Eco-Driving Assistance Systems based on Model Predictive Control and Machine Learning

Eco-Fahrerassistenzsysteme basierend auf modellprädikativer Regelung und maschinellem Lernen

Vom Fachbereich Elektrotechnik und Informationstechnik
der Rheinland-Pfälzischen Technischen Universität
Kaiserslautern-Landau
zur Verleihung des akademischen Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
genehmigte Dissertation

von

M.Sc. Sai Krishna Chada
geboren in Nalgonda, Indien

D 386

Tag der mündlichen Prüfung:	28.11.2025
Dekan des Fachbereichs:	Prof. Dr.-Ing. Daniel Görges
Vorsitzender der Prüfungskommission:	Prof. Dr. rer. nat. Marco Rahm
1. Berichterstatter:	Prof. Dr.-Ing. Daniel Görges
2. Berichterstatter:	Prof. Dr.-Ing. Roman Teutsch

Acknowledgements

I am profoundly grateful for the support and encouragement that I received throughout my doctoral journey from my supervisors, colleagues, families and friends.

First and foremost, I wish to extend my heartfelt gratitude to my advisor, Prof. Dr.-Ing. Daniel Görges, for his unwavering guidance and mentorship throughout my doctoral journey. I am deeply thankful for the generous amount of time he devoted to engaging in scientific discussions, and for fostering a highly supportive and convinient work environment within the Institute of Electromobility at the RPTU Kaiserslautern-Landau.

Furthermore, I would like to express my deepest gratitude to Prof. Dr.-Ing. Roman Teutsch for his continuous support during my dissertation work, and joining the dissertation committee as a reviewer. I also want to thank Prof. Dr. rer. nat. Marco Rahm for joining the thesis committee as a chairman.

Furthermore, I would like to sincerely thank apl. Prof. Dr.-Ing. Achim Ebert for his support during my work. I would also like to thank my colleagues and fellow researchers at the Institute of Electromobility at the RPTU Kaiserslautern-Laudau, for many insightful scientific discussions.

Furthermore, I would like to acknowledge the funding provided by the Ministry for Education and Research (BMBF) of the Federal Republic of Germany and the support from the Center of Commercial Vehicle Technology (Zentrum für Nutzfahrzeugtechnologie, ZNT) at the RPTU Kaiserslautern-Landau. Their financial assistance has allowed me to conduct this research and make valuable contributions to my field.

In addition, my special thanks go to my students, who provided great assistance to my research endeavors through their bachelor and master thesis contributions. I am also deeply thankful to the participants of the driving simulator studies, whose active engagement provided invaluable data for my dissertation.

I dedicate this dissertation to my parents, my in-laws, my wife, Mounika and my sons, Arjun and Ashrith whose unwavering love, support and patience have been a constant source of strength throughout my PhD journey.

Eberdingen, December 2025 *Sai Krishna Chada*

Abstract

Energy efficiency in vehicles has emerged as a central priority for the automotive industry, driven by pressing environmental concerns, economic factors, and regulatory demands. This dissertation proposes and validates a predictive eco-driving assistance system (EDAS) designed to assist drivers in adopting a more energy-efficient driving style, specifically for battery electric vehicles, while simultaneously enhancing driving safety and comfort. The proposed predictive EDAS is equipped with eco-driving model predictive controllers (Eco-MPCs) that leverage data from onboard sensors, signal phase and timing information from traffic light infrastructure, and geographical information of the driving route to compute energy-optimal speed trajectories. These optimal speed recommendations are communicated to drivers through an eco-driving feedback system providing both visual and auditory cues. To assess the effectiveness of the proposed system across diverse traffic scenarios, user studies were conducted using a dynamic driving simulator. The objective results demonstrated that drivers using the predictive EDAS achieved significant energy savings, reduced speed limit violations, maintained safer speeds on turns and curved roads, and minimized unnecessary stops at signalized intersections. Furthermore, recurrent neural network-based prediction models were developed to accurately forecast the behavior of the leader vehicle by incorporating critical environmental information. To further enhance the efficiency of the system, this work introduces a learning-based framework for modeling individual driver behavior using machine learning techniques, thereby improving drivers' advisory speed-tracking ability. Finally, user acceptance of the predictive EDAS was evaluated using the Technology Acceptance Model (TAM) and the Theory of Planned Behavior (TPB) model. The findings revealed an overall positive user attitude, with perceived usefulness and perceived behavioral control emerging as key factors influencing drivers' intentions to adopt the predictive EDAS in daily life.

Contents

Nomenclature

Acronyms

ACC	Adaptive Cruise Control
ADAS	Advanced Driver Assistance Systems
API	Application Programming Interface
ATT	Attitude Toward Behaviour
AVE	Average Variance Extracted
BEV	Battery Electric Vehicle
BI	Behavioral Intention
Cα	Cronbach's Alpha
CA	Constant Acceleration
cEDAS	Eco-Driving Assistance System with Driver Speed Error Compensation
CM4SL	CarMaker4Simulink
CO2	Carbondioxide
CR	Composite Reliability
CV	Constant Velocity
C-V2X	Cellular Vehicle-to-Everything
DiL	Driver-in-the-Loop
DNN	Deep Neural Network
DoF	Degree of Freedom
DP	Dynamic Programming
DSRC	Dedicated Short-Range Communications
Eco-HMPC	Eco-Driving Hybrid Model Predictive Controller
Eco-MPC	Eco-Driving Model Predictive Controller
EDAS	Eco-Driving Asssitance System
FG	Feature Group
FTMPC	Frozen-Time Model Predictive Control
GPS	Global Positioning System
GRU	Gated Recurrent Unit
GWOS	Green-Wave Optimized Speed
HCA	Hierarchical Cluster Analysis
HMI	Human Machine Interface
HMPC	Hybrid Model Predictive Control
HUD	Heads-up Display
ICE	Internal Combustion Engine
ITS	Intelligent Transportation System
KPI	Key Performance Indicator
LSTM	Long-Short Term Memory
LTE	Long-Term Evolution
LTI	Linear Time-Invariant

MAE	Mean Absolute Error
MCMC	Monte Carlo Markov Chain
MIQCQP	Mixed-Integer Quadratically Constrained Quadratic Programming
MPC	Model Predictive Control
MSE	Mean Squared Error
OBU	On-Board Unit
OCP	Optimal Control Problem
OSM	Open Street Maps
PBC	Perceived Behavioral Control
PC	Principal Component
PCA	Principal Component Analysis
pEDAS	Predictive Eco-driving Asssitance System
PEoU	Perceived Ease of Use
PMP	Pontryagin's Minimum Principle
PMPC	Prescient Model Predictive Control
PnG	Pulse and Glide
PU	Perceived Usefulness
QP	Quadratic Programming
RC	Rotated Component
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
RSU	Road Side Unit
s.t.	subject to
SD	Standard Deviation
SEM	Structural Equation Modeling
SN	Subjective Norms
SoC	State of Charge
SPaT	Signal Phase and Timing
SRTM	Shuttle Radar Topography Mission
SUMO	Simulation of Urban Mobility
SV	Stochastic Volatility
TAM	Technology Acceptance Model
TCN	Temporal Convolutional Network
TL	Traffic Light
TPB	Theory of Planned Behavior
TRA	Theory of Reasoned Action
TraCI	Traffic Control Interface
UTAUT	Unified Theory of Acceptance and Use of Technology
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
VAR	Vector Autoregressive

Symbols

$(.)_{\text{h,without EDAS}}$	Variable of the host car without assistance from EDAS
$(.)_{\text{h,with EDAS}}$	Variable of the host car equipped with EDAS
c_{a}	Aerodynmic drag coefficient
A_{f}	Frontal cross-sectional area of the vehicle
ρ	Density of the air
g	Acceleration due to gravity
c_r	Rolling friction coefficient
θ	Road slope
p_*	Coefficients for approximation of aerodynamic friction
H	Prediction horizon
$C_{(.)}$	Cell state
$f_{(.)}$	Forget gate
$i_{(.)}$	Input gate
$o_{(.)}$	Output gate
U_*, W_*	Weights in RNNs
b_*	Biases in RNNs
r_{t}	Reset gate
y_{t}	Update gate
h_{t}	Hidden state
sig	Sigmoid activation function
$tanh$	Tanh activation function
l	Vehicle length
d	Relative distance
s_{TL}	Traffic light signal state
$\sigma^2_{(.)}$	Variance
$\sigma_{(.)}$	Standard deviation
$\mu_{(.)}$	Mean
m	Mass of the vehicle with driver and cargo
m_r	Equivalent mass of the rotational components in the vehicle
F_{t}	Traction force
F_{b}	Braking force
F_{a}	Aerodynamic resistance
F_{roll}	Rolling resistance
F_{g}	Gradient resistance
v_{min}	Minimum road speed limit
v_{max}	Maximum road speed limit
v_{ref}	Reference speed
$v_{\text{h,opt}}$	Optimal speed of the host vehicle
$v_{(.)}$	Vehicle speed
$\hat{v}_{(.)}$	Predicted vehicle speed
Δv_{h}	Driver speed tracking error
γ	Rate of mean reversion
ΔW	Increment of a Wiener process

ν	Average magnitude of random fluctuations
$X_{(.)}$	Rate of driver speed tracking error variation
$\mathcal{N}(.)$	Normal distribution
$E_{(.)}$	Energy consumption
a	Acceleration
t	Time
Δt	Sampling time
s	Distance
λ	Standardized factor loading
CI	Confidence interval
B	Unstandardized coefficient
$SE\ B$	Standard error for the unstandardized coefficient
β	Standardized coefficient
R^2	Coefficient of determination
R^2_{Adj}	Adjusted coefficient of determination
ΔR^2	Change in coefficient of determination
p	Statistical significance (p-value)
$\odot$	Scalar product of the vectors
G	Matrix with principal components
H	Matrix with stored input variables of driving data
X	Normalized matrix
$D_{(.)}$	Euclidean distance
Z_x	Covariance matrix
Z_Y	Diagonal matrix with eigenvalues
P	Matrix of eigenvectors
N	Total number of data points in the driving dataset

1 Introduction

1.1 Background and Motivation

The global automotive industry is predicted to grow at a compound annual rate of 3.71% from 85.32 million in 2020 to 122.83 million units by 2030 [Ren21]. With the increasing vehicle ownership, the transportation industry is confronted with enormous challenges such as traffic congestion and environmental pollution due to the emission of harmful greenhouse gases [Bar10]. To tackle the pressing issue of alarming climate change, the legislative bodies across the globe have set strict regulatory standards to curb the global CO_2 emissions emanating from the road vehicles [EU 23]. Furthermore, the transportation industry is transitioning towards sustainable mobility solutions such as battery electric vehicles (BEVs). Although BEVs are regarded as environmentally friendly solution due to their ability to produce zero tailpipe emissions [NPB+14], they must however, be operated as efficiently as possible since they rely on electric power produced from fossil fuels in the transition phase [Tho12]. Moreover, the limited battery capacity and driving range are the major challenges to BEV adoption among users [NFC+15]. Therefore, it is crucial to enhance the energy efficiency of BEVs through advanced technologies in order to address the aforementioned challenges.

Another sustainable transportation solution that reduces the requirement for private vehicle ownership and facilitates collective mobility, is public transportation. This mode of transportation offers a variety of benefits, including cost-effectiveness, shorter commute times, reduced traffic congestion and air pollution. According to [ACE22], around 93.5% of the buses in the European Union are diesel-powered, thus positioning the internal combustion engines (ICEs) as the primary power source in this segment. Despite being equipped with modern ICEs that comply with strict emission regulations, buses must still reduce fuel consumption to enhance environmental sustainability and decrease overall fleet operational costs.

To enhance the energy efficiency in BEVs and ICE vehicles, the global research is focused on the one hand on integrating lightweight materials [PFN12], developing advanced battery technologies [NPB+14], amd enhancing the powertrain [SKT+20, LKSM20] and thermal [GWM+21, HAK+22] efficiencies. On the other hand, eco-driving is another area of research that has received significant attention over the years, known as a cost-effective approach to improving driver's driving style in an energy-efficient manner [Bar10, AM14]. This dissertation contributes to the field of eco-driving with the primary goal of enhancing the energy efficiency in both BEVs and ICE vehicles.

The state-of-the-art relevant to this dissertation is presented across several sections. First, an overview of eco-driving research is provided in Section 1.2. A detailed discussion of the state-of-the-art related to each specific research question is offered in Section 1.3. Additionally, the theoretical background of the methods employed throughout this dissertation is outlined in Chapter 2.

1.2 State-of-the-Art of Eco-Driving

Many studies in the past have reviewed the benefits and limitations of the eco-driving policies [Bar10, AM14, HNZ+18]. Initial studies have presented several eco-driving strategies for drivers using the conventional ICE vehicles [Bar10, ACNW17]. For instance, encouraging the drivers to avoid unnecessary acceleration and deceleration, reduce idling, travel with constant driving speeds, drive below the regulatory speed limits, anticipate traffic objects and traffic light signals are some of the existing eco-driving techniques [Bar10]. To increase awareness among drivers on fuel-efficient driving, the initial eco-driving systems in the market provided simplified indications on the vehicle instrument cluster, such as, the eco gauge that displays the instantaneous fuel consumption during the trip and average fuel economy after the trip [Hon]. Over the years, many automotive manufacturers have introduced features that promote efficient driving techniques in conventional and hybrid electric vehicles, such as Nissan eco-pedal [Nisb], Toyota eco-driving indicator [Toy], Volvo eco-drive mode [Volb] and Ford eco-mode [For], etc. These features primarily assist drivers in minimizing sudden accelerations and decelerations by adjusting the throttle response, and moreover, indicating the optimal gear shift to optimize fuel economy. In recent years, BEVs like the Nissan Leaf [Nisa] and Volkswagen ID.3 [Vola] have introduced ECO modes, where specific settings are adjusted to enhance efficiency and extend the driving range. For example, these adjustments include increasing regenerative braking when the accelerator is released (i.e., during coasting), lowering the power output of the electric motor, and limiting the power to auxiliary systems such as for heating and air conditioning. In heavy-duty vehicles, advanced driver assistance systems (ADAS) such as Volvo Trucks' I-See and Scania's Look-Ahead or Active Prediction systems utilize global positioning system (GPS) data and digital map information to anticipate changes in road elevation, curvature, and speed limits, enabling the vehicle to optimize its speed and gear selection accordingly [HIÅN09]. The limitation in the current eco-driving systems available in the market is that they employ simple optimization methods and offer momentary suggestions to drivers aimed at minimizing the fuel or energy consumption within constrained driving situations.

To promote the adoption of energy-efficient driving habits among drivers, various eco-driving programs have been investigated in the literature (Table A.1 provides a curated list). Coaching the drivers on eco-driving techniques has demonstrated the potential to reduce fuel consumption by 5.5-16.9% [BBD+09, SDN15, BCV+17, ACNW17, PBEM21].

Many studies have pointed out that the drivers tend to forget the eco-driving techniques learned through training sessions and get back to their old driving habits in a short period, thus showcasing no long term benefits of such trainings [BBD+09, BCV+17]. For example, in a driving simulator experiment by [DBF+13], displaying a few eco-driving advisory messages to the drivers has improved their fuel savings up to 12%. However, in a consecutive experiment when the drivers were not provided with any advisory messages, their driving behavior did not yield in any fuel saving benefits. This points out the necessity for a continuous eco-driving feedback or assistance system [DBF+13, VBH+13, SSKK14, ZWRZ15, FKA15, JHJ15, GICC21] to avail the long-term benefits in terms of fuel or energy savings.

