression classifier is used on historical data in a sliding window of size W. For each time s, the feature vector is:

$$\mathbf{x}(s,h) = \begin{bmatrix} v(s) \\ \alpha(s+h) \\ h \end{bmatrix} \quad \text{and} \quad \delta(s,h) = \begin{cases} 1, & v(s+h) > v_{\text{set}} \\ 0, & \text{otherwise.} \end{cases}$$

where $v(s)$ is the current speed, $\alpha(s+h)$ is the previewed slope h steps ahead. The predicted probability is:

$$\hat{p}(s,h) = \frac{1}{1+\exp(-\boldsymbol{\beta}^\top \mathbf{x}(s,h))}$$

where $\boldsymbol{\beta}$ is the learned parameter vector. If $\hat{p} > 0.5$, the predictor lifts the constraint from v_{set} to v_{limit}.

Model training minimizes the negative log-likelihood over available labeled samples.

This model will repeat the prediction over the horizon to get a vector of decision variables $\hat{\delta}(s+1), \ldots \hat{\delta}(s+H)$.

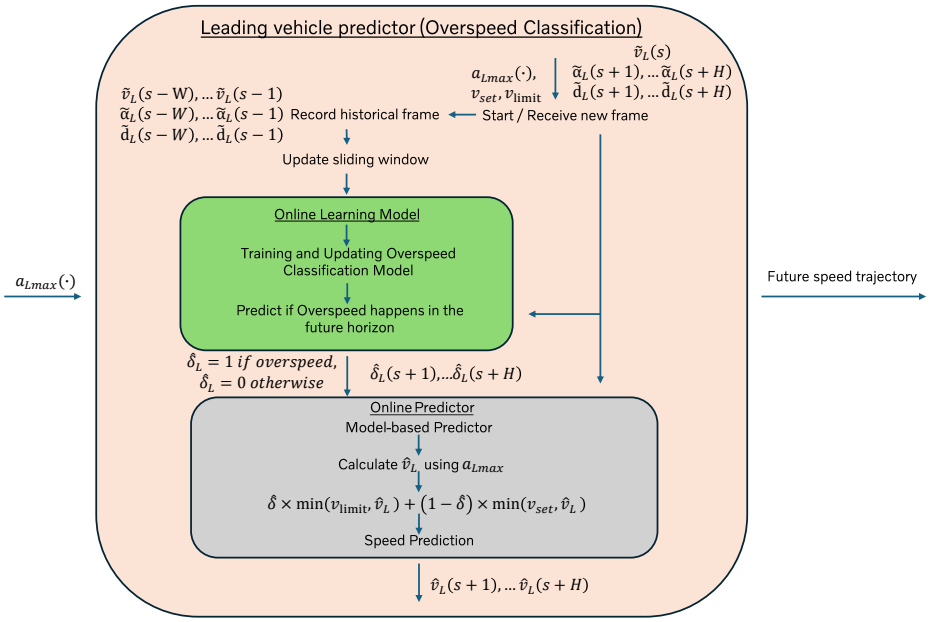

Figure 4.6: The overview of the Online Overspeed Classification Predictor Structure: real-time logistic regression on vehicle speed and road slope to dynamically switch between set cruise speed and legal speed limit for speed forecasting.

4.3.3 Integrated with Online Acceleration Regression

4.3.3.1 Motivation

The baseline approach assumes a fixed value for the comfort acceleration a_{comfort}, which is used to cap the predicted acceleration in the interest of ride comfort. However, in real-world driving, this parameter effectively reflects the driver's actual acceleration behavior, which can vary significantly with context, terrain, and driving habits. Using a constant value fails to capture such variation and often leads to either overly conservative or inaccurate predictions.

Analysis of logged driving data shows that the actual acceleration behavior is influenced by both the instantaneous vehicle speed and the upcoming road slope. However, the relationship between these variables and the applied acceleration is nonuniform and varies across situations. The distribution of observed acceleration responses suggests that fixed-rule models cannot capture this variability effectively.

Therefore, instead of treating a_{comfort} as a static constant, this method dynamically estimates it using a regression model trained on recent data. The model predicts the expected driver-induced acceleration based on past data, allowing the predictor to better adapt to context-specific and driver-specific behavior.

4.3.3.2 Workflow

1) Online Model Training To capture recent driving behavior, a model is trained using the historical data in a sliding window of size W.

- Record data: speed $v_{L(t-W:t)}$, slope $\alpha_{L(t-W:t)}$, and cumulative distance $d_{L(t-W:t)}$.
- Compute time intervals and observed accelerations:

$$\Delta t_i = \frac{D_{i+1} - D_i}{v_i}, \quad a_i = \frac{v_{i+1} - v_i}{\Delta t_i}, \quad i = t - W, \ldots, t - 1$$

- Fit a linear regression model with inputs and labels:

$$X_i = [v_i, \alpha_{i+1}], \quad y_i = a_i$$

$$\hat{a}_{comfort} \approx \beta_0 + \beta_1 v + \beta_2 \alpha$$

  Then the model will repeat the prediction over the horizon to get a vector of future comfort accelerations $\hat{a}_{comfort}(s+1)$, ... $\hat{a}_{comfort}(s+H)$.

- The model can be replaced or extended by more expressive learners (e.g., CNNs, RNNs) to enhance adaptability for a dataset with more patterns.

2) Parallel Speed Prediction

2.1) Learning-based prediction:

$$\hat{v}_{\mathrm{L(learning)}}(\zeta \mid s) = \min\Bigg\{ v_{\max}(\zeta \mid s), \\ \tilde{v}_L(s) + \int_{\zeta_L(s)}^{\zeta} \min\left\{ \frac{\hat{a}_{\mathrm{comfort}}(s)}{v_L(p \mid s)}, \frac{a_{\mathrm{Lmax}}(s)}{v_L(p \mid s)} \right\} \mathrm{d}p \Bigg\} \tag{4.35}$$

Where:

- $\hat{a}_{comfort}(s)$: future possible comfort acceleration at position s (to be predicted);
- $v_{\mathrm{limit}}(\zeta \mid s)$: legal road speed limit.

2.2) Model-based prediction (same as 4.3.1):

$$\hat{v}_{L(Model)}(\zeta \mid s) = \min\Bigg\{ v_{\max}(\zeta \mid s), \\ \tilde{v}_L(s) + \int_{\zeta_L(s)}^{\zeta} \min\left\{ \frac{a_{\mathrm{comfort}}}{v_L(p \mid s)}, \frac{a_{\mathrm{Lmax}}(s)}{v_L(p \mid s)} \right\} \mathrm{d}p \Bigg\}, \\ \forall\, \zeta \in \Big(\tilde{x}_L(s),\, s + H\Big] \tag{4.36}$$

When the model-predicted speed is less than or equal to the set cruise speed, the model is trusted; otherwise, the learning-based prediction is used.

2.3) Final fused prediction:

$$\hat{v}_L(\zeta \mid s) = \begin{cases} \hat{v}_{\mathrm{L(learning)}}, & \text{otherwise} \\ \hat{v}_{L(Model)}, & \text{if } \hat{v}_{L(Model)} \leq v_{\mathrm{set}}(\zeta) \end{cases} \tag{4.37}$$

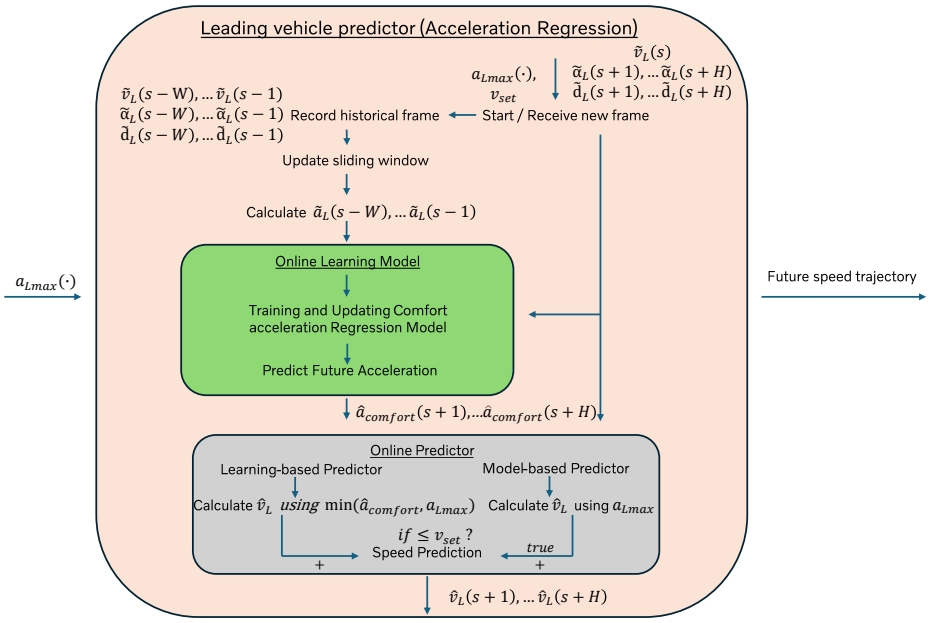

Figure 4.7: The overview of the Online Acceleration Regression Predictor Structure: linear regression of comfort acceleration from historical data combined with maximum engine acceleration and speed-limit constraints to produce adaptive speed predictions

5

Results

This chapter presents the evaluation of the proposed learning-based observer–predictor structure through simulation and real-world driving data. The performance of the observer and predictor components is assessed separately to provide a clear understanding of their contributions. Various anomaly removal techniques and observer configurations are tested to identify the most accurate estimation strategy for the leading vehicle's parameters. The predictor's accuracy is then benchmarked against conventional baselines and two proposed learning-based enhancements under representative driving cycles. Overall, the results demonstrate the structure's accuracy and effectiveness in various scenarios.