Moreover, to enhance driver engagement and improve the overall driving experience while using an eco-driving assistance system, an effective in-vehicle human-machine interface (HMI) plays a crucial role. In Mcllroy's study [MSGW17], the use of in-vehicle feedback systems to provide continuous suggestions in a visual, haptic and auditory manner has enabled drivers to adopt additional eco-driving behaviors such as coasting and reduce harsh acceleration. [JHJ15] assessed visual and haptic eco-driving feedback systems within the context of hill driving in a driving simulator investigation. Their findings indicated that any form of eco-driving advice enhances the driving performance, whilst visual feedback proving to be the most effective. Despite its effectiveness, one of the negative impacts of a visual feedback system is that it enforces drivers to focus on the suggestions and may increase the risk of accidents by causing driver distractions [AM14, LVR+20]. Therefore, it is crucial for a HMI to provide simple and clear indications that are easy for the drivers to interpret.

Many studies have utilized predictive information about the route and surroundings to enable proactive adjustments to driving strategies, thereby enhancing energy efficiency [NHSS10, DBF+13, AV11, SSKK14]. The concept of predictive eco-driving assistance system (EDAS) based on topology (such as elevation, curvature and speed limits) of the route has been studied by [DBF+13]. In this work, the efficacy of a predictive EDAS that provides the driver with situational and contextual advice to improve the fuel efficiency in heavy commercial vehicle was analysed through a driving simulator experiment. However, the study relied on very limited parameters to determine an optimal driving strategy. A recent work [AM18] indicates that the predictive information of the surrounding traffic elements obtained from the wireless communication systems such as vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) in combination with eco-driving techniques can help to reduce the CO2 emissions and traffic congestion on a network level. A few studies have demonstrated the benefits of using the signal phase and timing (SPaT) information from the traffic light signals to optimally plan the host vehicle trajectories at signalized intersections [AV11, SSKK14]. In [SSKK14], the authors developed an eco-driving support system to suggest an optimal speed with which the drivers can cross the immediate traffic light signals at green. Using the support system, drivers achieved average fuel savings up to 15.9-18.4%. The limitation in their study is that the interactions with surrounding traffic were not considered.

Numerous studies in the literature have formulated the eco-driving objectives as an optimization problem [PWD18, TNS18, HVS19, WYBD25]. The formulations range from simple approaches, such as minimizing the energy consumed by the vehicle [PWD18, HVS19], reducing acceleration and deceleration events [LLRW09, LGW17] or optimizing gear shifting strategies for ICE vehicles [NSLF19], to more complex optimization methods. Advanced approaches often address multiple objectives simultaneously, such as jointly optimizing route selection and vehicle speed [MLZC18, TNS18] or optimizing the distance to the leader vehicle while enhancing driving comfort [KMMK13, JJG20]. However, the majority of these studies primarily focused on problem formulation aspects and lacked driver-in-the-loop validations. Furthermore, it is essential to examine the user acceptance of advanced eco-driving approaches, as it plays a crucial role in understanding user preferences and ensuring the long-term adoption and effective utilization of these technologies.

This dissertation aims to make a significant contribution to the field of eco-driving by addressing the critical gaps identified. The primary objective is to develop a novel predictive eco-driving assistance system (EDAS) capable of providing continuous speed advisory feedback to drivers across various driving situations. This involves improving the problem formulation of the eco-driving optimal control problem (OCP) for both BEVs and ICE vehicles, with the goal of enhancing not only energy-efficient driving behavior but also driving safety and comfort. Furthermore, the impact of the surrounding environmental and driver-specific information in enhancing the efficiency of the predictive EDAS will be studied in this dissertation. Moreover, the proposed EDAS will be evaluated both objectively and subjectively through driving simulator studies in this work.

1.3 Research Questions and Contributions

This section presents the research questions formulated on the basis of the identified gaps in the existing literature. For each research question, the state-of-the-art is reviewed and the novel contributions made within the scope of this dissertation are summarized.

Q1. How can a multi-objective OCP for eco-driving be formulated mathematically to control both BEVs and ICE vehicles in real-time under varied traffic situations?

State-of-the-Art: The optimal control problem (OCP) for eco-driving is a multi-objective problem as in addition to minimizing the host vehicle's energy or fuel consumption improving the driving safety and comfort are focussed. To solve the eco-driving OCP [SNO15], previous studies have investigated several algorithms such as Pontryagin' minimum principle (PMP) [SRDV13] and dynamic programming (DP) [LGL14, MGC$^+$16, OOF17]. Despite the fact that PMP provides only necessary conditions for optimality and DP is capable of yielding a globally optimal

solution, both approaches are computationally intensive and therefore not suitable for online implementation. While overcoming the limitations of the aforementioned methods, model predictive control (MPC) [RAG+10, HCTK12] has gained popularity due to its ability to handle the state and input constraints effectively and facilitate online control. Several studies in the past employed MPC to control the host vehicle in an energy-efficient manner while following a leader vehicle [LLRW09, WGL18, JJG20]. However, the influence due to other important factors in the environment such as traffic light signals and traffic signs (e.g. stop signs) on the host vehicle while performing car-following was rarely investigated in the literature. There is no conclusion on which optimization structure is the best for the real-time control of both BEVs and ICE vehicles under varied traffic situations.

Contributions: One of the contributions in this dissertation is to propose an optimization structure with two linear model predictive controllers (MPCs) for reference tracking and car-following, respectively. Furthermore, the DP optimization problem is solved in offline at the beginning of the trip and the optimal solution is used as a reference for online implementation. To select the appropriate MPC controller that derives the optimal control sequences for the host vehicle, a switching logic is proposed in this work. Furthermore, the optimization problems for both MPCs are mathematically formulated. The optimization problem has been developed to accomodate the additional constaints due to traffic light signals and stop signs while performing car-following. The performance of all the proposed MPC approaches is evaluated through simulations in this work. Another contribution of this work is to extend the problem formulation to Eco-(H)MPC (a hybrid model predictive control approach) for a city bus equipped with a conventional ICE use case. The optimization problem combines the car-following MPC and reference-tracking MPC to one single problem formulation. The optimization problem is formulated as a mixed-integer quadratically constrained quadratic programming (MIQCQP) problem to not only optimize the continuous power from ICE but also control the engine ON/OFF state. It is also revealed in this dissertation how the information about the leader vehicle's future speed affects the performance of the eco-driving MPC. A comparison about the computation times for different prediction horizon has been presented.

Q2. What environmental information can be used for accurate leader vehicle speed prediction to improve the efficiency of the car-following MPC?

State-of-the-Art: [LLRW09, WGL18, JJG20] have highlighted the energy-saving benefits of incorporating the future velocities of the leader vehicle within a car-following MPC framework. However, these studies commonly assumed that the future velocities of the leader vehicle are perfectly known. In reality, such information is not available a priori in real-world scenarios and must instead be predicted using either behavioral models or data-driven approaches [RAG+21]. Previous studies on speed forecasting in transportation systems primarily focused on network wide traffic speed prediction [CKPW20] and host vehicle velocity prediction

[LSBL14, LAG+19, SKSH19, WHK+23] for the purpose of efficient energy management in hybrid electric vehicles [SHMS15, GAL+19]. [LSBL14] presented a comparative analysis on the parametric and non-parametric approaches for speed prediction in highway driving. [LAG+19] investigated on the one hand stochastic models such as Markov chain and conditional linear Gaussian (CLG) for the host vehicle velocity prediction. On the other hand, deterministic models such as auto-regressive moving average (ARMA), nonlinear auto-regressive exogenous model (NARX) and recurrent neural networks such as long-short term memory (LSTM) units were studied. [SKSH19] proposed a fuzzy Markov chain model with speed constraints to perform host vehicle speed prediction. Moreover, the authors in [WHK+23] studied longitudinal vehicle speed prediction in urban environments using CLG and deep neural networks (DNNs). Only a handful of studies in the literature investigated the leader vehicle speed prediction for the energy optimal control of the host vehicle [SWD15, MWS+15, JCG20, SKH22]. [SWD15] presented a simplified prediction model that uses sinusoidal functions to predict the behavior of a target vehicle and demonstrated a predictive cruise control approach. [MWS+15] employed a Bayesian network approach with a CLG model to predict target vehicle information for a cooperative adaptive cruise control use case. [SKH22] used a vector autoregressive model to generate simultaneous predictions of the target vehicle. In [JCG20], the authors investigated on an LSTM technique for predicting the target vehicle speed in an urban environment for an energy-optimal adaptive cruise control use case. In previous research [SKSH19, LAG+19, JCG20, WHK+23], speed prediction models were trained using datasets that were collected from repeated trials on the same driving route, with limited or no surrounding traffic considered. While this approach allows for relatively reasonable predictions of driving patterns for evaluations on the same driving route, however, the limitation is that the prediction model may not generalize well on the unseen data from a different route. Moreover, the previous studies on vehicle speed prediction did not consider the influence of multiple preceding vehicles and its influence on the prediction accuracy and are only limited to either urban or highway scenarios.

Contributions: In this dissertation, to predict the leader vehicle future velocities, both deep recurrent neural networks (LSTM and GRU) and physics-based models (CV and CA) are explored. Additionally, a scalable and more generalizable approach is proposed for preparing time series driving datasets from urban and highway networks using the microscopic traffic simulation tool SUMO. This work further investigates the influence of various input variables (e.g. leader vehicle speed, multiple preceding vehicles behavior, traffic light signal phase and road speed limits) as well as the impact of incorporating additional V2V and V2I information on the accuracy of leader vehicle speed predictions. This dissertation identifies the critical environmental information that enhance the prediction accuracy and evaluates the performance of car-following MPC on predicted leader vehicle velocities, while investigating the energy-saving potential across different

prediction horizons.

Q3. How effective is the predictive EDAS in improving drivers' driving style in an energy-efficient manner?

State-of-the-Art: Although several previous studies on eco-driving support systems [BBD+09, SDN15, BCV+17, ACNW17, PBEM21] have demonstrated significant reductions in fuel consumption, these works often rely on relatively simple optimization methods and provide only momentary recommendations to drivers in less complex driving scenarios. In the context of advanced eco-driving approaches, several studies have implemented MPC for predictive cruise control [LLRW09, WGL18, JCG20, HLW+24], however, a limited number of studies have investigated driver-in-the-loop demonstrations of eco-driving systems [WAG+19, HLW+24]. For instance, [WAG+19] organised field trials to demonstrate a connected eco-driving system for a heavy-duty truck, providing speed recommendations to navigate through signalized intersections. However, a limitation of this study was the use of a simplified eco-driving algorithm that does not account for disturbances caused by surrounding traffic. In [HLW+24], a predictive cruise control system was explored for car-following scenarios involving a leader vehicle. The limitations of this study, as well as others [LLRW09, WGL18, JCG20], include the absence of considerations for critical elements, such as traffic light signals and traffic signs, that influence the host vehicle's behavior during car-following. Furthermore, there is a notable lack of literature investigating the potential benefits of predictive EDAS for BEVs.

Contributions: In this work, a continuous eco-driving feedback (visual and acoustic) system was developed to provide drivers with optimal advisory speed and alerts during unsafe driving situations. To assess the effectiveness of the proposed predictive EDAS, user studies were conducted with 34 participants using a dynamic driving simulator. A virtual driving environment featuring both urban and highway driving routes with a total length of 13.2 km was modeled in IPG CarMaker. This environment featured realistic driving scenarios, including surrounding traffic objects, traffic light signals, traffic signs (e.g., stop signs) and curved roads. To ensure realistic interactions between the host vehicle and surrounding traffic, a co-simulation framework was developed, enabling real-time data exchange between the traffic simulation software SUMO and IPG CarMaker. Objective evaluation of the predictive EDAS was performed using key performance indicators (KPIs) derived from the driving data collected during the user studies. This dissertation highlights the potential of predictive EDAS to reduce energy consumption across drivers with diverse driving styles, while also assisting in lowering speed limit violations and minimizing unnecessary stops at traffic light signals.

Q4. How to use driver information to improve the efficiency of predictive EDAS?

State-of-the-Art: Predictive eco-driving assistance systems have the potential to provide continuous optimal speed suggestions to drivers (by leveraging predictive

information), leading to improved driving behavior and long-term energy savings. Previous EDAS approaches predominantly considered average driver characteristics and often may not adapt to the specific driving styles of individual drivers. Drivers', however, may frequently fail to precisely track the recommended advisory speed suggested by the driver-assistance systems due to their persisting perception and reaction delays. Such a behavior will induce speed-tracking errors and degrade the overall system performance. To improve the effectiveness of the predictive EDAS, driver behavior must be modeled and delays associated with drivers must be predicted. To effectively model and analyze driver behavior, numerous studies advocate a data-centric approach and use of machine-learning techniques [MT16]. For clustering driving styles, probabilistic models for unsupervised learning, such as Gaussian mixture models and hidden Markov models, have been extensively studied [MT16]. Additionally, some studies have employed principal component analysis (PCA) [WLW+20] to extract key driving patterns and reduce dimensionality of the original variables. Furthermore, hierarchical cluster analysis (HCA) [WLW+20] has also been used to group drivers with similar driving patterns by computing the Euclidean distance between samples. For the online classification of the driver's data, the k-nearest neighbor algorithm has been applied in [WLW+20]. Moreover, neural networks are increasingly favored for driver behavior modeling, as driving style classification can be framed as a time series problem due to the sequential nature of the data. [BKK18] explored temporal convolutional networks (TCNs), which adopt a simple convolutional neural network architecture tailored for time series tasks. Their results demonstrated that TCNs outperformed traditional recurrent neural networks (RNNs) across various benchmarks. Only a few studies in the literature have investigated on the prediction of driver error while tracking advisory speeds [WLW+20, QWW+18, CL22]. The authors of [QWW+18] used Markov Chain Monte Carlo (MCMC) method to model the driver error for a cooperative eco-driving system. However, in their approach the transition probability matrices of drivers' must be priorly known, which makes the storing of these matrices increasingly complex as the sample size grows. In [WLW+20], a deterministic nonlinear autoregressive neural network was implemented to predict the driver error for the cooperative on-ramp merging scenario. [CL22] proposed an EDAS with driver delay compensation, however a major limitation is that the study was restricted to deceleration scenarios alone.

Contributions: In this work, a learning-based driver delay compensation approach is proposed to improve the speed-tracking accuracy and thereby enhance the efficiency of the predictive EDAS. To model the driver behavior and identify driving patterns through driving data clustering, data-driven unsupervised learning methods, namely, principal component analysis (PCA) and hierarchical cluster analysis (HCA) are investigated. Additionally, a supervised deep-learning method, the temporal convolutional network (TCN) is explored for driving style classification. This work also examines deterministic gated recurrent unit (GRU) and stochastic volatility (SV) models to predict driver speed-tracking delay. Finally, the effec-

tiveness of the proposed human driver delay compensation approach is evaluated through user studies, concluding with an analysis of which regression model (deterministic or stochastic) proves more effective in achieving better speed-tracking accuracy and highlighting the resulting energy savings improvements.

Q5. What are the influencing factors that affect the user acceptance of predictive EDAS?

State-of-the-Art: To successfully incorporate any new technology into the transportation sector, the user acceptance plays a pivotal role. Previous studies utilized theories of human behavior to identify the factors that influence a user's intention to use a technology [RLHS17]. Commonly used models include the technology acceptance model (TAM) [Dav85, DBW89], theory of planned behavior (TPB) [Ajz91], and the unified theory of acceptance and use of technology (UTAUT) [VMDD03, VTX12], which assess acceptance using both behavioral intention and actual system use. These models were initially applied to evaluate user acceptance of computer-based information systems [Igb93]. Prior research on the user acceptance in advanced driver assistance systems (ADAS) utilized established theories and models to evaluate the variables that contribute to the intention to use state-of-the-art technologies such as lane keeping assist, adaptive cruise control, and lane departure warning systems [RLHS17, LDX$^+$19, VPD$^+$20]. [RLHS17] studied TAM, TPB, and UTAUT for modeling driver acceptance in ADAS, and gathered subjective data through both driving simulator and online survey approaches. Their findings indicated that all three models were able to explain the driver acceptance in terms of behavioral intention (BI) to use ADAS. Moreover, TAM was found to be better at predicting BI, and TPB was regarded as the second best. It also showed that the moderating factors of UTAUT were not significant in predicting BI. Furthermore, TAM was used in [LDX$^+$19] to evaluate the effectiveness of lane departure warning (LDW) and forward collision warning (FCW) assistance systems through field operation tests. Their study revealed that the drivers' acceptance of FCW was much higher than LDW. They also stated that the road type and driving experience have a significant influence on driving behaviors. [VPD$^+$20] utilized TAM to assess the driver acceptance of a smartphone-based ADAS that combined navigation and collision warning features with an objective to improve driving safety. The results of their experiment showed that TAM was able to explain significant variance in BI, thus user acceptance as well. The study revealed that the main factors that influenced the users in accepting the system were attitude and perceived usefulness. The limitation in their study is the relatively small sample size used of 23 participants that may limit the generalizability of the findings. Furthermore, the effect of moderating factors such as varying age, gender and driving experience on the acceptance of ADAS was investigated in previous works [SPP15, LDX$^+$19, GP21]. The literature is rich in studies that evaluated the user acceptance on simplified eco-driving assistance systems for ICE vehicles [VBH$^+$13, SSKK14, HBBD15]. The authors in [SSKK14]

evaluated an eco-driving support system that suggested optimal gear-shifting indication and acceleration/deceleration behaviors in less complex scenarios. The acceptance questionnaire was designed based on perceived ease of use, perceived usefulness and behavioral intention to use, and were evaluated on a likert scale. Moreover, [VHD97] scale was used to determine the attitude towards using the system. Moreover, [HBBD15] used UTAUT to evaluate the driver acceptance of various eco-driving feedback systems such as haptic, graphical user interface and series HMI in heavy duty vehicles. Very limited studies in the past have evaluated the user acceptance on advanced eco-driving assistance systems, specifically with a focus on BEVs.

Contributions: This work investigates user acceptance of a predictive EDAS for BEVs through a survey questionnaire based on the constructs of the TAM and TPB models. The survey was administered to 41 participants who had previously tested the predictive EDAS on a driving simulator, and a subjective evaluation was conducted using the collected data. A priori hypotheses were proposed to assess the significant impact of various constructs on user behavior and actual system usage. Subsequently, structural equation modeling (SEM) was applied in a two-step process: first, a measurement model analysis to evaluate the reliability and validity of the latent constructs, and second, a structural model analysis to test the hypotheses using hierarchical regression analysis. This dissertation provides valuable insights into the key factors influencing users' behavioral intention to adopt predictive EDAS.

1.4 Structure

The structure of this dissertation is outlined in this chapter and illustrated in Figure 1.1. It is organized into eight chapters, encompassing the state-of-the-art, methodological development, and validation.

Chapter 1 presents the background and motivation for the dissertation, establishes the current state-of-the-art in eco-driving, identifies existing research gaps, and formulates the resulting research questions. For each research question, the chapter reviews the relevant literature and summarizes the novel contributions made within the scope of this dissertation.

Chapter 2 provides the theoretical foundations necessary for the subsequent work. It begins by introducing key concepts in optimal control theory, including the optimal control problem (OCP) and Pontryagin's minimum principle. This is followed by a detailed discussion of dynamic programming (DP) and model predictive control (MPC) in Section 2.1, which form the basis for the control strategies applied throughout the study. Section 2.2 presents key machine learning techniques, including dimensionality reduction, clustering, classification, regression, and relevant evaluation metrics. The chapter

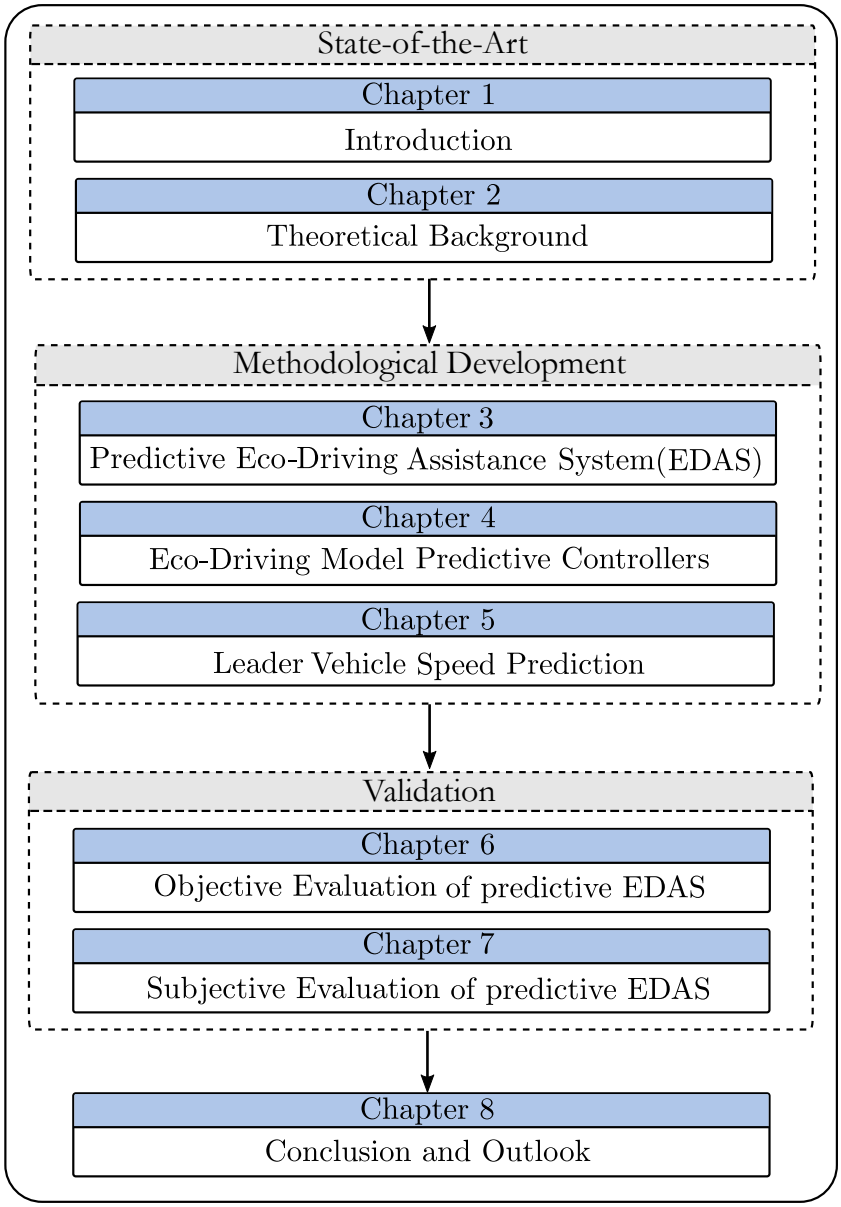


Figure 1.1: Structure of the dissertation

concludes with an overview of state-of-the-art user acceptance models, specifically the technology acceptance model (TAM) and the theory of planned behavior (TPB), which are used to assess the acceptance of the proposed technologies in this work.

Chapter 3 introduces the predictive eco-driving assistance system (EDAS) developed in this dissertation. Section 3.1 provides a detailed overview of the system architecture, including core components such as predictive information processing, switching logic, the eco-driving feedback system, and the integration of a driver-in-the-loop setup. The chapter continues with the design of a learning-based driver error compensation framework, presented in Section 3.1.5.

Chapter 4 is dedicated to the development of eco-driving model predictive controllers (Eco-MPCs). It begins with the modeling of longitudinal vehicle dynamics, covering both system dynamics and prediction models. The chapter then focuses on the design of Eco-MPCs for battery electric vehicles (BEVs), including powertrain modeling, problem formulations for reference tracking and car-following MPCs, and the formulation of the optimization problem using quadratic programming with linear inequality constraints. A comprehensive performance evaluation of the proposed controllers is provided. Additionally, the chapter addresses the development of the Eco-HMPC problem formulation for city buses with internal combustion engines (ICEs), and presents a corresponding performance evaluation. The chapter concludes by summarizing the key contributions and findings.

To improve the efficiency of the car-following MPC proposed in Chapter 4, accurately predicting the leader vehicle speed is focused in Chapter 5. It begins with the data preparation process for training the prediction models. Section 5.2 presents a detailed discussion of the results, including the performance evaluation of the models and the impact of incorporating predicted leader vehicle speeds into the car-following MPC.

Chapter 6 presents the objective evaluation of the predictive EDAS through two main assessments. The first evaluation, described in Section 6.1, examines system performance in a driver-in-the-loop setup. This section outlines the experimental procedure, analyzes driver behavior, and presents the corresponding objective performance results. The second evaluation, detailed in Section 6.2, investigates the efficiency gains achieved by incorporating driver behavior modeling into the predictive EDAS. It includes the results of a learning-based approach for compensating speed-tracking errors.

Chapter 7 focuses on the subjective evaluation of the predictive EDAS, investigating factors influencing user acceptance. The chapter then details the experimental procedure, formulates hypotheses, and presents comprehensive results from both measurement and structural model analyses. The chapter concludes with a discussion and summary of the findings.

Finally, Chapter 8 concludes the dissertation by summarizing the main contributions, discussing current limitations, outlining potential directions for future work, and presenting the overall conclusions drawn from the research.

1.5 Publications

Several research papers have been published during the course of the doctoral studies. A chronological list of these publications is given below.

[CPG+20] **S.K. Chada**, A. Purbai, D. Görges, A. Ebert and R. Teutsch. Ecological Adaptive Cruise Control for Urban Environments using SPaT Information. *In Proceedings of the IEEE Vehicle Power and Propulsion Conference (VPPC)*, pages 1-6, Gijon,

Spain, 2020.

[CGET21] **S.K. Chada**, D. Görges, A. Ebert and R. Teutsch. A driver-in-the-loop co-simulation framework for testing predictive EDAS for commercial vehicles in urban environments. *In Proceedings of the 6th International Commercial Vehicle Technology Symposium 2020/2021*, pages 107-118, Kaiserslautern, Germany, 2021.

[CTG+21] **S.K. Chada**, J. M. Thomas, D. Görges, A. Ebert and R. Teutsch. Ecological Adaptive Cruise Control for City Buses based on Hybrid Model Predictive Control using PnG and Traffic Light Information. *In Proceedings of the IEEE Vehicle Power and Propulsion Conference (VPPC)*, pages 1-7, Gijon, Spain, 2021.

[CKR+22] **S.K. Chada**, M. Kunz, Y. Ranker, R. Teutsch, D. Görges, A. Ebert and K. Mahjoub. Kompakter 6-DoF-Fahrsimulator für flexible Forschungsanwendungen. *In Proceedings of the 7th Commercial Vehicle Technology Symposium*, pages 15-35, Kaiserslautern, Germany, 2022.

[SCG22] G. Sundaram, **S.K. Chada**, and D. Görges. A novel approach to classify and replicate human drivers using model predictive control. *In Proceedings of the 7th Commercial Vehicle Technology Symposium*, pages 36-51, Kaiserslautern, Germany, 2022.

[CGE+22] **S.K. Chada**, D. Görges, A. Ebert, R. Teutsch and C. G. Min. Learning-based driver behavior modeling and delay compensation to improve the efficiency of an eco-driving assistance system. *In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 415-422, Prague, Czech Republic, 2022.

[CGET23] **S.K. Chada**, D. Görges, A. Ebert and R. Teutsch. Deep learning-based vehicle speed prediction for ecological adaptive cruise control in urban and highway scenarios. *In Proceedings of the 22nd IFAC World Congress*, pages 1107-1114, Yokohama, Japan, 2023.

[CGE+23] **S.K. Chada**, D. Görges, A. Ebert, R. Teutsch and S. P. Subramanya. Evaluation of the Driving Performance and User Acceptance of a Predictive Eco-Driving Assistance System. *Transportation Research Part C: Emerging Technologies*, 153:104193, 2023.

2 Theoretical Background

This chapter provides a comprehensive overview of classical control methodologies, including Pontryagin's minimum principle, dynamic programming and model predictive control, for solving the optimal control problem (OCP) in both continuous-time and discrete-time control systems. Additionally, the fundamentals of machine learning techniques, such as dimensionality reduction, clustering, classification and regression, are presented in this chapter. The chapter concludes with a review of state-of-the-art user acceptance models, specifically the Technology Acceptance Model (TAM) and the Theory of Planned Behavior (TPB).

2.1 Fundamentals of Control Methods

2.1.1 Optimal Control Problem

The optimal control problem (OCP) is a mathematical framework used to determine the optimal trajectories of control and state variables for a given dynamical system over a specific time period. The goal is to minimize a predefined objective or cost function while satisfying the dynamic constraints. Formally, the fundamental continuous-time constrained OCP with terminal constraints can be expressed as

$$\min_{u(t)} \; T(x(t_\mathrm{f}), t_\mathrm{f}) + \int_{t_0}^{t_\mathrm{f}} f(x(t), u(t), t)dt \tag{2.1a}$$

$$\text{s.t.} \quad \dot{x}(t) = \phi(x(t), u(t), t) \tag{2.1b}$$

$$x(t_0) = x_0 \tag{2.1c}$$

$$g(x(t), u(t), t) \leq 0 \tag{2.1d}$$

$$h_\mathrm{f}(x(t_\mathrm{f}), t_\mathrm{f}) = 0 \tag{2.1e}$$

In (2.1a), $f(x(t), u(t), t)$ denotes the running cost minimized over the initial time $t = t_0$ and a terminal time $t = t_\mathrm{f}$, while $T(x(t_\mathrm{f}), t_\mathrm{f})$ is a terminal cost penalizing the final state. The state vector $x(t) \in \mathcal{X} \subseteq \mathbb{R}^n$ and the input vector $u(t) \in \mathcal{U} \subseteq \mathbb{R}^q$ must fulfill the state equation (2.1b) with $\phi : \mathbb{R}^n \times \mathbb{R}^q \times \mathbb{R} \to \mathbb{R}^n$. Moreover, the initial state condition is defined by (2.1c), the inequality constraints with $g : \mathbb{R}^n \times \mathbb{R}^q \times \mathbb{R} \to \mathbb{R}^w$ are enforced through (2.1d) and the terminal constraints $h_\mathrm{f} : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^{l_\mathrm{f}}$ must be satisfied at t_f as stated in (2.1e).

Numerical methods for solving the OCP are divided into two classes: indirect and direct approaches. Indirect methods are based on the first-order necessary conditions derived from Pontryagin's minimum principle (PMP), resulting in a boundary value problem (BVP) that is solved using numerical techniques like shooting methods. On the other hand, in the direct methods the OCP is transformed into a constrained optimization problem by discretizing the state and control variables, making them large-scale constrained static problems with robust solvers like IPOPT. To solve the OCPs recursively, methods such as dynamic programming (DP) and model predictive control (MPC) are employed. The detailed methodology of the aforementioned methods is addressed in the next section.