5.1 Evaluation metric and setup

The evaluation will be conducted in two stages, as the estimation of the leading vehicle's future speed trajectory is carried out using a learning-based observer and predictor architecture. The performance of both components will be evaluated separately.

The performance of different anomaly removal techniques is evaluated, along with an assessment of the observer's accuracy by comparing the estimated coefficient $\mathbf{b}$ with its true value. Then, the performance of the predictor is evaluated by comparing the predicted speed trajectory with the actual speed trajectory.

Two types of data sources will be used for evaluation. The first data source comes from Volvo's internal simulation platform, where parameters such as vehicle mass and driving cycles can be specified. The second source consists of real driving cycle logs recorded during field tests conducted by Volvo.

When evaluating the observer, since the dynamics of the leading vehicle are unknown in the real driving cycle data logs, the evaluation will be performed using the ego vehicle. In this case, the ego vehicle's speed is treated as the leading vehicle's speed, and the estimated results are compared with the known parameters of the ego vehicle.

For the evaluation of the predictor, due to the assumption of a simple speed dynamics profile for the leading vehicle in Volvo's current simulation platform, two simulations of the ego vehicle with different vehicle masses will be used. One will act as the leading vehicle and the other as the ego vehicle.

Compared to the fast-changing conditions and unstable detections in real-world scenarios, the simulated data logs provide more stable and continuous signals, making them suitable for the final demonstration of the predicted trajectory.

5.2 Anomaly removal techniques evaluation

Anomaly removal techniques play an important role in processing data after applying a Wiener filter to the speed measurements. The performance of the anomaly removal method used directly affects the quality of the data fed into the optimization problem, and therefore influences the final estimation of the leading vehicle's maximum acceleration capability. As introduced in Section 2.5, both parametric and non-parametric statistical methods, as well as GPR, can be employed for anomaly removal. An iterative GPR approach has been explored; however, the algorithm requires a large amount of time, and testing has confirmed that it is not suitable for real-time applications. Therefore, Gaussian Process Regression will not be considered in the subsequent evaluation steps.

In this thesis, three methods are explored: Isolation Forest, KNN clustering, and recursive polynomial fitting based on the Z-score approach. The evaluation is conducted using logs generated by Volvo's internal simulation platform, along with real logs recorded during field tests.

Test technique configuration All three anomaly removal techniques are implemented in MATLAB R2021a and applied to the same pre-processed input, i.e., paired speed measurements and corresponding calculated force-to-mass ratios $\left(\tilde{v}_L, \tilde{f}_{Lf}\right)$, which have already been denoised using a Wiener filter and limited to the physically meaningful speed range defined in Section 4.2.4.1.

- Isolation Forest (IF): The feature vector $\left[\tilde{v}_L, \tilde{f}_{Lf}\right]^\top$ was used to train an isolation forest model with a contamination fraction of 0.25, meaning that at most 25% of the samples could be flagged as anomalies.
- KNN Clustering (KNN): A pairwise distance matrix was constructed for the force-to-mass ratio only. For each point, the average distance to its $k = 5$ nearest neighbors was computed. A sample was declared an anomaly if the average exceeded μ.
- Recursive polynomial fitting (Recursive Polyfit): A fourth-order polynomial was iteratively fitted to the valid $\left(\tilde{v}_L, \tilde{f}_{Lf}\right)$ pairs. After each fit, the residuals were screened with a two-sided Z-score threshold of $\pm 2\sigma$. Detected outliers were masked, and the procedure was repeated for up to three iterations or until no new outliers appeared.

The above hyperparameters were selected based on multiple preliminary experiments to ensure strong performance within each method's respective category. Unless otherwise noted, these parameters are kept constant throughout all experiments. Anomalies identified by each method are removed from the dataset before solving the optimization problem to generate the most accurate estimation of the leading

vehicle's maximum acceleration capability. The results are shown in Table B.1.

Table 5.1: Calculated vector **b** with different anomaly–removal techniques under various driving-cycle logs (Refer to the appendix Table A.1 for a record of all results).

Driving cycle	IF	KNN	Recursive Polyfit	$\boldsymbol{b}_{\text{true}}$
Simulated Logs				
Log 1 ($85t$)	$\begin{bmatrix} 3.86 \\ 5.68 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 5.51 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 4.23 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 4.39 \\ 3.62 \times 10^{-5} \end{bmatrix}$
Log 4 ($55t$)	$\begin{bmatrix} 5.82 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 6.56 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 7.22 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 6.78 \\ 5.6 \times 10^{-5} \end{bmatrix}$
Real-world Logged Data				
Log 10 (EU)	$\begin{bmatrix} 7.73 \\ 5.24 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 8.90 \\ 5.24 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 8.67 \\ 5.24 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 8.58 \\ 7.44 \times 10^{-5} \end{bmatrix}$
Log 12 (BRA)	$\begin{bmatrix} 6.20 \\ 5.19 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 13.99 \\ 5.19 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 6.97 \\ 5.19 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 5.17 \\ 6.08 \times 10^{-5} \end{bmatrix}$

Method performance evaluation To quantify how closely each anomaly removal technique reproduces the ground truth vector $\mathbf{b}_{\text{true}}$, two standard distance-based metrics are computed over all 13 driving cycle logs:

- **MAE** (Mean Absolute Error): $\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{b}_i - \mathbf{b}_{i,\text{true}}\|_2$
- **RMSE** (Root Mean Square Error): $\sqrt{\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{b}_i - \mathbf{b}_{i,\text{true}}\|_2^2}$

where $\|\cdot\|_2$ is the Euclidean norm, and $N = 13$. Because the second element (air drag) of each vector is roughly 10^5 times smaller than the first element, it is normalized so that both elements contribute comparably to the final metric. The calculated evaluation metrics are shown in Table 5.2.

Table 5.2: Accuracy of anomaly removal techniques after scaling the second element of **b** by 10^5 (lower is better).

Method	MAE	RMSE
Isolation Forest (IF)	3.55	4.61
k-Nearest Neighbors (KNN)	4.98	6.05
Recursive Polyfit (RP)	**3.25**	**4.29**

Recursive Polyfit consistently achieves the lowest MAE and RMSE, indicating that it removes anomalous measurement points most successfully and is more effective than the other two alternatives. Isolation forest ranks second, while KNN is the least accurate. Consequently, **Recursive Polynomial Fitting** is selected as the anomaly removal technique for further analysis, offering the best overall accuracy and robustness across both simulated and real-world driving cycles.

5.3 Observer Accuracy

The learning-based observer aims to accurately estimate the acceleration capability of the leading vehicle, which will be further used by the predictor to forecast the final speed trajectory. The acceleration capability can be effectively represented by the coefficient vector **b**, which contains all the unknown parameters such as the leading vehicle's mass, engine maximum power, etc. Therefore, the observer is evaluated by comparing its estimate of **b** with the corresponding ground truth vector.

To benchmark the proposed observer, three alternative estimation strategies are considered:

- **NN without b**: As described in the observer design, the feature vector used by the neural network consists of four elements, one of which is the coefficient **b** obtained from the preliminary estimation by the polynomial fitting filter. In this method, the same neural network is used but without including the coefficient **b** in the feature vector. This serves as one of the comparison baselines.
- **Polynomial fitting**: This method solely uses polynomial fitting to estimate the coefficient **b**, without incorporating any machine learning techniques. It relies purely on statistical characteristics to perform the prediction.
- **Polyfit + NN Filter**: As described in the observer design, this method combines the polynomial fitting filter and the neural network filter. This is the method used in the final system, and its output is fed into the leading vehicle predictor.

It is worth mentioning that all the methods listed above filter the data pairs and select the target sample points. All the final selected "speed-force-to-mass" ratio pairs are then fed into the optimization problem, which outputs the final coefficient vector **b**.

Test technique configuration All three evaluation methods are implemented in MATLAB R2021a and applied to the same pre-processed input. Both neural networks require a pretrained model, and the training details of these models are provided in Section 4.2.3. The combination of Polyfit and neural network logic follows the procedure used in observer design. The evaluation results are shown in Table B.2.

Table 5.3: Observer accuracies with different estimation methods under various driving cycle logs (Refer to the appendix Table A.2 for a record of all results).

Driving cycle	**NN without b**	**Polyfit**	**Polyfit + NN**	$\boldsymbol{b}_{\text{true}}$
Simulated Logs				
Log 1 ($75t$)	$\begin{bmatrix} 6.06 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 4.53 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 5.18 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 4.97 \\ 4.11 \times 10^{-5} \end{bmatrix}$
Log 4 ($35t$)	$\begin{bmatrix} 10.95 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 10.45 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 10.95 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 10.66 \\ 8.81 \times 10^{-5} \end{bmatrix}$
Real-world Logged Data				
Log 5	$\begin{bmatrix} 7.83 \\ 5.74 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 16.91 \\ 5.74 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 12.70 \\ 3.45 \times 10^{-4} \end{bmatrix}$	$\begin{bmatrix} 12.98 \\ 1.74 \times 10^{-4} \end{bmatrix}$
Log 6	$\begin{bmatrix} 5.77 \\ 5.24 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 10.64 \\ 5.24 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 8.41 \\ 5.24 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 8.58 \\ 7.43 \times 10^{-5} \end{bmatrix}$

Similar to the anomaly removal technique evaluations, the output of different approaches is assessed using **MAE** and **RMSE**. A total of $N = 12$ logs are used in the experiments, and the second element of each vector is normalized so that both elements contribute comparably to the final metric. The calculated evaluation metrics are shown in the Table 5.4.

Table 5.4: Observer accuracy comparison after scaling the second element of **b** by 10^5 (lower is better).

Method	**MAE**	**RMSE**
NN without **b**	12.37	15.90
Polynomial fitting	7.09	12.79
Polyfit + NN	**6.92**	**11.13**

After normalizing the scale of both elements, the combination of polynomial fitting and neural network yields the lowest MAE and RMSE, outperforming both standalone polynomial fitting and the neural network variant. The neural network without the feature **b** performs the worst, as expected. However, the polynomial fitting approach still delivers solid performance and does not lag significantly behind. The improvement in the combined approach comes from the neural network's ability to identify data pairs with the highest probability of operating at maximum power, which are then directly fed into the optimization step. For this reason, the **Polyfit + NN** approach is selected as the final structure of the learning-based observer.

The results also show that, when more real-world driving logs are included, the error increases compared to the values in Table 5.2, which primarily uses simulation logs. Therefore, in real-world implementation, noise still prevents the algorithm from accurately estimating the vector **b**.

5.4 Predictor Performance

To comprehensively evaluate the effectiveness of the proposed predictor workflows, the following baseline methods and proposed enhancements are considered for comparison (All predictor inputs used are the outputs of the most performing observer):

Benchmark Approaches

- **Benchmark 1 (Constant Speed Predictor)**: Assumes the leading vehicle maintains a constant speed equal to its current value throughout the prediction horizon. This method does not involve any observer or model.
- **Benchmark 2 (Maximum Acceleration Predictor 4.3.1)**: Utilizes a physics-based observer that assumes the leading vehicle always accelerates at its maximum engine capacity, constrained by road slope and comfort bounds.

Proposed Approaches

- **Proposed 1 (Overspeed Classification Predictor 4.3.2)**: Enhances the baseline model by identifying downhill overspeed events using a logistic classifier. When overspeed is predicted, the model lifts the fixed speed cap v_{set} and allows prediction up to the road speed limit to improve accuracy.
- **Proposed 2 (Acceleration Regression Predictor 4.3.3)**: Trains an online linear regression model using recent vehicle dynamics data to estimate acceleration under varying conditions. The final speed prediction is chosen from either the learning-based or model-based result, depending on the scenario.

The aforementioned baselines and proposed methods are evaluated on two representative driving cycles, DC-1 and DC-2, under identical total travel distance constraints. The comparative performance across these scenarios is illustrated in Figure 5.3 and Figure 5.6. These baselines represent widely adopted strategies in the eco-driving domain and provide a meaningful benchmark for assessing the effectiveness of the proposed learning-based predictors.

Evaluation Metrics for Predictor Performance

The following metrics and analysis strategies have been adopted to quantify accuracy and performance gains:

- **Prediction Accuracy:**
  - *Root-Mean-Square Error (RMSE)*: For N trajectories each sampled at K points over a prediction horizon,

$$\text{RMSE} = \sqrt{\frac{1}{N\,K}\sum_{i=1}^{N}\sum_{j=1}^{K}\left(v_{ij}^{\text{pred}} - v_{ij}^{\text{true}}\right)^2},$$

– *Mean-Absolute Error (MAE)*:

$$\mathrm{MAE} = \frac{1}{N\,K}\sum_{i=1}^{N}\sum_{j=1}^{K}\left|v_{ij}^{\mathrm{pred}} - v_{ij}^{\mathrm{true}}\right|,$$

providing a robust average deviation measure.

– *Mean-Absolute Percentage Error (MAPE)*:

$$\mathrm{MAPE} = \frac{100\%}{N\,K}\sum_{i=1}^{N}\sum_{j=1}^{K}\left|\frac{v_{ij}^{\mathrm{pred}} - v_{ij}^{\mathrm{true}}}{v_{ij}^{\mathrm{true}}}\right|,$$

normalizing errors by true speed for fair comparison across varying speed levels.

- **Improvement Rate:** Improvement relative to the constant speed predictor (benchmark 1) is defined as

$$\mathrm{Improve}_{\mathrm{RMSE}} = \frac{\mathrm{RMSE}_1 - \mathrm{RMSE}_m}{\mathrm{RMSE}_1} \times 100\%,$$

with analogous expressions for MAE and MAPE. A positive value indicates error reduction.

- **Full-Cycle vs. Segment Comparison:**
  – *Full-Cycle Evaluation:* Metrics are computed over the entire set of prediction instances (all N trajectories with K samples each across the prediction horizon). This provides a comprehensive measure of average accuracy and overall error reduction.
  – *Segment Evaluation:* A critical sub-range of interest is isolated, and only those trajectories whose start points lie within this segment (N_{seg} instances) are evaluated. Recomputing RMSE, MAE, and MAPE on this subset reveals method robustness under the most challenging road conditions (e.g. sustained grade changes include both uphill and downhill cases).

The outcomes of both full-cycle and segment evaluations are illustrated in Figures 5.4, 5.5, 5.7, 5.8 and detailed in Tables 5.5, 5.6, 5.7, 5.8.

Overview of Driving cycles Two representative driving cycles are selected for validation and comparative analysis:

- **DC-1**: A relatively smooth terrain of approximately 90 km in length. The road includes mild uphill and downhill sections with low slope angles, and the elevation transitions are gradual. The cycle features moderate topographical variation with infrequent slope switching, making it suitable for general eco-driving evaluation.
- **DC-2**: A more challenging route of approximately 65 km in length. This cycle contains steeper and more frequent changes in road slope, with rapid

transitions between ascent and descent. It introduces stronger nonlinearity in driving conditions and poses higher demands on prediction robustness and model adaptability.

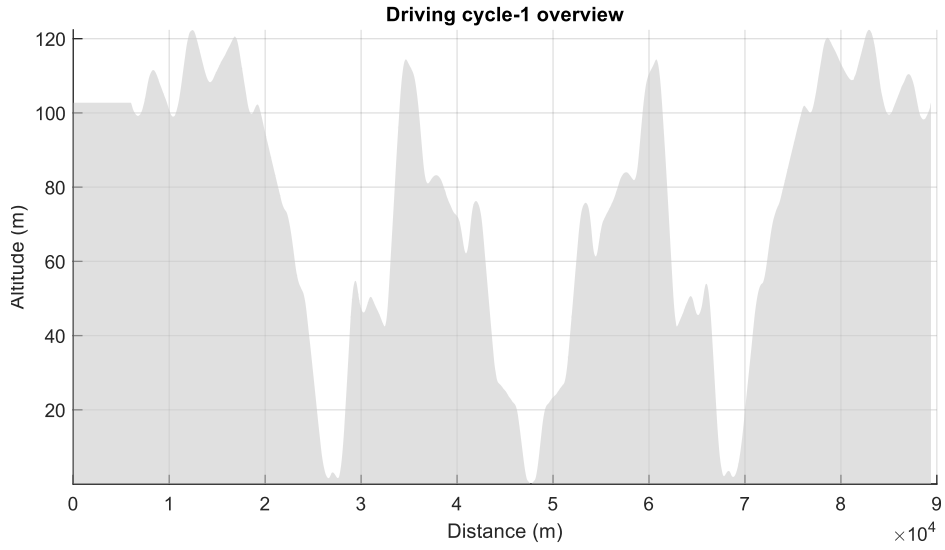

Figure 5.1: The predicted speed of the leading HDV overview in DC-1.