2.1.2 Pontryagin's Minimum Principle

Pontryagin's minimum principle (PMP) is used in optimal control theory to provide necessary conditions for an optimal control trajectory. To solve the constrained OCP (2.1) in an analytical way, a Hamiltonian function is defined by combining the objective function f and the system dynamics (2.1b) using a Lagrange multiplier vector $\lambda \in \mathbb{R}^{n_x}$, referred to as the costate variable. Additionally, system constraints $g(x(t), u(t), t)$ are enforced through the multiplier μ, leading to the augmented Hamiltonian:

$$H(x(t), u(t), \lambda(t), \mu(t), t) = f(x(t), u(t), t) + \lambda(t)^T \phi(x(t), u(t), t) + \mu(t)^T g(x(t), u(t), t) \tag{2.2}$$

The PMP establishes the necessary condition for optimality (2.3), which states that the optimal control trajectory $u^*(t)$ must minimize the augmented Hamiltonian H over the set of admissible controls $\mathcal{U}$ at all times $t \in [t_0, t_f]$.

$$u^*(t) = \underset{u(t) \in \mathcal{U}}{\operatorname{argmin}} H(x^*(t), u(t), \lambda^*(t), \mu^*(t), t) \tag{2.3}$$

The state equation is given by

$$\dot{x}^*(t) = \nabla_\lambda H(x^*(t), u^*(t), \lambda^*(t), \mu^*(t), t) = \phi(x^*(t), u^*(t), t) \tag{2.4}$$

where $\nabla_\lambda H$ denotes gradient of the Hamiltonian with respect to the costate vector λ.

The costate equation is formulated as

$$\begin{aligned} \dot{\lambda}^*(t) \;=\; & -\nabla_x H(x^*(t), u^*(t), \lambda^*(t), \mu^*(t), t) = -\nabla_x f(x^*(t), u^*(t), t) - \\ & J_{\phi x}(x^*(t), u^*(t), t)^\intercal \lambda^*(t) - J_{gx}(x^*(t), u^*(t), t)^\intercal \mu^*(t) \end{aligned} \tag{2.5}$$

where $J_{\phi x}$ and J_{gx} denote the Jacobians of the system dynamics ϕ and the path constraints g with respect to the state vector x, respectively.

The coupling equation is given by

$$\begin{aligned} \nabla_u H(x^*(t), u^*(t), \lambda^*(t), \mu^*(t), t) \;=\; & \nabla_u f(x^*(t), u^*(t), t) + J_{\phi u}(x^*(t), u^*(t), t)^\intercal \lambda^*(t) + \\ & J_{gu}(x^*(t), u^*(t), t)^\intercal \mu^*(t) = 0 \end{aligned} \tag{2.6}$$

Furthermore, the product of the inequality constraints and the associated multiplier μ need to satisfy

$$\mu_i^*(t)\ g_i(x^*(t), u^*(t), t) = 0 \quad \forall\ i \in 1, ..., w \tag{2.7}$$

where w denotes the total number of inequality constraints and the multipliers must satisfy $\mu(t) \geq 0$.

The transversality condition on the terminal state is given by

$$\left(\nabla_{x(t_\mathrm{f})} T(x^*(t_\mathrm{f}^*), t_\mathrm{f}^*) + J_{h_\mathrm{f} x(t_\mathrm{f})}(x^*(t_\mathrm{f}^*), t_\mathrm{f}^*)^\intercal - \lambda^*(t_\mathrm{f}^*)\right)^\intercal \delta x_\mathrm{f} = 0 \tag{2.8}$$

The transversality condition on the terminal time is described as

$$\left(H(x^*(t_\mathrm{f}^*), u^*(t_\mathrm{f}^*), \lambda^*(t_\mathrm{f}^*), t_\mathrm{f}^*) + \frac{\partial T}{\partial t}\,(x^*(t_\mathrm{f}^*), \lambda_\mathrm{f}^*) + J_{h_\mathrm{f} t_\mathrm{f}}(x^*(t_\mathrm{f}^*), t_\mathrm{f}^*)^\intercal \lambda_\mathrm{f}^*\right) \delta t_\mathrm{f} = 0 \tag{2.9}$$

The initial condition corresponds to (2.1d) and the end condition is given by

$$h_\mathrm{f}(x^*(t_\mathrm{f}^*), t_\mathrm{f}^*) = 0 \tag{2.10}$$

Together, (2.8), (2.9), (2.1d) and (2.10) define the boundary conditions of the resulting two-point boundary value problem.

Moreover, if the optimal control trajectory $u^*(t)$ leads to a minimum for the problem (2.1), then a second-order necessary condition for a minimum denoted as Legendre condition must be satisfied.

$$\nabla_{uu}^2 H(x^*(t), u^*(t), \lambda^*(t), \mu^*(t), t) \geq 0 \tag{2.11a}$$

$$\text{s.t.} \quad J_{gu}(x^*(t), u^*(t), t)^\intercal \delta u(t) = 0 \tag{2.11b}$$

where $\nabla_{uu}^2 H$ is the Hessian matrix and ≥ 0 signifies that this matrix must be positive semi-definite.

Although PMP is well-suited for solving continuous-time OCPs, its applicability is limited by several challenges. One key drawback is that deriving an analytical solution using PMP is only feasible for specific problem classes, restricting its general usability. Additionally, PMP-based methods require prior knowledge of the optimal solution's structure, which is often unavailable before solving the OCP. A major challenge also arises in handling inequality constraints, particularly in systems with highly nonlinear dynamics, where enforcing constraints leads to complex optimal trajectories. Furthermore, the accuracy and stability of the computed solution are highly sensitive to the initial guess of costate variables, making convergence more difficult. These limitations highlight the constraints of PMP-based approaches and emphasize the need for alternative direct methods, such as dynamic programming or model predictive control, which offer greater flexibility and robustness in handling constrained and nonlinear optimization problems.

2.1.3 Dynamic Programming

The Dynamic Programming (DP) approach is based on the Bellman's principle of optimality [Ber12], which states that an optimal policy possesses the property that regardless of the initial state and decision, the remaining decisions must constitute an optimal policy with respect to the resulting state from the first decision. In this context, the term *decision* refers to the selection of a control input at a specific time, while *policy* denotes the complete sequence or function of control inputs over the time horizon. By employing dynamic programming, a complex dynamic optimization problem can be systematically decomposed into a sequence of smaller, structurally similar subproblems. The overall solution is then constructed efficiently by combining the solutions of these subproblems. One of the key advantages of dynamic programming is its ability to guarantee a globally optimal solution under the given constraints [EEG13]. The finite-horizon OCP in the context of dynamic programming is mathematically formulated as

$$J^* = \min_{u_k} \left(\zeta_N(x_N) + \sum_{k=0}^{N-1} \zeta_k(x_k, u_k) \right) \tag{2.12a}$$

$$\text{s.t.} \qquad x_{k+1} = f(x_k, u_k) \tag{2.12b}$$

$$x_k \in [x_{\text{lb},k}, x_{\text{ub},k}] \tag{2.12c}$$

$$u_k \in [u_{\text{lb},k}, u_{\text{ub},k}] \tag{2.12d}$$

$$x_0 = x_{\text{start}} \tag{2.12e}$$

$$x_N \in [x_{\text{lb},N}, x_{\text{ub},N}] \tag{2.12f}$$

The objective of the optimization problem (2.12) is to determine the sequence of control inputs u_k that minimizes the total cost J^*. This cost comprises the terminal cost $\zeta_N(x_N)$ at the final state and the sum of the stage costs $\zeta_k(x_k, u_k)$ over the prediction horizon from $k = 0$ to $N - 1$. The optimization is subject to several constraints that define the feasible solution space: the system dynamics (2.12b), state constraints (2.12c) and control input constraints (2.12d). The corresponding upper and lower bounds for these constraints are illustrated in Figure 2.1. Additionally, the initial condition is specified by (2.12e) and the final state must lie within a predefined terminal region, as defined in (2.12f).

In this formulation, the discrete-time state variable x_k, with $k = 0, 1, ..., N$, is discretized in the state space with a resolution of Δx. The m-th discretized state at time step k, where $m = 0, 1, ..., M$, is denoted by x_k^m. Similarly, the control input u_k defined for $k = 0, 1, ..., N - 1$, is also discretized with a resolution of Δu. The complete set of discretized states and control inputs at each time step can therefore be represented as $X_k = (x_k^0, x_k^1, ..., x_k^m, ..., x_k^M)$ and $U_k = (u_k^0, u_k^1, ..., u_k^l, ..., u_k^L)$, respectively. This discretization scheme is visually summarized in Figure 2.1.

Dynamic programming can be decomposed into three distinct stages [EEG13], as outlined below:

Stage I. Initialization of the Cost-to-Go Function

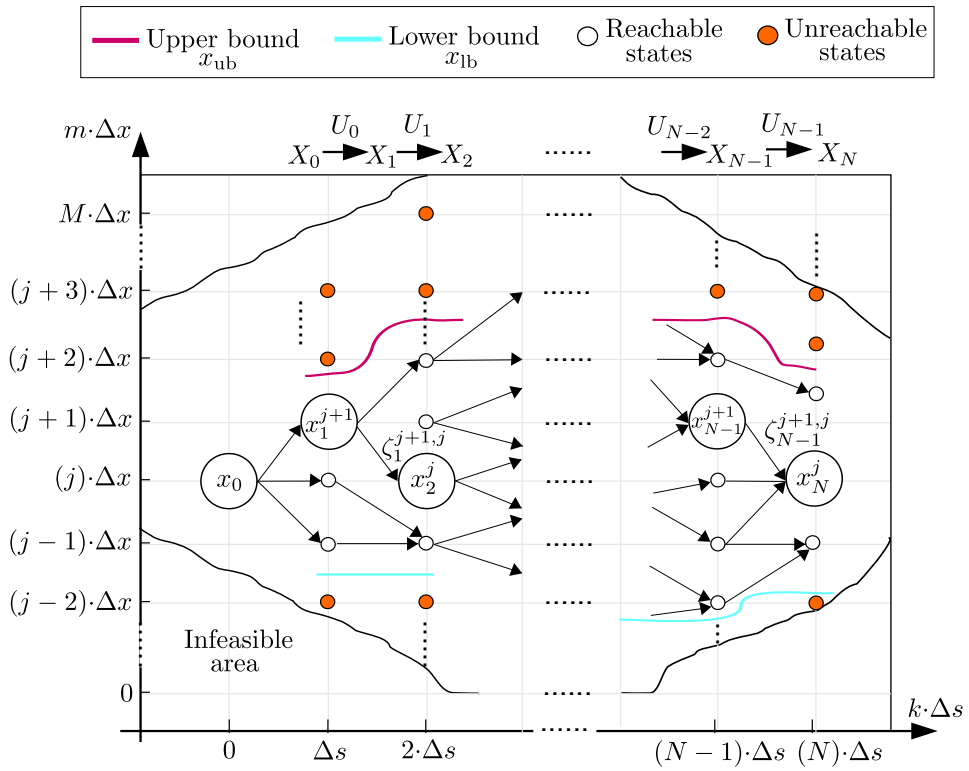


Figure 2.1: Schematic Representation of the Fundamentals of Dynamic Programming

This stage involves defining the initial conditions for the cost-to-go function $J_N(x_N^j)$ at the final stage N, as given by

$$J_N(x_N^j) = \begin{cases} \zeta_N(x_N^j), & x_N^j \in [x_{\text{lb},N}, x_{\text{ub},N}] \\ \infty, & x_N^j \notin [x_{\text{lb},N}, x_{\text{ub},N}] \end{cases} \tag{2.13}$$

This function represents the cumulative cost associated with reaching the final state from a given point in the state space. A finite terminal cost $\zeta_N(x_N^j)$ is assigned if the final state x_N^j lies within the admissible bounds $[x_{\text{lb},N}, x_{\text{ub},N}]$. Conversely, if x_N^j falls outside this range, the cost is set to infinity, indicating that such states are either infeasible or highly undesirable.

Stage II. Recursive Iteration using Principle of Optimality

This stage represents the core recursive process in Dynamic Programming (DP), guided by the principle of optimality. At each stage k, the cost-to-go function $J_k(x_k^i)$ is computed by minimizing over the control input u_k, as defined by

$$J_k(x_k^i) = \min_{u_k} \{\zeta_k^{i,j}(x_k^i, u_k^l) + J_{k+1}(f(x_k^i, u_k^l))\} \tag{2.14}$$

Here, the transition from the current state x_k^i to the next state x_{k+1}^j incurs an immediate cost $\zeta_k^{i,j}(x_k^i, u_k^l)$, while $J_{k+1}(f(x_k^i, u_k^l))$ represents the future cost, recursively computed

from the next stage onward. By iteratively solving this equation in a backward manner, i.e. from the terminal stage N to the initial stage, DP identifies the optimal control policy.

Stage III. Forward Iteration or Decision Stage

The forward iteration, also known as the decision stage, involves determining the optimal control policy u_k^* at each stage, using the previously computed cost-to-go functions $J_k(x_k^i)$ from the recursive backward pass in (2.14). The optimal control input is obtained as

$$u_k^*(x_k) = \underset{u}{\operatorname{argmin}} \underbrace{\{\zeta_k^{i,j}(x_k^i, u_k^l) + J_{k+1}(f(x_k^i, u_k^l))\}}_{J_k(x_k^i)} \tag{2.15}$$

Once the optimal control inputs u_k^* are determined, they are applied to the state transition function (as defined in (2.12a)) to generate the corresponding optimal sequence of state variables x_k^*.

By combining these three stages, namely, initialization, recursive iteration based on the principle of optimality and forward iteration, DP provides a structured and efficient method for solving multistage decision problems. Although DP provides a powerful framework for solving OCPs and guarantees a global optimal solution, it also has several limitations. One major drawback is the curse of dimensionality, particularly in high-dimensional state spaces, where the computational complexity grows exponentially with the number of state variables, making it impractical for large-scale systems. Additionally, the accuracy of the DP solution heavily depends on the discretization grid and step size. A coarse discretization may introduce approximation errors, leading to suboptimal results, while a fine-grained discretization significantly increases computational cost. Therefore, achieving a balanced trade-off between accuracy and computational efficiency is crucial. These challenges make real-time implementation of DP difficult in practical applications.

2.1.4 Model Predictive Control

Model Predictive Control (MPC) is a widely popular advanced control strategy based on optimal control principles for nonlinear constrained systems. It is also referred to as receding horizon control due to its unique operational approach, where the prediction horizon continuously shifts forward in time with each control iteration. This allows MPC to dynamically incorporate new measurements and updated system states, enabling it to adapt to changing conditions and disturbances.

Initially, MPC was predominantly applied in process control systems [MGP88], particularly those with slow dynamics, as solving real-time optimization problems required significant computational resources. However, recent advancements in computer hardware have drastically reduced computational time, making MPC feasible for applications with fast dynamics. This has enabled the application of MPC in various domains , including

vehicular systems [RAG+10, HCTK12], power electronics [VLF+14] and mechatronic systems [SZ21, SABA21].