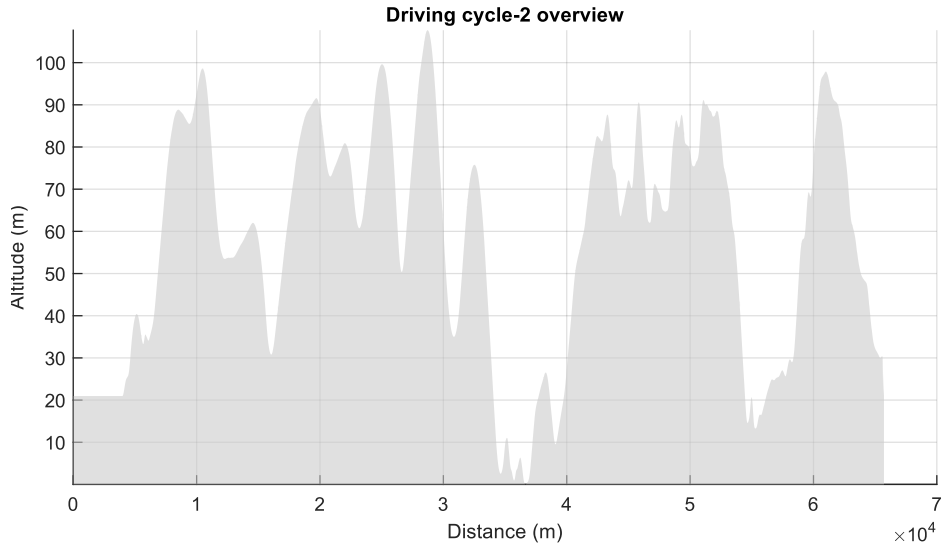

Figure 5.2: The predicted speed of the leading HDV overview in DC-2.

To enable detailed analysis, a 10 km segment with the most complex and representative topographic variation is selected from each driving cycle. These segments present the most challenging prediction conditions, making them suitable for highlighting the differences among the tested methods.

For DC-1, the interval from 58 km to 68 km is chosen, as it includes both extended uphill and downhill transitions with mild curvature.

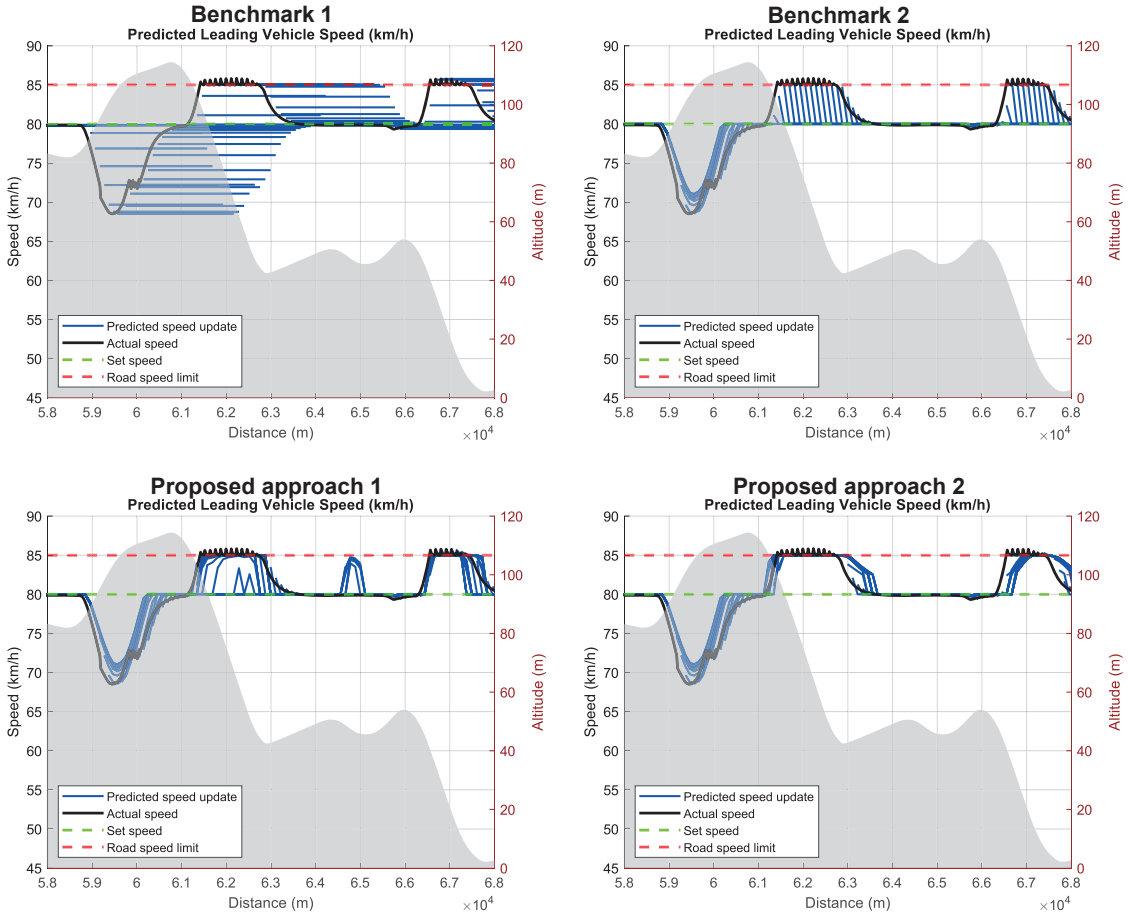

Figure 5.3: Predicted and actual speed trajectory of the leading HDV over DC-1 in case 1-4

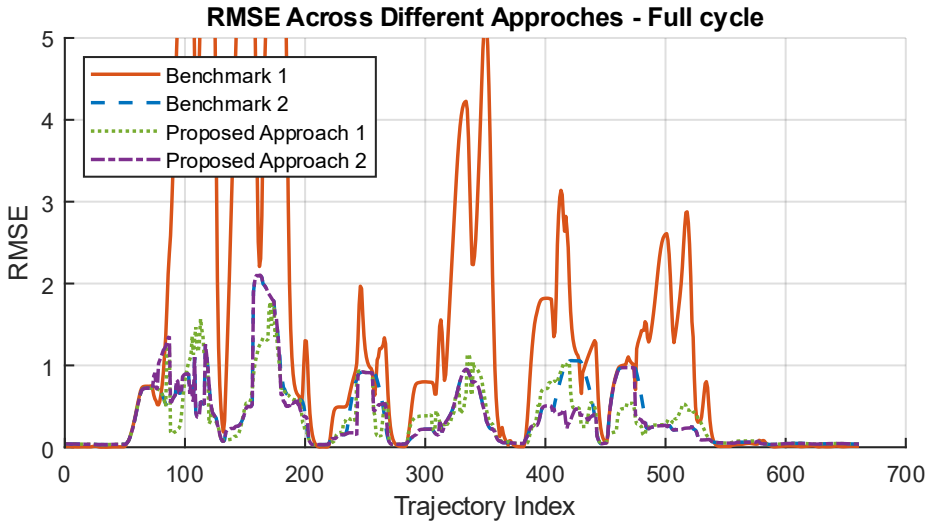

Figure 5.4: RMSE of four approaches' prediction over the full cycle in DC-1.

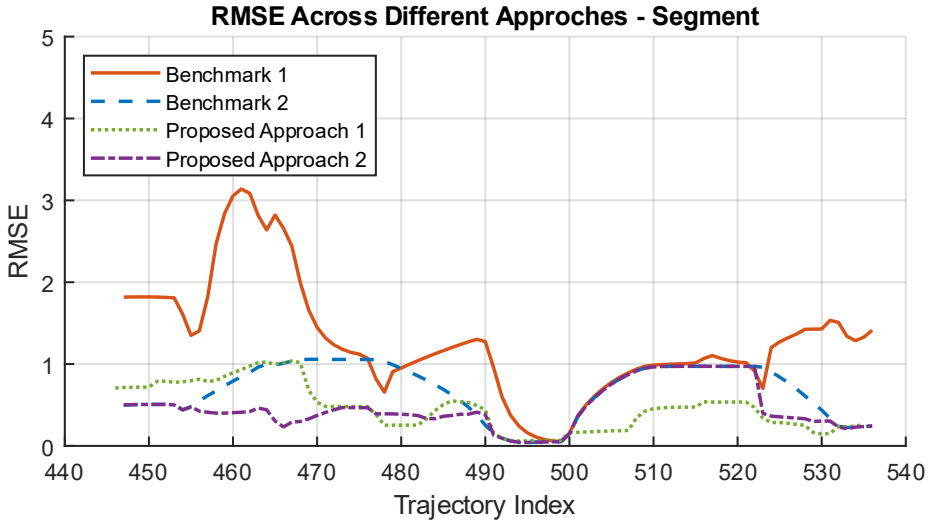

Figure 5.5: RMSE of four approaches' prediction over the segment in DC-1.

Method	RMSE	Improve (%)	MAE	Improve (%)	MAPE	Improve (%)
Benchmark 1	2.6228	–	1.2684	–	7.0472	–
Benchmark 2	0.5722	78.18	0.2933	76.88	1.5705	77.72
Proposed 1	0.5062	80.70	0.2609	79.43	1.3864	80.33
Proposed 2	0.5423	79.32	0.2721	78.55	1.4876	78.89

Table 5.5: Global errors and improvement rates in DC-1.

Method	RMSE	Improve (%)	MAE	Improve (%)	MAPE	Improve (%)
Benchmark 1	1.4558	–	1.0639	–	4.7830	–
Benchmark 2	0.7565	48.03	0.4940	53.57	2.1477	55.10
Proposed 1	0.5404	62.88	0.3160	70.29	1.4270	70.17
Proposed 2	0.5392	62.96	0.3064	71.20	1.3601	71.57

Table 5.6: Segment errors and improvement rates in DC-1.

For DC-2, the segment from 45 km to 55 km is used, featuring sharp slope changes, frequent undulations, and abrupt transitions, posing significant challenges for trajectory prediction.

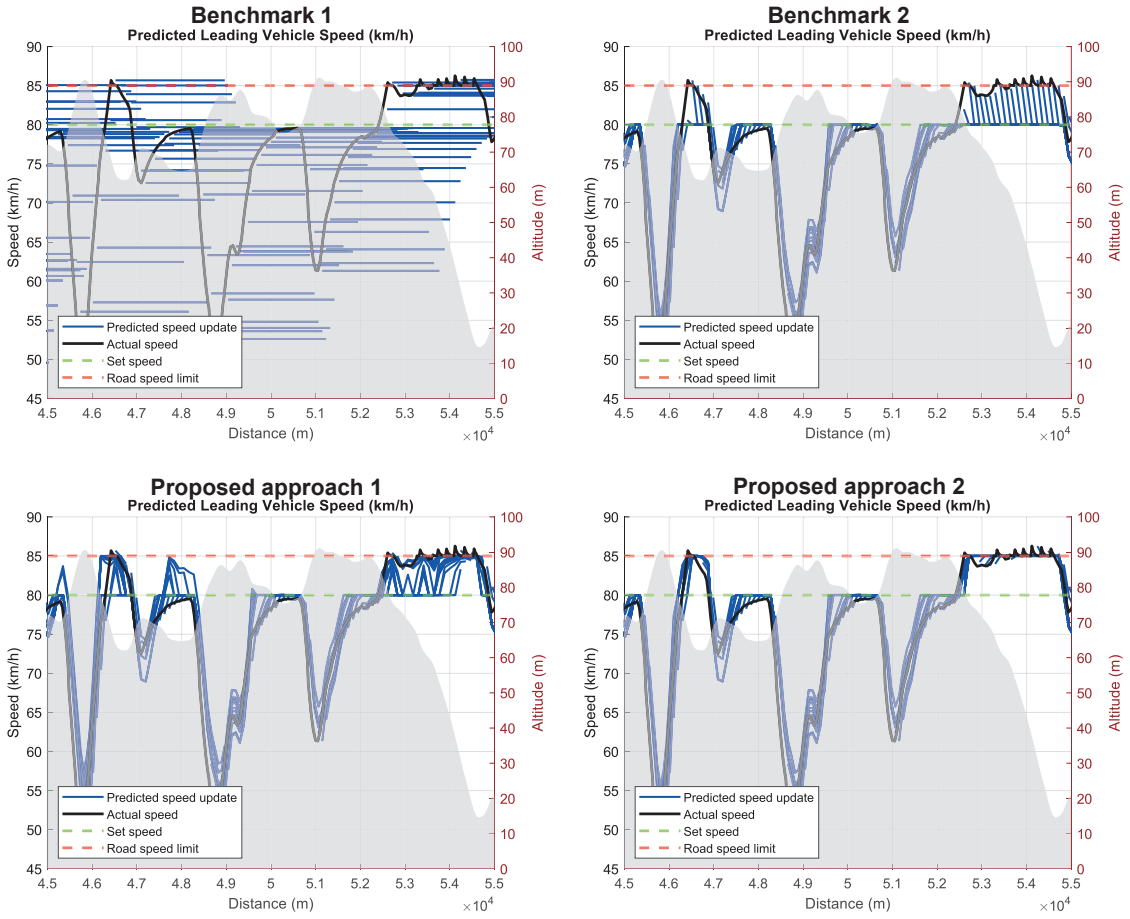

Figure 5.6: Predicted and actual speed trajectory of the leading HDV over DC-2 in case 1-4.

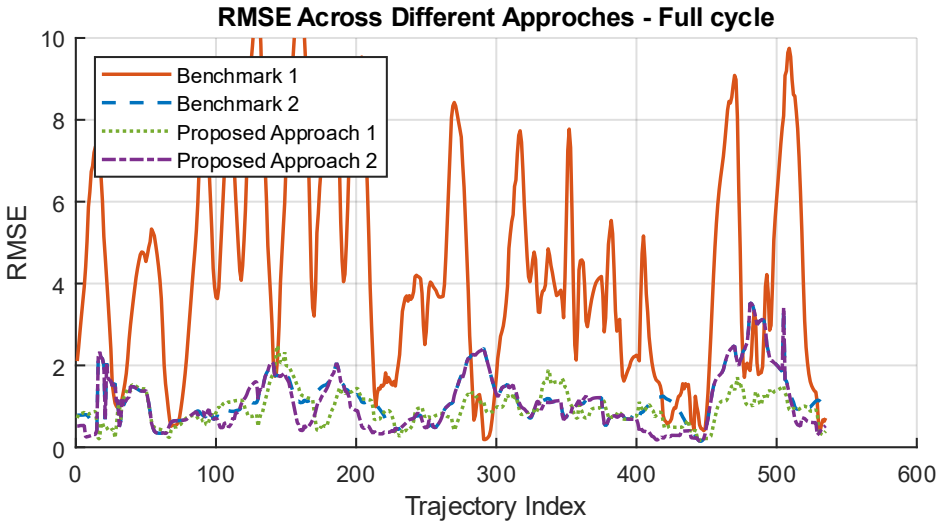

Figure 5.7: RMSE of four approaches' prediction over the full cycle in DC-2.

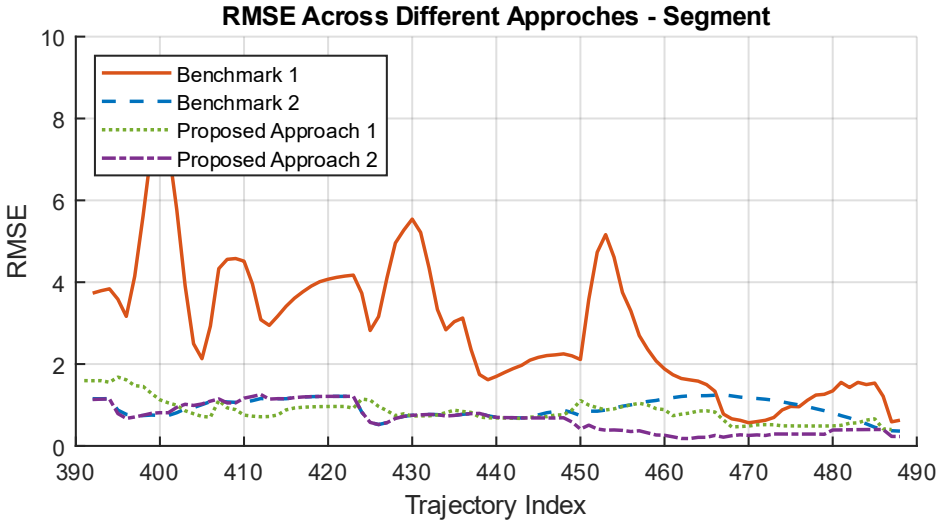

Figure 5.8: RMSE of four approaches' prediction over the segment in DC-2.