The fundamental concept of MPC is to predict the future states of a dynamic system over a given prediction horizon using a mathematical model. The MPC algorithm utilizes these predictions to compute optimal control inputs that achieve the desired control objective while respecting system constraints. The concept of the model predictive control is illustrated in Figure 2.2, which consists of the following steps:

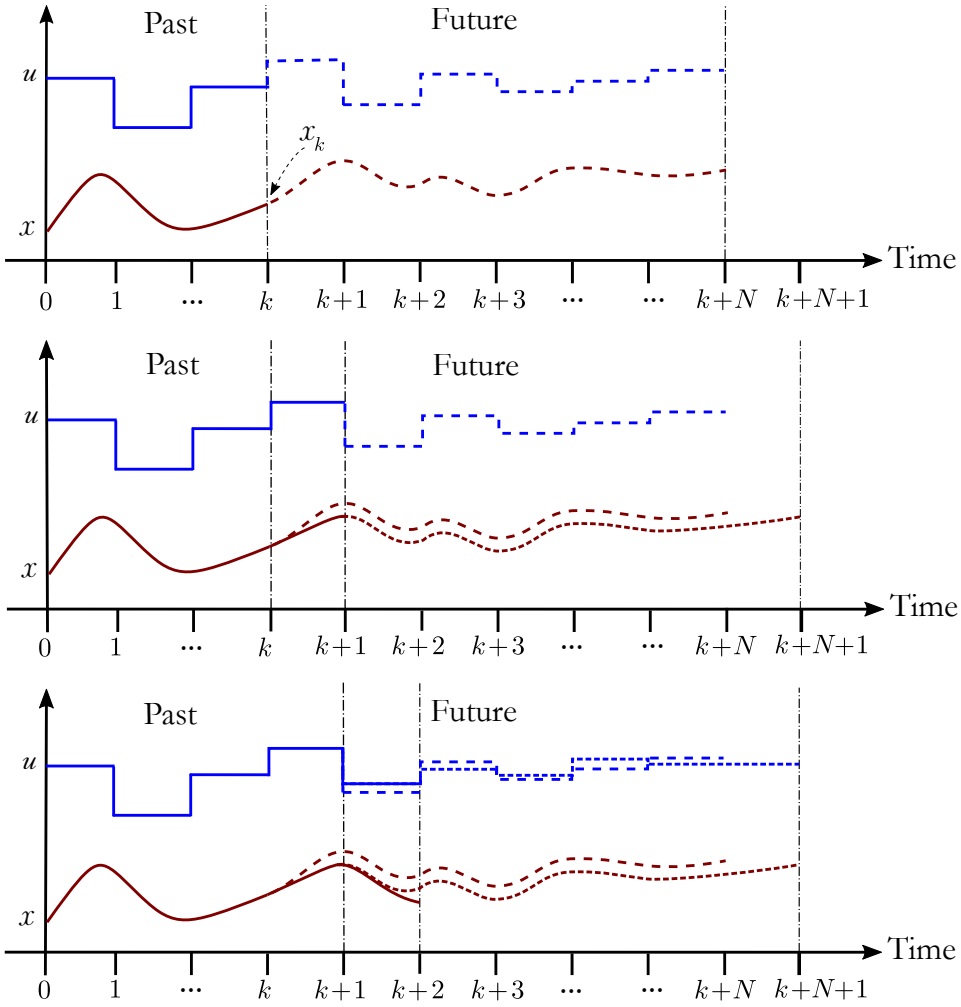


Figure 2.2: Concept of Model Predictive Control

1. Measure the current state $x(k)$.
2. Compute the optimal control sequence $u^*(k+1)$ by predicting future states over the prediction horizon and solving an optimization problem based on a given cost function.
3. Apply only the first control input from the computed sequence and proceed to the

next step at the following sampling instant.

4. Repeat the process at each time step by re-solving the optimization problem using updated state measurements and system outputs.

The general mathematical formulation of an MPC optimization problem is given as

$$J = \min_{U_k} \sum_{k=0}^{N-1} \underbrace{\pi(x_k, u_k, w_k, t_k)}_{\text{Stage cost}} + \underbrace{T(x_{k+N})}_{\text{Terminal cost}} \tag{2.16a}$$

$$\text{s.t.} \qquad x_{k+1} = f(x_k, u_k, w_k, t_k) \tag{2.16b}$$

$$g(x_k, u_k, t_k) \leq 1 \tag{2.16c}$$

$$h(x_k, u_k, t_k) = 1 \tag{2.16d}$$

where, J represents the cost function, which consists of the sum of the stage cost π and the terminal cost T and is minimized over the prediction horizon N. Moreover, x_k denotes the system states, u_k represents the control inputs, w_k accounts for external disturbances and t_k is the time at which the control input is applied. The goal of MPC is to solve the optimization problem (2.16a) to find the optimal control sequence U_k. The system dynamics are governed by (2.16b), where $f(\cdot)$ represents the state transition function, which maps the current state and control input to the next state x_{k+1} while considering external disturbances. The inequality constraints $g(\cdot)$ and equality constraints $h(\cdot)$ are given in (2.16c) and (2.16d), respectively.

Convex Functions and Quadratic Programming

Definition 2.1.1. A set $\mathcal{X}$ is convex if the line segment connecting any two arbitary points $x_1, x_2 \in \mathcal{X}$ remains entirely within the set and satisfies the condition

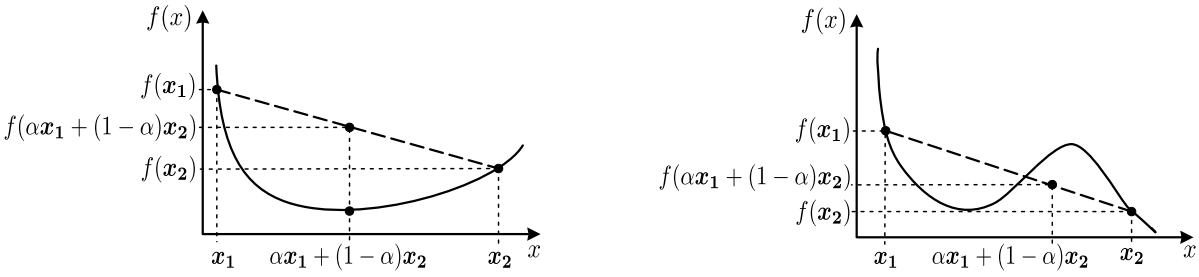
$$(\alpha x_1 + (1-\alpha)x_2) \in \mathbb{X} \quad \forall\ x_1, x_2 \in \mathbb{X} \text{ und } \alpha \in [0,1] \tag{2.17}$$

Definition 2.1.2. A function $f : \mathcal{X} \subseteq \mathbb{R}$ defined on a convex set $\mathcal{X}$ is convex if the secant connecting two arbitrary points $(x_1, f(x_1))$ and $(x_2, f(x_2))$ lies on or above the graph of f. This is mathematically represented as

$$f(\alpha x_1 + (1-\alpha)x_2) \leq \alpha f(x_2) + (1-\alpha) f(x_2)\ ; \quad \forall x_2, x_2 \in \mathbb{X}\ ; \quad \forall \alpha \in [0,1] \tag{2.18}$$

Figures 2.3a and 2.3b illustrate examples of convex and non-convex functions, respectively. Convex functions are characterized by a single global minimum, ensuring that any local minimum is also the global optimum. In contrast, non-convex functions may have multiple local minima, making optimization more complex and increasing the likelihood of converging to suboptimal solutions.

Formulating an optimization problem in Quadratic Programming (QP) form offers significant advantages, particularly when dealing with convex functions (Figure 2.3a) and



(a) Convex function (b) Non-convex function

Figure 2.3: Examples of the convex and non-convex functions

linear constraints. Convex QP problems ensure that solutions are close to the global optimum, providing robustness and reliability. Additionally, efficient solvers such as interior-point and active-set methods make QP problems computationally feasible and numerically stable. The ability to accommodate linear constraints makes QP particularly suitable for applications such as Model Predictive Control (MPC). A convex QP problem can be formulated as an optimization problem of the form:

$$\min_{x} \quad \frac{1}{2}x^T H x + f^T x \tag{2.19a}$$

$$\text{s.t.} \quad A_{\text{eq}}x = b_{\text{eq}} \tag{2.19b}$$

$$A_{\text{ieq}}x \leq b_{\text{ieq}} \tag{2.19c}$$

where $A_{\text{eq}} \in \mathbb{R}^{m\times n}$, $b_{\text{eq}} \in \mathbb{R}^m$, $A_{\text{ieq}} \in \mathbb{R}^{q\times n}$, $b_{\text{ieq}} \in \mathbb{R}^q$ and $H \in \mathbb{R}^{n\times n}$. The matrix H is symmetric and positive semidefinite ($H = H^T \succeq 0$), ensuring convexity of the problem. This convex formulation guarantees that minimizing the objective function leads to a global minimum, making it computationally efficient and well-suited for various optimization applications.

If the optimization problem involves a quadratic objective function with both quadratic and affine inequality constraints, it can be formulated as a Quadratically Constrained Quadratic Program (QCQP), expressed as

$$\min_{x} \quad \frac{1}{2}x^T H x + f^T x \tag{2.20a}$$

$$\text{s.t.} \quad (2.19\text{a}), (2.19\text{c})$$

$$x^T Q_i x + c_i^T x + d_i \leq 0 \tag{2.20b}$$

where H is a symmetric positive semi-definite matrix. In this formulation, the quadratic program seeks to minimize a convex objective function over a feasible region defined by both affine and quadratic constraints.

2.2 Fundamentals of Machine Learning

Machine learning is a subfield of artificial intelligence that uses statistical methods and algorithms to enable machines to identify patterns, make predictions and improve their performance without explicit human instructions. Machine learning algorithms can be broadly categorized into unsupervised and supervised learning, as depicted in Figure 2.4.

The unsupervised learning primarily deals with unlabeled data, where the algorithm aims to discover hidden patterns and relationships without explicit guidance or labeled information. Common approaches in unsupervised learning include dimensionality reduction, which reduces the number of variables while preserving important information and clustering, which groups similar data points together. Unsupervised learning plays a crucial role in exploratory data analysis, data preprocessing and feature extraction and it has applications in various fields such as customer segmentation, image recognition and anomaly detection.

The supervised learning algorithms on the other hand, learn from labeled data that comprises of input features and corresponding desired outputs. The goal is to learn a mapping function that can accurately predict the output for unseen inputs. Unlike unsupervised learning, the model is trained on a labeled dataset, where the algorithm iteratively adjusts its parameters to minimize the difference between predicted and actual outputs. Supervised learning encompasses two main tasks: classification, which predicts categorical outputs and regression, which predicts continuous outputs. Supervised learning has widespread applications in various domains, including image recognition, natural language processing and medical diagnosis.

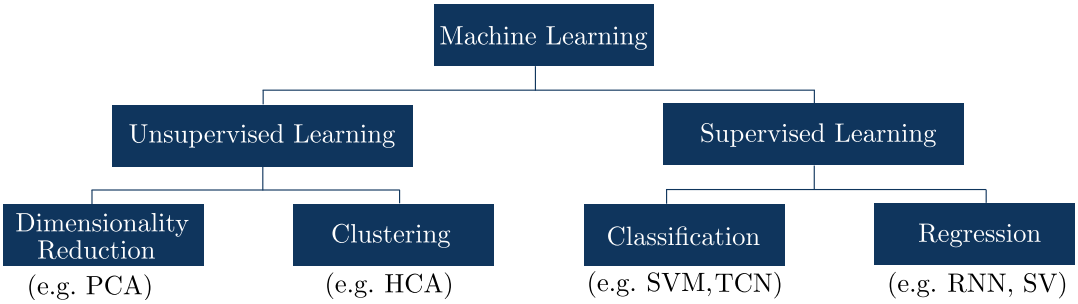


Figure 2.4: Machine Learning Methods

In the following subsections the dimensionality reduction, clustering, classification and regression methods are discussed in detail.

2.2.1 Dimensionality Reduction

Dimensionality reduction is a machine learning technique used to simplify data by reducing the number of input features or variables while preserving its most important

characteristics. This is achieved by identifying and removing redundant or irrelevant features, to create a model in a low-dimensional space. Dimensionality reduction techniques are particularly useful for high-dimensional datasets, where the number of features can be very large, making it computationally expensive to train and evaluate machine learning models. Popular methods include principal component analysis (PCA), linear discriminant analysis and t-distributed stochastic neighbor embedding (t-SNE).

Principal Component Analysis (PCA)

In this dissertation, the PCA is chosen as a dimensionality reduction technique as it presents several advantages. Firstly, PCA is computationally less expensive, making it more suitable for large datasets. Secondly, PCA effectively preserves the global structure of the data, capturing the relationships between data points across the entire dataset. In contrast, t-SNE focuses on preserving local neighborhoods, which can sometimes distort the overall relationships in the data. Finally, PCA is simpler to interpret, as the principal components are directly derived from the original variables and represent directions of maximum variance. The pseudo-code for PCA is described in Algorithm 2.1, which aims to identify the principal components that are orthogonal linear combinations of the original variables capturing maximum variance within a dataset. By projecting

Algorithm 2.1 PCA: Identify important variables and reduce data dimensionality

Require: Input variables stored in matrix H.

1: Normalize the data matrix H into the normalized matrix X:

$$x_{ab} = \frac{h_{ab} - \mu_b}{\sigma_b}$$

where a are rows, b are columns, μ_b is the mean and σ_b is the standard deviation.

2: Compute the covariance matrix: $Z_x = X^T X = \frac{1}{N} \sum_{a=1}^{N} (x_a - \mu)(x_a - \mu)^T$
where N denotes the total number of data points in the dataset

3: Perform eigendecomposition on Z_x to obtain the corresponding eigenvalues and eigenvectors:

$$Z_Y = P Z_X P^T$$

where Z_Y is a diagonal matrix with eigenvalues and P is the matrix of eigenvectors.

4: Sort the eigenvalues in descending order.

5: Retain only the principal components that satisfy the predefined criteria.

the data onto these principal components, PCA reduces the number of variables while preserving the most important information, simplifying analysis and visualization. This is achieved by finding the eigenvectors of the covariance matrix, which represent the directions of greatest variability. The corresponding eigenvalues indicate the amount of variance explained by each principal component, allowing for the selection of a subset that retains a desired level of information.

2.2.2 Clustering Methods

Clustering, a fundamental technique in unsupervised machine learning, involves grouping unlabeled data points into clusters based on their inherent similarities. By analyzing the relationships between data points, clustering algorithms identify natural groupings where objects within a cluster are more similar to each other than to those in different clusters. Clustering methods can be broadly categorized into four categories: centroid-based approaches (such as k-means clustering), hierarchical methods (such as hierarchical clustering analysis), density-based methods (such as DBSCAN) and distribution-based approaches, (such as Gaussian mixture models).

Hierarchical Clustering Analysis (HCA)

In this study, HCA is selected as the clustering algorithm due to its distinct advantages. First, it provides a hierarchical structure of clusters, represented as a dendrogram, allowing for a multi-resolution view of data structure and relationships between clusters at various granularities. This is particularly valuable when exploring datasets with potential nested structures. Second, unlike k-means clustering, HCA does not require predefining the number of clusters, offering flexibility in exploring different cluster solutions through the dendrogram. Additionally, hierarchical methods can accommodate clusters of diverse shapes and sizes, whereas k-means tends to favor spherical clusters. Finally, the dendrogram provides an interpretable visualization of the clustering process, facilitating a clearer understanding of the results. The pseudocode of HCA is presented in Algorithm 2.2.

Algorithm 2.2 HCA: Clustering driving styles into driver types

Require: Matrix G containing n principal components.

1: **while** the number of clusters > 1 **do**
2: Merge the two clusters with smallest squared distance.
3: Update the Euclidean distance matrix.
4: Save current smallest squared distance and its respective cluster ID in the stack.
5: Reduce the number of clusters by half.
6: Update the dendrogram.
7: **end while**
8: **return** Identify optimal number of clusters based on the generated dendrogram.