Method	RMSE	Improve (%)	MAE	Improve (%)	MAPE	Improve (%)
Benchmark 1	5.1276	–	3.6757	–	22.7680	–
Benchmark 2	1.3439	73.79	1.0365	71.80	7.0477	69.05
Proposed 1	1.0264	79.98	0.7863	78.61	4.9450	78.28
Proposed 2	1.2750	75.13	0.9027	75.44	6.5288	71.33

Table 5.7: Global errors and improvement rates in DC-2.

Method	RMSE	Improve (%)	MAE	Improve (%)	MAPE	Improve (%)
Benchmark 1	3.2990	–	2.4367	–	12.3510	–
Benchmark 2	0.9485	71.25	0.7759	68.16	3.7985	69.25
Proposed 1	0.8919	72.97	0.6721	72.42	3.3604	72.79
Proposed 2	0.7433	77.47	0.5358	78.01	2.7888	77.42

Table 5.8: Segment errors and improvement rates in DC-2.

6

Discussion

This chapter discusses key insights, limitations, and practical considerations related to the proposed observer–predictor design. It examines the convergence behavior of the observer under different terrain conditions and analyzes the effectiveness of the predictors across varying driving scenarios. Real-world challenges such as object ID switching, sensor noise, and multimodal traffic behaviors are also explored to highlight potential failure cases. The discussion aims to provide a balanced assessment of the system's robustness, adaptability, and applicability in realistic deployment settings.

6.1 Observer Convergence

As illustrated in Section 2.1, the physical formulation of this problem is based on the assumption that the leading vehicle operates at maximum power when going uphill. Similar to what polynomial fitting and neural networks attempt to do, all methods aim to determine when the leading vehicle is operating at maximum power. Consequently, the corresponding "speed-force-to-mass" ratio pairs are applied to calculate the vehicle's maximum acceleration capability and, ultimately, to predict the future speed profile.

This logic also raises a concern, which is evident in some of the simulation logs and real driving cycle logs. If the route the heavy-duty vehicle travels on is too flat or lacks significant fluctuations, the HDV rarely operates at maximum power. In such cases, the designed observer-predictor structure struggles to estimate the maximum power profile. Moreover, for the observer to make a meaningful initial prediction, the leading vehicle must encounter at least one uphill segment. This characteristic can also be seen in Figure 6.1. When there are several uphill sections in the driving cycle, the estimation converges after the vehicle has passed at least one uphill section. In conclusion, whether the observer can converge depends on the presence of significant altitude changes in the driving cycle and the timing of the first uphill segment.

That said, once the uphill driving measurements are fed into the observer, it can converge relatively quickly, aided by the polynomial fitting filter and the neural network-based identification. After convergence, the subsequently estimated vector $\mathbf{b}$ may slightly deteriorate in performance compared to earlier estimations. This is common, as newly introduced measurements can introduce uncertainties, poten-

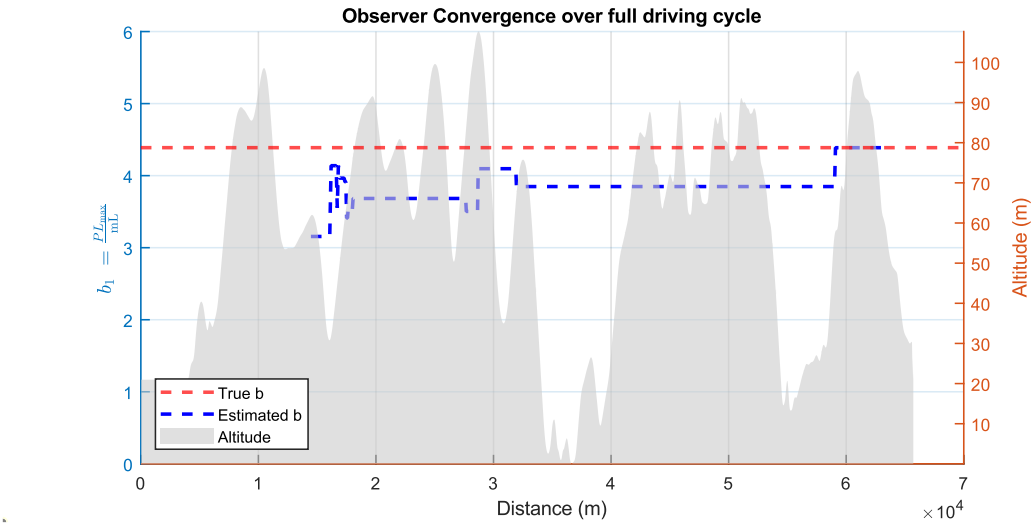

Figure 6.1: Evaluation of observer convergence.

tially reducing the effectiveness of anomaly removal filters. In summary, once the necessary measurements are obtained, the learning-based observer converges quickly after encountering an uphill, though some estimation deviation may occur due to the influence of new data.

6.2 Speed Prediction Performance

In both DC-1 and DC-2, the proposed learning-based predictors show clear improvements over the baseline methods in terms of prediction accuracy and adaptability.

For DC-1, which features smoother terrain and relatively moderate slope changes, the baseline methods perform reasonably but still exhibit limitations. As shown in Figure 5.3, Benchmark 1 (constant speed assumption) fails to reflect the vehicle's actual speed variation in uphill and downhill segments, especially between 62 km and 64 km. Benchmark 2, which uses the maximum engine acceleration, improves prediction during uphill phases but tends to underestimate the speed during downhill sections because it does not account for coasting or momentum. In contrast, Proposed approach 1 improves the prediction during these downhill parts by identifying overspeed cases and lifting the artificial speed cap. Proposed approach 2 further improves prediction by learning acceleration patterns from recent data, resulting in more accurate speed estimates over varying slopes.

For DC-2, where slope changes are more frequent and sharper, the baseline predictors show significant mismatches. Benchmark 1 often overestimates speed during uphill segments and underestimates during downhill runs. Benchmark 2 provides more reasonable predictions in uphill cases but remains overly conservative during descents. The proposed methods adapt better to these complex conditions. Proposed approach 1 correctly adjusts speed limits when overspeed is likely, improving downhill predictions. Proposed approach 2 learns the vehicle's recent behavior in real time and provides more consistent and accurate speed prediction throughout the segment, even under rapid slope changes.

In summary, the proposed methods outperform the baselines by reducing prediction errors in both simple and complex driving environments. Proposed approach 1 is especially useful for handling short-term downhill overspeed, while Proposed approach 2 provides overall better adaptability by capturing changing driving patterns.

The numerical results in Tables 5.5 - 5.8 confirm that both proposed learning-based predictors deliver substantial accuracy gains over the two baseline methods in DC-1 and DC-2.

Overall, Proposed 1 excels at correcting short-term downhill overspeed, and Proposed 2 demonstrates superior adaptability across both moderate and steep terrain. The combined full-cycle and segment evaluations provide a comprehensive validation of prediction accuracy and method robustness.

6.3 ID Switching Cases

In real-world driving scenarios, the leading vehicle is unlikely to remain the same throughout the entire driving cycle, unlike in a simulation environment. Therefore, the designed leading vehicle speed prediction function is integrated with the Adaptive Cruise Control (ACC) system to better adapt to real-world conditions. The prediction function itself is not responsible for selecting the leading vehicle; this task is handled by the ACC system. This integration allows the prediction function to work alongside the ACC system, enhancing safety in practical applications.

If the leading vehicle's predicted speed is inaccurate, resulting in a dangerously short following distance or other undesirable behavior, the ACC system will proactively intervene by braking the ego vehicle to prevent a collision. Thus, the safety boundary is primarily maintained by the ACC system, while the prediction function focuses more on driving efficiency and energy management.

As described, the ACC system provides the prediction function with the leading vehicle's ID. Consequently, each speed measurement is associated with this object ID. This allows for accurate estimation of different leading vehicles' properties and enables the reuse of previously stored estimation results if a vehicle reappears as the leading vehicle after temporarily disappearing. Additionally, this stored information can be uploaded to the cloud for future analysis, potentially accompanied by the captured images from the camera. This data could serve as commercially valuable information, as it may include estimates of maximum engine power and aerodynamic characteristics of other trucks.

6.4 Failing Situations

Even though the proposed method demonstrates solid performance and relatively accurate estimation of the leading vehicle's future speed trajectory, there are still certain scenarios where the method may fail and be unable to generate accurate and reasonable estimations.

Noisy measurements The leading vehicle's speed measurements are assumed to follow a Gaussian distribution, based on the analysis of Volvo's internal data. However, if the sensor does not function as expected or unforeseen circumstances occur, the measurement noise may become significantly non-Gaussian. In such cases, the anomaly removal module may fail to handle the noise effectively, leading to highly inaccurate estimates of the maximum acceleration capability. These inaccurate estimates cannot be reliably used in the predictor.

No maximum power measurements As discussed in Section 6.1, the physical formulation of the leading vehicle's dynamics relies on the assumption that the vehicle operates at maximum power. If, for any reason, the leading vehicle does not operate at maximum power, the designed architecture will be unable to capture the desired properties. Consequently, prediction will fail under these circumstances.

Lack of continuous detection In real-world driving scenarios, the leading vehicle may change frequently due to other vehicles cutting in or the ego vehicle overtaking others. In such situations, the system can't collect continuous measurements from the same leading vehicle. The object ID may switch before enough data is gathered to make a reliable prediction. Even if a short sequence of measurements is successfully collected, it may not include any uphill segments, meaning the vehicle never operates at maximum power. The need for consistent measurements from the same leading vehicle, along with the requirement for an uphill scenario, makes it difficult to validate the current approach using real-world driving logs. This is also why the predictor's performance is primarily evaluated using simulation logs, which provide stable measurements and consistent leading vehicle behavior.

Multimodal driving scenarios in practice The predictors in this study leverage only speed and gradient information and do not incorporate real-world factors such as driver intent, road signage, traffic signals, weather conditions, or interactions with other vehicles. In practice, these elements introduce multimodal behavior—sudden braking, acceleration for green signals, lane changes or congestion—which can degrade prediction accuracy. Consequently, on datasets capturing complex, heterogeneous driving scenarios, performance gains may be substantially lower than those reported here.

7

Conclusion & Future Work

This chapter summarizes the main contributions and findings of the thesis, highlighting the effectiveness of the proposed learning-based observer–predictor structure for future speed estimation of non-communicating leading vehicles. It reflects on the system's performance in both simulated and real-world scenarios, and discusses how its components contribute to improved accuracy and robustness. In addition, the chapter outlines several directions for future work, including integration with onboard controllers, real-world field testing, and the development of more advanced learning models to enhance adaptability and performance in practical applications.

7.1 Conclusion

In this thesis, a learning-based observer-predictor structure was developed to estimate the future speed trajectory of a leading vehicle in non-communicative highway scenarios. The system is designed to support predictive energy management and eco-driving strategies, particularly in hilly uphill areas where engine limitations and elevation changes lead to significant speed variations of the leading vehicle, and also in downhill areas where potential overspeed may occur.

The proposed observer leverages a hybrid approach that combines recursive polynomial fitting with a neural network-based filter to estimate the leading vehicle's maximum acceleration capability from noisy speed measurements and road slope data. This estimate is then utilized by the predictor module, which forecasts the future speed profile of the leading vehicle. Two variants of the predictor were proposed—one based on overspeed classification using logistic regression, and the other based on online linear acceleration regression.

The results from both simulated and real-world driving logs show that the observer achieves a high degree of accuracy in estimating the physical capabilities of the leading vehicle. Among the anomaly removal methods considered, recursive polynomial fitting delivered the most consistent performance, and the combined observer structure (Polyfit + NN) provided the most robust estimation of the force-to-mass vector.

In terms of speed prediction, both proposed predictor variants outperform baseline methods in multiple scenarios. The overspeed classification predictor (Proposed 1) excels in identifying and lifting unnecessary speed caps during downhill segments, while the regression-based predictor (Proposed 2) provides better adaptability in

rapidly changing driving environments by learning short-term behavior. Overall, the integration of these components results in a more reliable and accurate speed forecast. This improved output can be used in the eco-driving controller or other ADAS functions, enabling energy-efficient and safer following behavior for the ego HDV.

In conclusion, the learning-based observer–predictor structure developed in this thesis offers a promising and practical solution for predictive energy management in HDVs, especially in the absence of vehicle-to-vehicle communication. The framework successfully demonstrates how machine learning techniques can be combined with physical models to overcome challenges posed by uncertainty, limited power, and noisy real-world signals.

7.2 Future Work

7.2.1 Integration with Volvo's Controller

Currently, the function operates independently of Volvo's predictive energy management system. From data input to output usage, the thesis code runs in isolation. The evaluation metric is primarily based on the accuracy of predicting the leading vehicle's future speed. However, by integrating this function with Volvo's predictive energy management controller, the ego vehicle model could utilize the prediction data to develop a more efficient eco-driving strategy. With this integration, the fuel consumption of the ego vehicle can be monitored, and tests can be conducted to evaluate whether the activated function contributes to fuel savings and improved eco-driving performance.

7.2.2 In-Vehicle Test

Due to time and resource constraints, the outcome of this thesis has not been tested on a real Volvo truck as an onboard function. All validations have been performed using either simulation platforms or pre-recorded driving cycle logs. Therefore, once the function is integrated with the current onboard systems, field tests are recommended to evaluate the real-world performance of the proposed learning-based observer-predictor structure.

7.2.3 Further development on learning model

Future developments will explore the integration of richer multimodal inputs (e.g., traffic flow, signal timing, meteorological data) and the design of hierarchical or hybrid architectures that first classify driving context before applying specialized speed prediction submodels. Personalized driver behavior modules and lightweight online adaptation schemes will be investigated to enable real-time tuning of model parameters. Finally, large-scale field validation on naturalistic driving data sets will be performed spanning urban, suburban, and highway environments to assess generalizability and robustness in diverse operational scenarios.

Bibliography

[1] International Transport Forum, “ITF transport outlook 2013,” *OECD Publishing/ITF*, *OECD/ITF*, Paris, France, Tech. Rep., 2013.

[2] L. Yan, Q. Zhang, B. Zheng, and K. He, “Modeling fuel-, vehicle-type-, and age-specific CO_2 emissions from global on-road vehicles in 1970–2020,” *Earth Syst. Sci. Data*, vol. 16, pp. 4497–4509, 2024, doi:10.5194/essd-16-4497-2024.

[3] R. Muncrief and B. Sharpe, “Overview of the heavy-duty vehicle market and CO_2 emissions in the European Union,” *International Council on Clean Transportation Working Paper* 2015–6, 1 Dec. 2015.

[4] C. Façanha, K. Blumberg, and J. Miller, “Global Transportation Energy and Climate Roadmap: The impact of transportation policies and their potential to reduce oil consumption and greenhouse gas emissions,” *International Council on Clean Transportation*, Nov. 2012.

[5] Y. Huang, E. C. Y. Ng, J. L. Zhou, N. C. Surawski, E. F. C. Chan, and G. Hong, “Eco-driving technology for sustainable road transport: A review”, *Renewable and Sustainable Energy Reviews*, vol. 93, pp. 596–609, Jun. 2018.

[6] M. A. S. Kamal, M. Mukai, J. Murata and T. Kawabe, "Ecological Vehicle Control on Roads With Up-Down Slopes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 783-794, Sept. 2011.

[7] E. Hellström, J. Åslund, and L. Nielsen, "Management of kinetic and electric energy in heavy trucks," *SAE Int. J. Engines*, vol. 3, no. 1,pp. 1152–1163, Apr. 2010.

[8] N. Murgovski, B. Egardt, and M. Nilsson, “Cooperative energy management of automated vehicles,” *Control Engineering Practice*, vol. 57, pp. 84–98, Sep. 2016.

[9] L. Johannesson, N. Murgovski, E. Jonasson, J. Hellgren, and B. Egardt, “Predictive energy management of hybrid long-haul trucks,” *Control Engineering Practice*, vol. 41, pp. 83–97, May. 2015.

[10] V. Turri, Yeojun Kim, J. Guanetti, K. H. Johansson and F. Borrelli, "A model predictive controller for non-cooperative eco-platooning," *2017 American Control Conference (ACC)*, Seattle, WA, USA, 2017, pp. 2309-2314, doi: 10.23919/ACC.2017.7963297.

[11] G. S. Sankar, M. Kim, and K. Han, “Data-Driven Leading Vehicle Speed Fore-

cast and Its Application to Ecological Predictive Cruise Control," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 11, pp. 11504–11514, Nov. 2022.

[12] R. Chaudhary, N. K. Sharma, R. Kala and S. N. Singh, "Deep Reinforcement Learning-Based Speed Predictor for Distributionally Robust Eco-Driving," *IEEE Access*, vol. 13, pp. 13904-13918, 2025, doi: 10.1109/ACCESS.2025.3530087.

[13] N. K. Sharma, A. Hamednia, N. Murgovski, E. R. Gelso, and J. Sjöberg, "Optimal Eco-Driving of a Heavy-Duty Vehicle Behind a Leading Heavy-Duty Vehicle," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 12, pp. 7792–7803, Dec. 2021.

[14] N. K. Sharma, N. Murgovski, and E. R. Gelso, "Online Learning for Chance-Constrained Observer of Leading Heavy-Duty Vehicle Power Capability," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 8356–8366, Jul. 2022.