2.2.3 Classification Methods

Classification methods are a fundamental aspect of supervised learning in machine learning, where the goal is to assign data points to predefined categories or classes. This process involves learning a mapping function from input features to discrete output labels

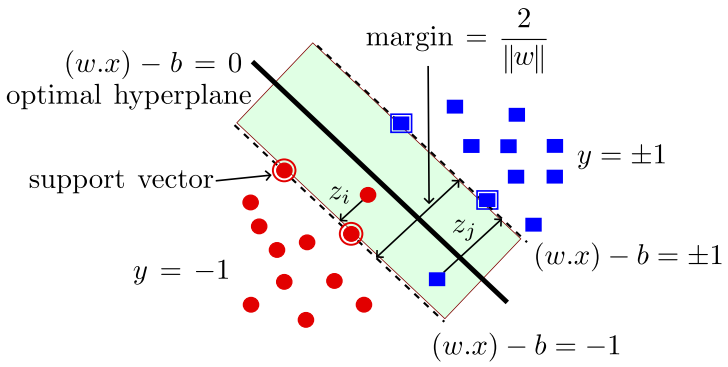


Figure 2.5: Illustration of SVM [Do20]

using a labeled training dataset, enabling accurate prediction and classification of new instances. Common classification algorithms include logistic regression, decision trees, support vector machines, naive Bayes, k-nearest neighbors and neural networks. In this study, the support vector machine (SVM) and the temporal convolutional network (TCN), a type of neural network, are examined for classification tasks.

Support Vector Machine (SVM)

The SVM is a powerful supervised learning algorithm used for classification tasks, capable of handling both linear and nonlinear decision boundaries through the use of various kernel functions. The fundamental idea behind SVM is to construct an optimal hyperplane that effectively separates data points into distinct categories. The objective is to maximize the margin between different classes, ensuring better generalization to unseen data. Given a dataset x with N training samples, SVM seeks to find the optimal hyperplane that separates the data [BSL+18], formulated as

$$\boldsymbol{x} = (\boldsymbol{x}_n, y_n), \boldsymbol{x}_n \in \mathbb{R}^g, y_n \in -1, 1, \ \forall\ n = 1, ..., N \tag{2.21}$$

where $\boldsymbol{x}_n$ represents the g-dimensional input vector and y_n denotes its corresponding class label. The hyperplane that separates the two classes is defined by

$$\boldsymbol{w}.\boldsymbol{x} + b = 0 \tag{2.22}$$

where $\boldsymbol{w}$ is the weight vector and b is the bias term. For a linearly separable dataset, the margins are defined by

$$\boldsymbol{w}.\boldsymbol{x} + b = 1 \quad \text{and} \quad \boldsymbol{w}.\boldsymbol{x} + b = -1 \tag{2.23}$$

The distance between these margins is given by $\frac{2}{\|\boldsymbol{w}\|}$, as illustrated in Figure 2.5. However, in real-world applications, data is often not perfectly separable using a linear

decision boundary. To address this, a regularization parameter C is introduced to balance the trade-off between maximizing the margin and minimizing classification errors [BSL+18]. The optimization objective is then formulated as

$$C\sum_{n=1}^{n}\zeta_n + \frac{1}{2}\|\boldsymbol{w}\|^2 \tag{2.24}$$

subject to $t_n y(\boldsymbol{x}_n) \geq 1 - \zeta_n$, where t_n represents the target labels and ζ_n are slack variables. To further simplify the optimization problem (2.24), the Lagrangian dual formulation is introduced using the dual multipliers β, leading to

$$\max \sum_{n=1}^{N}\beta_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\beta_n\beta_m y_n y_m \langle \boldsymbol{x}_n, \boldsymbol{x}_m\rangle \tag{2.25a}$$

$$\text{s.t.} \qquad 0 \leq \beta_i \leq C \quad \forall\ i = 1, ..., n \tag{2.25b}$$

$$\sum_{i=1}^{n}\beta_i y_i = 0 \tag{2.25c}$$

To enhance SVM's ability to handle nonlinear data, a kernel function $k(\boldsymbol{x}_n, \boldsymbol{x}_m)$ is introduced, mapping the input data into a higher-dimensional space where it becomes linearly separable. This transforms the optimization problem (2.25a) into

$$\max \sum_{n=1}^{N}\beta_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\beta_n\beta_m y_n y_m k(\boldsymbol{x}_n, \boldsymbol{x}_m) \tag{2.26}$$

Various kernel functions exist, including linear, polynomial, radial basis function (RBF) and sigmoid. This study primarily focuses on the RBF kernel, which is defined as

$$k(\boldsymbol{x}_n, \boldsymbol{x}_m) = \exp(-\gamma\|\boldsymbol{x}_n - \boldsymbol{x}_m\|^2) \tag{2.27}$$

The RBF kernel is particularly effective in capturing complex relationships within data, making it a widely used choice in SVM-based classification tasks.

Neural Networks for Classification

Neural networks have emerged as a powerful alternative to traditional classification methods, offering several key advantages. Firstly, they excel in learning complex, nonlinear relationships between input features and class labels, making them particularly effective for high-dimensional and intricate datasets where linear models often struggle. Secondly, neural networks can automatically extract and learn relevant features from raw data, reducing the reliance on manual feature extraction. Thirdly, they demonstrate strong generalization capabilities, enabling high accuracy on unseen data by learning robust and abstract representations. Lastly, with the advancements in deep learning, neural networks can leverage large-scale datasets and high-performance computational

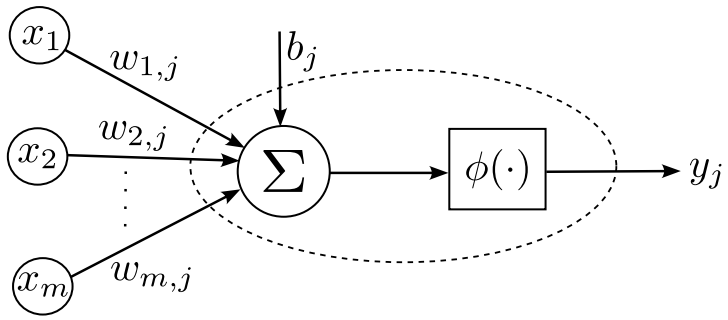


Figure 2.6: Illustration of a single neuron in a neural network

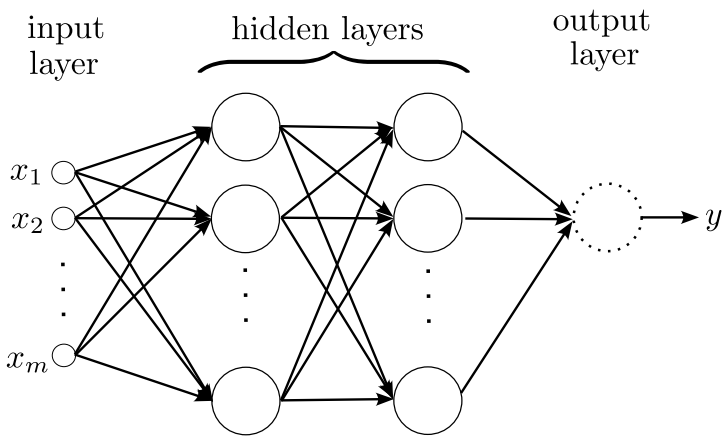


Figure 2.7: Illustration of a feed forward neural network

resources to achieve state-of-the-art results in various classification tasks, often outperforming conventional classification techniques.

A fundamental neural network unit, a single neuron, is illustrated in Figure 2.6. Consider a neuron j that receives m input signals $x_1, x_2, ..., x_m$, each associated with a corresponding weight $w_{1,j}, w_{2,j}, ..., w_{m,j}$, along with a bias term b_j. To introduce non-linearity into the model, an activation function $\phi(.)$ is applied to the weighted sum of the inputs. The output of neuron j is given by

$$y_j = \varphi\left(\sum_{i=1}^{m} w_{ij}x_i + b_j\right) \tag{2.28}$$

Common activation functions used in neural networks include the sigmoid function, the rectified linear unit (ReLU) and the hyperbolic tangent (Tanh). A feedforward neural network (FNN) represents the most fundamental type of artificial neural network, as illustrated in Figure 2.7. It maps input signals $x_1, x_2, ..., x_m$ to one or more output values $y$through multiple interconnected layers of neurons. In an FNN, data flows strictly in a forward direction, i.e. from the input layer, through one or more hidden layers, to the output layer. Each neuron in a given layer connects to neurons in the subsequent layer

through weighted connections and operates as described in (2.28).The output layer ultimately produces the final prediction, which can be utilized for classification or regression tasks.

FNNs learn through the backpropagation algorithm, where network parameters (weights w_{ij} and biases b_j) are iteratively adjusted using optimization techniques such as gradient descent. The network minimizes a loss function E, which quantifies the discrepancy between the predicted and true outputs, by updating weights according to:

$$w_{ij} \leftarrow w_{ij} - \eta \frac{\partial E}{\partial w_{ij}} \tag{2.29}$$

where η represents the learning rate that controls the rate at which the parameters are updated.

Although FNNs are effective for a wide range of tasks, they lack memory and are therefore unsuitable for sequential data processing. For tasks involving temporal dependencies, alternative architectures such as temporal convolutional networks (TCNs) or recurrent neural networks (RNNs) are often more appropriate.

Temporal Convolutional Networks (TCN)

A TCN is typically a 1-dimensional fully connected network with causal convolutions that can be effectively used for classification task by leveraging their ability to model sequential data for time series problems with long-range dependencies. However, with only the 1D fully connected network, the TCN is not able to compute efficiently. The causal convolutions with a very deep network or very large filter size are required to cover long historical size. To address this issue, the authors in [BKK18] proposed dilated convolutions to enable an exponentially large receptive field. In a 1D input sequence with $x \in \mathbb{R}^n$ and a filter $f : 0, ..., q-1 \rightarrow \mathbb{R}$, the dilated convolution R applied on element p of the sequence is given as

$$R(p) = (x *_d f)(p) = \sum_{i=0}^{q-1} f(i) \cdot x_{p-d \cdot i} \tag{2.30}$$

where $(p - d \cdot i)$ refers to the direction of the past, d is the dilation factor and q is the filter size.It is important to note that when $d = 1$, (2.30) reduces to an ordinary causal convolution. By adjusting the filter size q and the dilation factor d, the receptive field of TCN can be modified accordingly.

One of the key advantages of TCNs is their ability to perform convolutions in parallel, as the same filter is applied across all layers, significantly accelerating the training process. However, a notable drawback is their high memory consumption during the testing phase, as TCNs require storing the entire raw sequence, potentially leading to increased computational costs [BKK18].

2.2.4 Regression Methods

Regression models in machine learning aim to predict a continuous target variable based on one or more input features. These models learn a mapping between input and output variables by fitting a function to training data, typically by minimizing the difference between predicted and actual values. The function can take various forms, ranging from simple linear equations to complex nonlinear relationships. Common approaches include linear regression, polynomial regression and logistic regression. Additionally, neural networks can effectively perform regression for highly nonlinear data by mapping input features to continuous outputs. Regression techniques are widely used for time series forecasting, where future values are predicted based on past observations. Time series models can be broadly categorized as deterministic or stochastic.

Deterministic models, such as linear regression, polynomial regression, feedforward neural networks and recurrent neural networks, assume a fixed functional relationship between inputs and outputs and that the future values are entirely predictable based on past observations. These models are typically straightforward to implement and interpret. In contrast, stochastic models account for randomness in the data by incorporating probabilistic elements into their structure. These models assume that future values are influenced by both past observations and random disturbances. Examples include autoregressive (AR) models, moving average (MA) models and stochastic volatility (SV) models. These models are more complex but are particularly effective for time series with significant volatility.

In this study, the stochastic volatility (SV) model and recurrent neural networks (RNNs) are investigated for regression tasks.

Stochastic Volatility (SV) Model

Stochastic Volatility (SV) model is a widely popular method in financial mathematics, called Ornstein-Uhlenbeck process was used in this work [RSM20]. The Ornstein-Uhlenbeck stochastic differential equation is given by

$$dX_t = \gamma(\mu - X_t)\Delta t + \nu \Delta W_t \tag{2.31}$$

where, $\gamma > 0$ refers to the rate of mean reversion, μ refers to the long-term mean of the process, $\nu > 0$ can be explained as average magnitude used to define random fluctuations that can be modeled as Wiener process W.

To find a numerical solution to the Ornstein-Uhlenbeck stochastic differential equation, the authors in [RSM20] used Euler-Maruyama scheme resulting in a transformation of stochastic differential equations as

$$X_{t+1} = X_t + \gamma(\mu - X_t)\Delta t + \nu \Delta W_t \tag{2.32}$$

where $\Delta W_t \sim \mathcal{N}(0, \Delta t)$, the prior distribution of the mean μ is modeled using a normal distribution, the prior distribution for the remaining two parameters ν and γ are modeled using a uniform distribution.

Recurrent Neural Networks (RNNs)

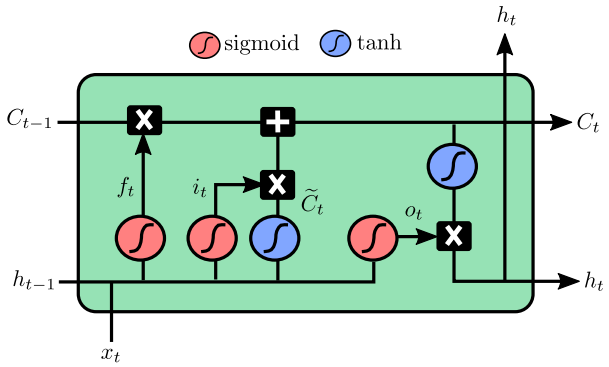
RNNs are popularly known in the category of deep neural networks as they are capable of using their internal state memory to process sequential or time series data. Two variants in the RNN architecture, namely, gated recurrent unit (GRU) and long short-term memory (LSTM) units have been investigated in this work. The internal cell structure of LSTM and GRU are illustrated in the Figure 2.8a and Figure 2.8b, respectively. Both variants use gating mechanism to solve the vanishing gradient problem present in the traditional RNNs [CGCB14].

The internal mechanisms of an LSTM (Figure 2.8a) consists of a cell state C_t and various gates such as a forget gate f_t, an input gate (i_t, $\widetilde{C}_\mathrm{t}$) and an output gate o_t. The cell states act as a transport highway to transfer the information from previous intervals all the way down the entire sequence chain. The gates can regulate the flow of information by either adding or removing the information from the cell state. The inputs to the LSTM are the previous hidden state $h_\mathrm{t\text{-}1}$ and the current state x_t. The forget gate uses a sigmoid activation function to decide on which information to throw away from the cell state and is described using (2.33a).

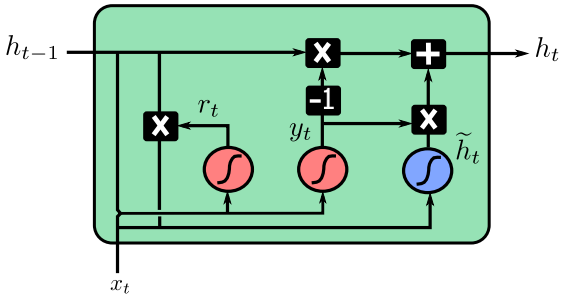
$$f_\mathrm{t} = sig\left(W_f\left[h_{t-1}^T, X_t^T\right] + b_f\right) \tag{2.33a}$$
$$i_\mathrm{t} = sig\left(W_i\left[h_{t-1}^T, X_t^T\right] + b_i\right) \tag{2.33b}$$
$$\widetilde{C}_\mathrm{t} = tanh\left(W_C\left[h_{t-1}^T, X_t^T\right]^T + b_C\right) \tag{2.33c}$$
$$C_\mathrm{t} = f_t * C_{t-1} + i_t * \widetilde{C}_\mathrm{t} \tag{2.33d}$$
$$o_\mathrm{t} = sig\left(W_o\left[h_{t-1}^T, X_t^T\right] + b_o\right) \tag{2.33e}$$
$$h_\mathrm{t} = o_t * tanh(C_t) \tag{2.33f}$$

The input gate in an LSTM is a combination of two layers. The first layer is referred to as input layer gate i_t, which decides on which important information to retain to update in the cell-state and the second layer is a *tanh* layer which adds new candidate values $\widetilde{C}_\mathrm{t}$ to the cell state as given in (2.33b) and (2.33c), respectively. Thus, the old cell state $C_\mathrm{t\text{-}1}$ is updated to a new cell state C_t using the forget gate and the input gate information according to (2.33d). Finally, the cell state passes through the *tanh* activation function and the final output is filtered using a sigmoid layer resulting in the next hidden state h_t as described in (2.33e) and (2.33f). In (2.33), W_f, W_i,W_C, W_o are the weights and b_f, b_i,b_C, b_o are the biases of the forget, input and output gates, respectively.

In contrary to the LSTM, the GRU has a simpler cell structure with only two gates, namely, reset r_t and update gate y_t, as illustrated in Figure 2.8(b). The reset gate is used to determine how much historical information to forget and the update gate decides



(a) Long Short-Term Memory (LSTM)



(b) Gated Recurrent Unit (GRU)

Figure 2.8: Internal cell structure of LSTM and GRU

which new information must be passed along to the future. The equations for the GRU are given by

$$y_t = sig\left(U_y x_t + W_y h_{t-1} + b_y\right) \tag{2.34a}$$
$$r_t = sig\left(U_r x_t + W_r h_{t-1} + b_r\right) \tag{2.34b}$$
$$\tilde{h}_t = tanh\left(U_h x_t + W_h(r_t \odot h_{t-1} + b_h)\right) \tag{2.34c}$$
$$h_t = (1 - y_t) \odot h_{t-1} + y_t \odot \tilde{h}_t \tag{2.34d}$$

where U_*, W_* are the weights, b_* is the bias, while $\odot$ is the scalar product of the vectors and *sig* refers to the sigmoid activation function.

2.2.5 Evaluation Metrics

To assess the performance of classification models, various evaluation metrics are employed in this study.

- **Recall** or true positive rate, measures the proportion of all actual positive instances correctly classified as positive:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.35}$$

where TP represents true positives (correctly classified positive cases) and FN represents false negatives (actual positives incorrectly classified as negatives).

- **Precision** calculates the proportion of correctly predicted positive instances among all instances classified as positive:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.36}$$

where FP denotes false positives (incorrectly classified negative instances as positive).

- **F1-score**, the harmonic mean of precision and recall, provides a balanced measure of model performance:

$$\text{F1} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2TP}{2TP + FP + FN} \tag{2.37}$$

To evaluate the performance of regression models, several metrics were used in this study.

- **Mean Absolute Error (MAE)**, which quantifies the average magnitude of errors in predictions. It is defined as

$$\text{MAE} = \sum_{i=1}^{N} \frac{|\hat{x}_{t(i)} - x_{t(i)}|}{N} \tag{2.38}$$

where $\hat{x}_{t(i)}$ is the predicted value at the i^{th} time step, $x_{t(i)}$ is the corresponding true observation and N represents the total number of observations.

- **Mean Squared Error (MSE)** calculates the average squared differences between predicted and actual values, given by

$$\text{MSE} = \sum_{i=1}^{N} \frac{(\hat{x}_{t(i)} - x_{t(i)})^2}{N} \tag{2.39}$$

- **Root Mean Squared Error (RMSE)** is the square root of MSE and is expressed as

$$\text{RMSE} = \sqrt{\sum_{i=1}^{N} \frac{(\hat{x}_{t(i)} - x_{t(i)})^2}{N}} \tag{2.40}$$

2.3 User Acceptance Models

This section gives an insight into the state-of-the-art user acceptance models that are widely used to evaluate and understand the factors influencing the adoption of new technologies. To explain and predict user acceptance of new technologies, Technology Acceptance Model (TAM) and Theory of Planned Behavior (TPB) are widely used in information systems.

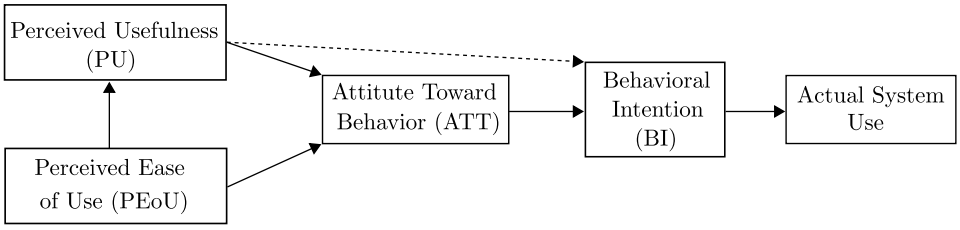


Figure 2.9: Technology Acceptance Model [Dav85]

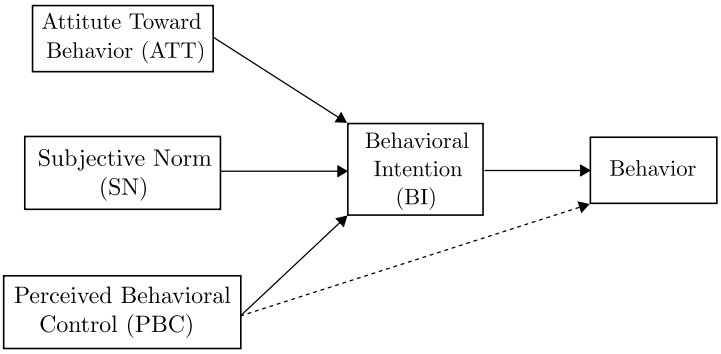


Figure 2.10: Theory of Planned Behavior [Ajz91]

2.3.1 Technology Acceptance Model (TAM)

TAM developed by Fred Davis [Dav85], is formulated based on the Theory of Reasoned Action (TRA). TAM posits that an individual's intention to use a technology is primarily determined by two important constructs: perceived usefulness (PU) and perceived ease of use (PEoU)(defined in Table 2.1). These two factors influence the user's attitude towards using the technology, which in turn affects their behavioral intention to use it and ultimately, their actual system use. The first version of the TAM is illustrated in Figure 2.9. Over the years, TAM has undergone several iterations and extensions, such as TAM2 and TAM3, which incorporate additional external variables and social influence factors to enhance its explanatory power.

2.3.2 Theory of Planned Behavior (TPB)

The TPB model proposed by Icek Ajzen [Ajz91], is also based on TRA and serves as an extension to TAM. This model overcomes the limitation of TAM in dealing with behaviors over which people have incomplete volition control and improves the predictive ability.

Table 2.1: Summary of the components in the user acceptance models

Models	Components	Definitions
TAM	PU	The extent to which an individual perceives that using a given system will improve his or her job performance [Dav85].
	PEoU	The extent to which an individual perceives that using a given system will require minimal effort [Dav85].
	ATT	An individual's positive or negative feelings towards engaging in a target behavior [FA75].
	BI	An individual's readiness to perform a particular behavior [Ajz91].
TPB	SN	An individual's perception of whether the people significant to them believe they should or should not perform a particular behavior [FA75].
	PBC	The perceived level of ease or difficulty associated with performing a particular behavior [Ajz91].
	ATT	same as in TAM
	BI	same as in TAM

TPB extends the TRA by incorporating perceived behavioral control (PBC), which reflects an individual's perception of their ability to perform a given behavior. TPB proposes that behavior intention (BI) and PBC jointly determine the actual use behavior rather than just attitude toward behavior (ATT) which determines the actual use in TAM. The TPB model is illustrated in Figure 2.10. The theory of TPB posits that three core components influence BI and actual behavior: ATT, Subjective Norms (SN) and PBC (defined in Table 2.1). Together, these factors shape an individual's intention to perform a behavior, which in turn is a strong predictor of actual behavior.

2.4 Summary

This chapter introduced the definition of a continuous-time constrained optimal control problem (OCP) and presented numerical methods for solving it. The limitations of Pontryagin's minimum principle in handling inequality constraints and discrete-time systems were discussed. The general form of dynamic programming was also examined, highlighting its robustness in managing constraints and ensuring global optimality, while also noting its computational inefficiency due to the curse of dimensionality. Additionally, the model predictive control (MPC) approach was explored, emphasizing its advantages in online optimization, computational efficiency and adaptability to disturbances. Furthermore, this chapter covered machine learning techniques for both unsupervised learning (dimensionality reduction and clustering) and supervised learning (classification and regression). The chapter concluded with a review of state-of-the-art user acceptance models, specifically the Technology Acceptance Model (TAM) and the Theory of Planned Behavior (TPB). In the next chapter, the architecture of the predictive eco-driving assistance system (EDAS) will be introduced.

3 Predictive Eco-Driving Assistance System (EDAS)

This chapter presents a comprehensive overview of the architecture and core building blocks of the predictive Eco-Driving Assistance System (EDAS) proposed in this dissertation. Additionally, it introduces the architecture of the learning-based driver behavior modeling approach, which plays a crucial role in improving the overall efficiency of the predictive EDAS.

3.1 Architecture of Predictive EDAS

The primary contribution in this dissertation is to develop a novel predictive EDAS with a goal to assist the drivers' in improving their driving style through optimal speed recommendations and thereby reduce the energy consumption in BEVs. The architecture of the proposed predictive EDAS is illustrated in Figure 3.1. Predictive EDAS in this work is a combination of eco-driving model predictive controllers (Eco-MPCs) and an eco-driving feedback system. The Eco-MPCs consist of two linear model predictive controllers, namely reference tracking MPC and car-following MPC. Moreover, an offline optimization using dynamic programming recommends a reference speed to the Eco-MPCs in certain driving situations (e.g., in the freeway scenario). The optimization uses the predictive surrounding information gathered through the onboard sensors (radar or lidar) and V2V communication, for instance the relative distance to the leader vehicle and its velocity, the upcoming traffic light (TL) signal phase and timing (SPaT) information obtained via V2I communication, and geographical information of the driving route (e.g., elevation, road curvature, road speed limits, location of the TL signals and stop signs).

The switching logic in the architecture primarily uses the inter-vehicle distance between the host vehicle and the leader vehicle ahead as the basis for choosing a suitable controller. In a freeway scenario, when a leader vehicle is not in the sensor range of the host vehicle, the reference tracking MPC is engaged. Conversely, when the leader vehicle is within the sensor range, the car-following MPC is selected. The energy-optimal control sequence obtained from either of the Eco-MPC controllers at each sample time step is in turn used to compute an optimal speed, which is communicated to the driver through an eco-driving feedback system in a visual and auditory manner. Eventually, the driver

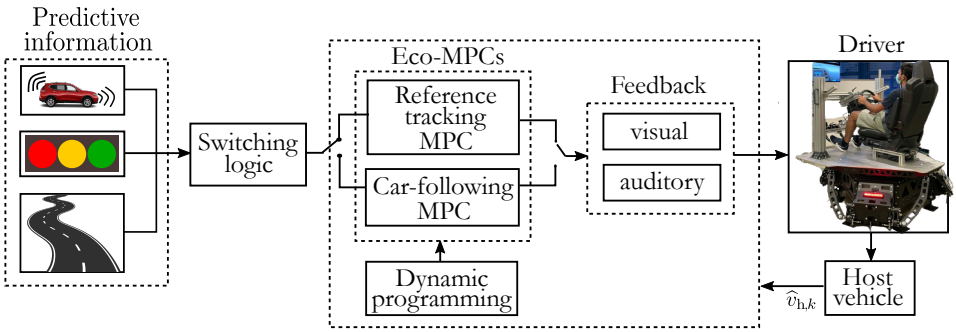


Figure 3.1: Architecture of the predictive eco-driving assistance system (EDAS)

tracks these continuous eco-driving speed recommendations to adjust their driving style, and drive the host vehicle energy-efficiently. At each timestep, the measured host vehicle velocity $\widehat{v}_{\mathrm{h}}$ is provided as input to the Eco-MPCs and the optimization takes place in a closed-loop fashion, thus promoting the real-time control of the host vehicle and also accounting for the additive disturbances (e.g. abrupt change in surrounding driving situation or in the driver behavior). Each of the blocks from the predictive EDAS architecture will be briefly elaborated in the following subsections.

3.1.1 Predictive Information

Eco-driving assistance systems significantly benefit from medium- to long-range predictive information, allowing to adapt the driver's driving style to changing conditions at an early stage. Radar, lidar and camera-based sensors are commonly utilized for the near- to medium-range object detection (up to 250 m), with each sensor offering unique capabilities. For instance, radar sensors are highly effective in accurately determining the relative speed of objects and function reliably under all weather conditions. Lidar sensors provide greater accuracy in measuring the distance to objects but are more sensitive to adverse weather conditions. Meanwhile, camera sensors are highly effective at capturing visual information for object recognition and classification, offering rich contextual details but requiring adequate lighting.

For long-range predictive information, real-time data exchange via vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication can be leveraged. In V2V communication, vehicles share critical information such as speed, location, and acceleration, allowing predictive systems to anticipate traffic behavior beyond the range of onboard sensors. To enable this communication, vehicles are equipped with onboard units (OBUs) that exchange relevant data with OBUs in nearby vehicles. In V2I communication, vehicles interact with infrastructure elements such as traffic lights, road signs, and traffic management systems to receive data, including signal phase and timing (SPaT), road speed limits, location of the TL signals and stop signs. To enable communication between vehicles and transportation infrastructure, such as receiving SPaT messages

from a traffic light controller to the host vehicle, a roadside unit (RSU) is typically installed alongside roads. This information is transmitted using dedicated short-range communications (DSRC) or cellular vehicle-to-everything (C-V2X) protocols.

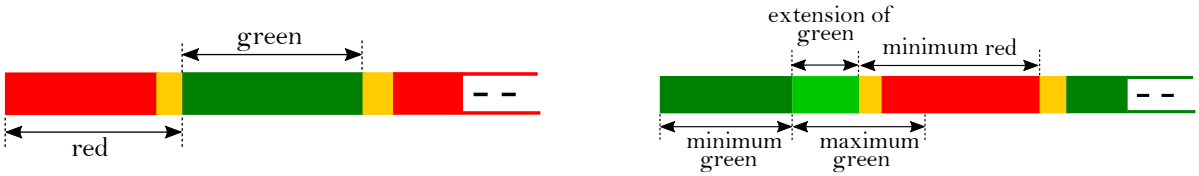
For energy-efficient longitudinal control of the host vehicle, this work primarily considers predictive information about three key elements: the leader vehicle, traffic light (TL) signals, and route topology. Leveraging predictive information about the leader vehicle, such as acceleration, velocity, and relative distance, facilitates smoother and more anticipatory driving behaviors for the host vehicle. This not only helps reduce energy consumption but also enhances driving safety by maintaining the desired inter-vehicle distance. Additionally, this dissertation focuses on predicting the future velocities of the leader vehicle using machine learning techniques. It is assumed that these predicted velocities will be transmitted to the host vehicle via V2V communication.

Moreover, traffic light (TL) signals in urban environments are pivotal not only in regulating traffic flow but also in influencing driving behavior. To adapt to varying traffic conditions at different times of the day and locations, TL signals are primarily controlled using two methods: deterministic and actuated signals. Deterministic signals operate on a fixed cycle time logic, where the signal pattern repeats consistently, as illustrated in Figure 3.2a. In contrast, actuated TL signals adjust their timing dynamically based on vehicle detection, typically using inductive loops as vehicles approach, as shown in Figure 3.2b. For simplicity, this work combines yellow and red light phases into a single category and assumes that the host vehicle receives SPaT information in real-time through V2I communication.

Furthermore, digital maps play a crucial role in helping drivers anticipate the route ahead, allowing them to extend their driving horizon beyond their immediate field of vision. While onboard sensors in the host vehicle can detect traffic signs and road characteristics within their range, predictive route information from digital maps provides additional value. To access such predictive data, the ADASIS v2 protocol [DBR+] standardises the compilation and transmission of route datasets through a uniform interface known as the Electronic Horizon [REKR08], which typically includes an electronic map and a satellite navigation system. Provided that the vehicle's route is predefined at the beginning of a trip, the electronic horizon enables the prediction of route features several kilometers ahead, such as road speed limits, elevation and road curvature. Since the commercial licensing of digital maps falls outside the scope of this dissertation, it is assumed that the host vehicle receives real-time route topology information within the simulation environment.

3.1.2 Switching Logic

The switching logic proposed in this work, as illustrated in Figure 3.3, is designed to select the appropriate Eco-MPC that determines the optimal control sequences for the energy-efficient longitudinal control of the host vehicle. The major deciding parameter



(a) Fixed cycle time (b) Varied cycle time

Figure 3.2: (a) Deterministic TL signals (b) Actuated TL signals

for the switching between the controllers is the inter-vehicle distance d_{rel} between the host vehicle and the leader vehicle. To measure d_{rel}, it is assumed that the host vehicle is equipped with a distance sensor (radar or lidar). Additionally, it is assumed that the host vehicle receives the SPaT information (e.g., traffic light states) in real-time through V2I communication.

If d_{rel} is smaller than a limit value d_{limit}, which represents the sensor range, it is considered that the leader vehicle is within range of the host vehicle. In such a scenario, car-following behavior is prioritized, and the car-following MPC is selected, as illustrated in Figure 3.3A. While performing car-following, the influence of the immediate traffic light signal on the host vehicle is accounted for only if the relative distance to the upcoming traffic light d_{ITS} is within a predefined horizon, $d_{\text{ITS,horizon}}$. Based on iterative analysis, this horizon is chosen as 200 m in this work, as the results showed smooth deceleration maneuvers for the host vehicle. It is to be noted that within this horizon the upcoming phase of the traffic light is calculated based on the host vehicle's current velocity and estimated time to reach the signal. If the calculated upcoming phase is red, the host vehicle stops tracking the leader vehicle and prepares to stop energy-efficiently at the traffic light signal. Conversely, if the upcoming phase is not red or if d_{ITS} is greater than $d_{\text{TS,horizon}}$, the host vehicle continues to follow the leader vehicle using the car-following MPC, maintaining a desired inter-vehicle distance.

The proposed switching logic ensures that the car-following MPC effectively achieves specific control objectives for the host vehicle while tracking a leader vehicle and approaching an immediate traffic light (TL) signal. First, if the leader vehicle encounters a green light but the host vehicle predicts an upcoming red TL using SPaT information, the host vehicle stops tracking the leader and prepares to stop energy-efficiently near the TL signal. Second, if both vehicles face a red TL signal, the host vehicle comes to a halt in an energy-efficient manner while maintaining a safe distance from the leader vehicle. Finally, if the traffic light phase is green for both vehicles, the host vehicle continues car-following while adhering to the DP constraint, which restricts the host vehicle's speed (v_{h}) from exceeding the reference speed (v_{ref}). This approach prioritizes energy efficiency by avoiding unnecessary acceleration maneuvers to catch up with the leader vehicle.

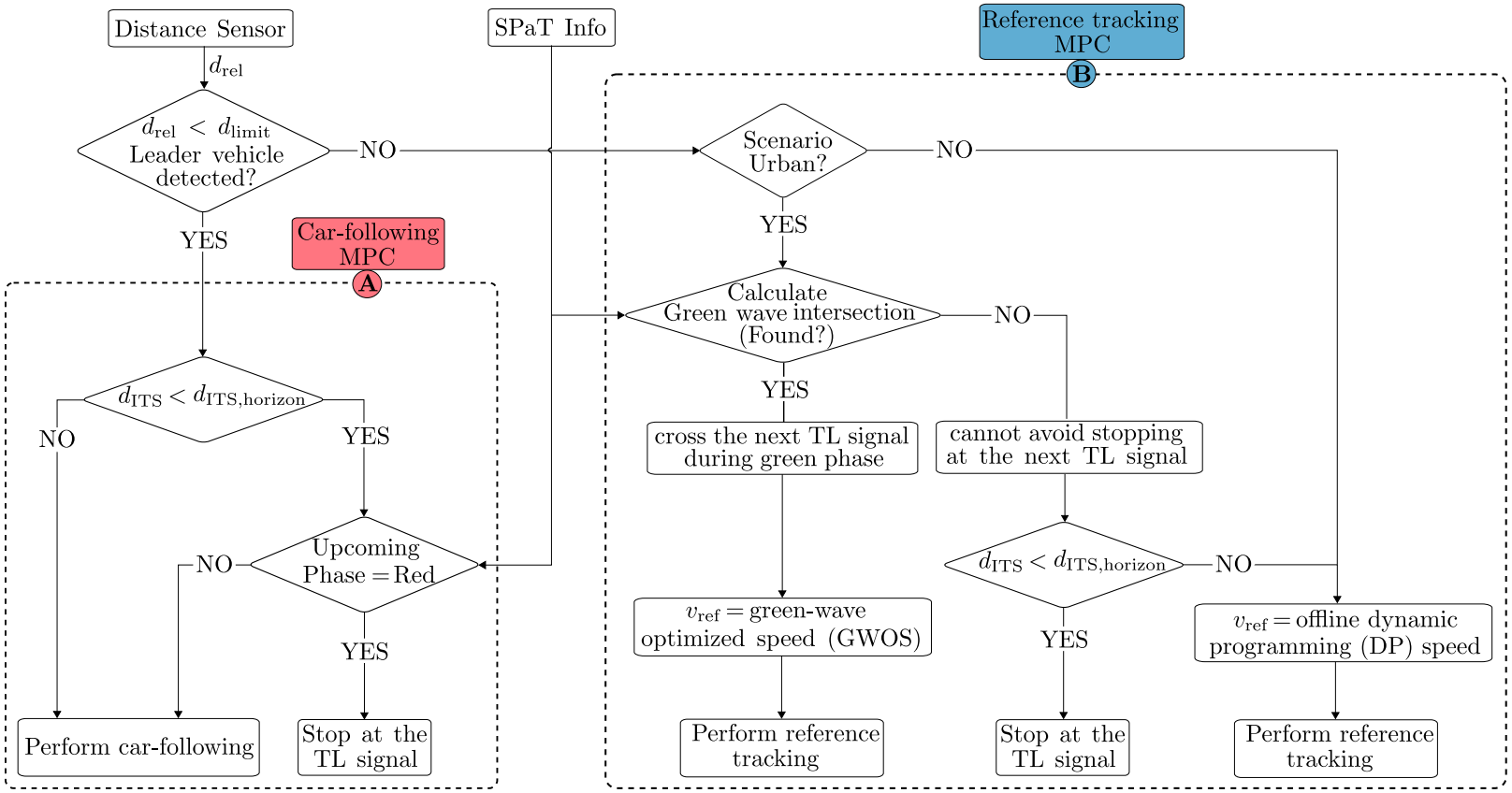


Figure 3.3: Flowchart showcasing the switching logic between the controllers

If d_{rel} is greater than a limit value d_{limit} it is considered that the leader vehicle is not in the detection range of the host vehicle's sensors. In such cases, a reference tracking MPC is selected, as illustrated in Figure 3.3B. An energy-optimal strategy for the host vehicle in urban environments is to minimize unnecessary stops at TL signals by maintaining a socially acceptable road speed. To generate optimal speed trajectories for the host vehicle at signalized intersections, a green wave velocity intersection range is calculated using a green wave Algorithm 3.1, based on the idea from [AV11]. It is assumed that the host vehicle receives SPaT information at every timestep.

Algorithm 3.1 Green Wave Optimal Speed (GWOS) Range Algorithm

$prev\_intsec \leftarrow [\,]$, $intsec \leftarrow [\,]$ ▷ Initialize GWOS range
$found \leftarrow 0$
for $i \leftarrow x$ to n_{TL} **do** ▷ loop through all the upcoming TLs
  for $j \leftarrow 1$ to n_{phase} **do** ▷ loop through signal phases
    $intsec \leftarrow \left[\frac{d_{\text{TS},i}}{r_{ij}}, \frac{d_{\text{TS},i}}{g_{ij}}\right] \cap [v_{\min}, v_{\max}]$
    if $intsec \neq \emptyset$ and $found \leftarrow 0$ **then**
      $prev\_intsec \leftarrow intsec$
      $found \leftarrow 1$
      break
    else if $intsec \neq \emptyset$ and $found \leftarrow 1$ **then**
      $prev\_intsec \leftarrow prev\_intsec \cap intsec$
      break
    end if
  end for
end for
$intsec \leftarrow prev\_intsec$
return $intsec$

The algorithm initializes an empty intersection range and then iteratively evaluates all upcoming TL signals and their respective signal phases. For each TL signal, the algorithm computes the intersection of the green wave speed range, where $v_{\min}$ and $v_{\max}$ denote the minimum and maximum road speed limits respectively, $d_{\text{TS},i}$ represents the distance to the i^{th} traffic light signal, r_{ij} represents the start time of the j^{th} red phase of the i^{th} traffic light signal. Similarly, g_{ij} represents the start time of the j^{th} green phase of the i^{th} traffic light signal. If a valid intersection range is found during any phase, it is stored as the initial range. For subsequent traffic lights, the range is iteratively updated by calculating the intersection with the previously determined range.

The reference tracking MPC fulfills specific control objectives in freeway scenarios and in the presence of TL signals through the proposed switching logic. If the GWOS intersection range, as discussed in Algorithm 3.1 is found, the reference tracking MPC tracks the optimal reference velocity v_{ref} to ensure the host vehicle crosses signalized intersections during a green phase. However, if the GWOS range for green wave is not

found, the host vehicle is advised to travel at a reference velocity determined by DP, until the switching criteria are met. It is important to note here that not finding an intersection range for green wave means that the host vehicle must definitely stop at the upcoming TL signal. In such a scenario, the host vehicle decelerates in an energy-efficient manner to reach a standstill at the signal once the relative distance to the TL signal is within $d_{\mathrm{ITS,horizon}}$.

3.1.3 Eco-driving Feedback System

In this work, continuous eco-driving feedback is provided to the driver in a visual fashion, with additional auditory alerts for unsafe driving situations. Visual feedback is displayed on the host vehicle's windshield through a heads-up display (HUD), while auditory feedback is delivered via an in-vehicle speaker. An internal study was conducted to evaluate four different visualization concepts (bar, scale, tachometer, and arrow) for the predictive EDAS. A total of 22 participants took part in an online survey and rated the visualization concepts using a questionnaire based on [VHD97] scales. The evaluation focused on the usefulness and satisfaction scores obtained from the survey, with results indicating that 64% of participants preferred the arrow-type visualization, albeit with suggestions for improvement. Additionally, the amount of visual information being displayed on the HUD may cause driver distraction and result in an increased cognitive workload. Considering these factors, an improvised design of the arrow visualization with minimal information is presented in this dissertation.

The speed advisory human-machine interface (HMI) is illustrated in Figure 3.4, displays information across three regions on the HUD. The first region shows the current speed of the host vehicle. The second region depicts an arrow providing speed adjustment guidance: to either speed up (pointing upwards and green in color) if the current speed is below the optimal speed or to slow down (pointing downwards and red in color) if above the optimal speed. The arrow's length represents the magnitude of speed increment or decrement. The third region provides supplementary information such as traffic light remaining time, green-wave indications, and traffic sign detection to help the driver better interpret the suggestions given by predictive EDAS. Additionally, the allowable road speed limit information is displayed below the current speed of the host vehicle for added clarity.

3.1.4 Driver-in-the-Loop Setup

To evaluate the predictive EDAS, a virtual driving environment is used in this work, as testing such a system in the real-world traffic can be complex and time-intensive. Moreover, it is essential for the driver to be actively involved in the simulation loop. Therefore, a dynamic driving simulator was chosen to provide drivers with a realistic driving experience.

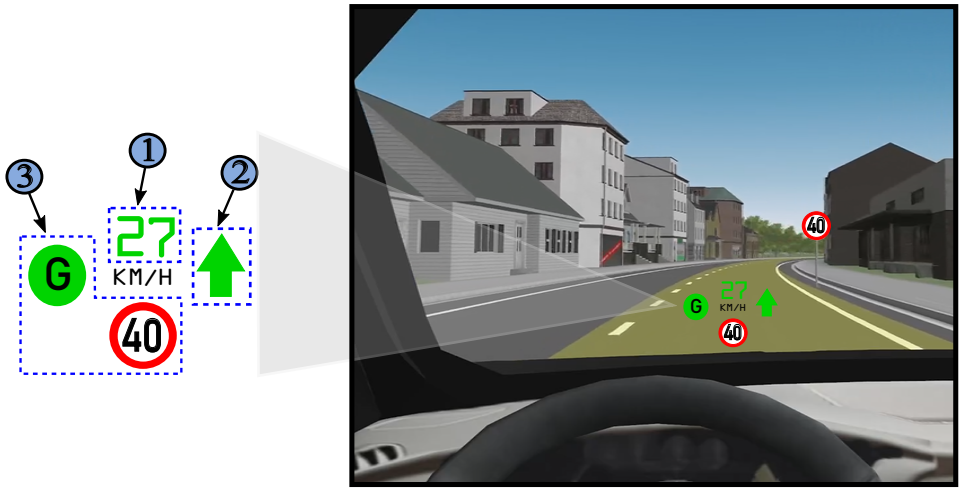


Figure 3.4: Human-Machine Interface. 1. Host vehicle actual speed, 2. Advisory speed indication using an arrow, 3. Additional useful information to the driver.

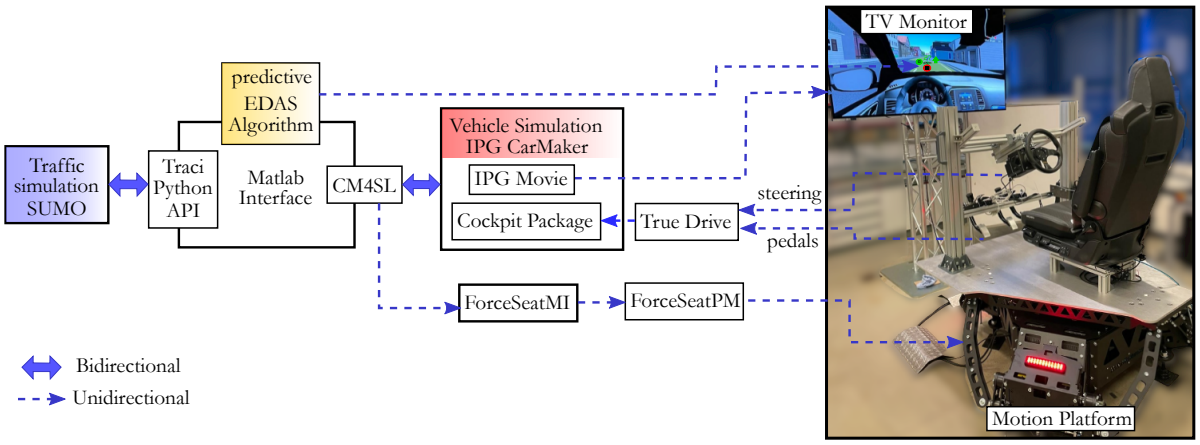


Figure 3.5: Schematic of the driving simulator setup

The schematic of the driving simulator setup used in this work is shown in Figure 3.5. The setup includes a PS-6TM-550 six-degree-of-freedom (6-DoF) motion platform from Motion Systems, a Simucube 2 Pro steering wheel, adjustable seat and pedals (throttle and brake). Using the ForSeatMI