[15] T. D. Gillespie, "Fundamentals of Vehicle Dynamics,"SAE International, 1992.

[16] Alan V. Oppenheim and Ronald W. Schafer, Discrete-Time Signal Processing, 3rd Edition, Pearson, 2010.

[17] Dr. Kamlesh Malpni "Detecting Outliers for Single Dimensional Data Using Interquartile Range," *International Journal of Engineering Research and Applications (IJERA)*, Vol. 09, No. 09, 2019, pp. 31-35

[18] MacQueen, J. B. (1967). "Some methods for classification and analysis of multivariate observations." In L. M. Le Cam & J. Neyman (Eds.), *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (Vol. 1, pp. 281–297). California: University of California Press.

[19] F. T. Liu, K. M. Ting and Z. -H. Zhou, "Isolation Forest," *2008 Eighth IEEE International Conference on Data Mining*, Pisa, Italy, 2008, pp. 413-422, doi: 10.1109/ICDM.2008.17.

[20] B. Egardt, N. Murgovski, "Model Predictive Control", *Lecture notes*, 2024.02. Chalmers University of Technology.

[21] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533-536, 1986, doi: 10.1038/323533a0.

[22] S. Choudhury, A. Roy and A. Naskar, "Truck Fuel Consumption Prediction Using Logistic Regression and Decision Tree Regression," *2023 6th International Conference on Computing, Power and Communication Technologies (GUCON)*, Greater Noida, India, 2023, pp. 1-6, doi: 10.1109/GUCON54154.2023.10312021.

[23] J. Liu, T. Qin, D. Chen, G. Wang, and T. Chen, "Driving power prediction of heavy commercial vehicles based on multi-task learning," *IFAC PapersOnLine*, vol. 58, no. 29, pp. 397–402, 2024, doi:10.1016/j.ifacol.2024.11.177.

[24] FACT SHEET Engine D13K540, EU6SCR Volvo Trucks. Driving Progress.

https://stpi.it.volvo.com/STPIFiles/Volvo/FactSheet/D13K540,

[25] Welch, P. D. (1967). The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. IEEE Transactions on Audio and Electroacoustics, 15(2), 70–73.

A

Appendix 1 Simulation Environment

The simulations were carried out using the following hardware and software configurations, as summarized in Table A.1.

Table A.1: Hardware and software configurations

Item	**Specification**
CPU	12th Gen Intel(R) Core(TM) i7-12850HX (2.10 GHz)
RAM	32 GB
Software	MATLAB R2021b

B

Appendix 2 Simulation outputs of the Observer

Table B.1: Calculated vector **b** with different anomaly–removal techniques under various driving-cycle logs

Driving cycle	**IF**	**KNN**	**Recursive Polyfit**	$\boldsymbol{b}_{\text{true}}$
		Simulated Logs		
Log 1 ($85t$)	$\begin{bmatrix} 3.86 \\ 5.68 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 5.51 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 4.23 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 4.39 \\ 3.62 \times 10^{-5} \end{bmatrix}$
Log 2 ($75t$)	$\begin{bmatrix} 4.39 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 5.64 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 4.87 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 4.97 \\ 4.11 \times 10^{-5} \end{bmatrix}$
Log 3 ($65t$)	$\begin{bmatrix} 5.05 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 6.33 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 5.55 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 5.74 \\ 4.75 \times 10^{-5} \end{bmatrix}$
Log 4 ($55t$)	$\begin{bmatrix} 5.82 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 6.56 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 7.22 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 6.78 \\ 5.6 \times 10^{-5} \end{bmatrix}$
Log 5 ($45t$)	$\begin{bmatrix} 6.94 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 7.68 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 8.50 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 8.29 \\ 6.85 \times 10^{-5} \end{bmatrix}$
Log 6 ($35t$)	$\begin{bmatrix} 8.70 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 8.52 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 10.98 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 10.66 \\ 8.81 \times 10^{-5} \end{bmatrix}$
Log 7 ($25t$)	$\begin{bmatrix} 9.09 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 9.65 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 12.22 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 14.92 \\ 1.23 \times 10^{-4} \end{bmatrix}$
		Real-world Logged Data		
Log 10 (EU)	$\begin{bmatrix} 7.73 \\ 5.24 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 8.90 \\ 5.24 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 8.67 \\ 5.24 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 8.58 \\ 7.44 \times 10^{-5} \end{bmatrix}$
Log 11 (EU)	$\begin{bmatrix} 8.71 \\ 5.51 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 9.63 \\ 5.51 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 9.71 \\ 5.24 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 10.19 \\ 1.36 \times 10^{-4} \end{bmatrix}$
Log 12 (BRA)	$\begin{bmatrix} 6.20 \\ 5.19 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 13.99 \\ 5.19 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 6.97 \\ 5.19 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 5.17 \\ 6.08 \times 10^{-5} \end{bmatrix}$
Log 13 (BRA)	$\begin{bmatrix} 6.32 \\ 5.09 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 10.27 \\ 5.09 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 6.39 \\ 5.09 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 5.31 \\ 6.19 \times 10^{-5} \end{bmatrix}$
Log 14 (BRA)	$\begin{bmatrix} 7.44 \\ 4.97 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 15.56 \\ 4.97 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 7.56 \\ 4.97 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 5.47 \\ 6.04 \times 10^{-5} \end{bmatrix}$

Table B.2: Observer accuracies with different estimation methods under various driving cycle logs

Driving cycle	**NN without b**	**Polyfit**	**Polyfit + NN**	$\boldsymbol{b}_{\text{true}}$
		Simulated Logs		
Log 1 ($75t$)	$\begin{bmatrix} 6.06 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 4.53 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 5.18 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 4.97 \\ 4.11 \times 10^{-5} \end{bmatrix}$
Log 2 ($55t$)	$\begin{bmatrix} 5.96 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 6.11 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 6.30 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 6.78 \\ 5.6 \times 10^{-5} \end{bmatrix}$
Log 3 ($40t$)	$\begin{bmatrix} 9.13 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 8.39 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 9.54 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 9.33 \\ 7.71 \times 10^{-5} \end{bmatrix}$
Log 4 ($35t$)	$\begin{bmatrix} 10.95 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 10.45 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 10.95 \\ 3.43 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 10.66 \\ 8.81 \times 10^{-5} \end{bmatrix}$
		Real-world Logged Data		
Log 5	$\begin{bmatrix} 7.83 \\ 5.74 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 16.91 \\ 5.74 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 12.70 \\ 3.45 \times 10^{-4} \end{bmatrix}$	$\begin{bmatrix} 12.98 \\ 1.74 \times 10^{-4} \end{bmatrix}$
Log 6	$\begin{bmatrix} 5.77 \\ 5.24 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 10.64 \\ 5.24 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 8.41 \\ 5.24 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 8.58 \\ 7.43 \times 10^{-5} \end{bmatrix}$
Log 7	$\begin{bmatrix} 12.2 \\ 2.15 \times 10^{-4} \end{bmatrix}$	$\begin{bmatrix} 9.37 \\ 9.85 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 8.42 \\ 2.15 \times 10^{-4} \end{bmatrix}$	$\begin{bmatrix} 8.84 \\ 5.08 \times 10^{-4} \end{bmatrix}$
Log 8	$\begin{bmatrix} 16.65 \\ 4.26 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 11.57 \\ 4.26 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 7.39 \\ 4.26 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 8.78 \\ 6.04 \times 10^{-5} \end{bmatrix}$
Log 9	$\begin{bmatrix} 13.63 \\ 4.63 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 11.31 \\ 4.63 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 9.11 \\ 2.78 \times 10^{-4} \end{bmatrix}$	$\begin{bmatrix} 10.21 \\ 1.14 \times 10^{-4} \end{bmatrix}$
Log 10	$\begin{bmatrix} 8.73 \\ 3.05 \times 10^{-4} \end{bmatrix}$	$\begin{bmatrix} 6.3 \\ 5.09 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 5.14 \\ 5.09 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 5.31 \\ 6.19 \times 10^{-5} \end{bmatrix}$
Log 11	$\begin{bmatrix} 3.15 \\ 3.03 \times 10^{-4} \end{bmatrix}$	$\begin{bmatrix} 6.52 \\ 5.05 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 5.48 \\ 5.09 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 5.16 \\ 5.92 \times 10^{-5} \end{bmatrix}$
Log 12	$\begin{bmatrix} 12.5 \\ 2.97 \times 10^{-4} \end{bmatrix}$	$\begin{bmatrix} 7.95 \\ 4.95 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 5.73 \\ 4.95 \times 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 5.46 \\ 6.02 \times 10^{-5} \end{bmatrix}$

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY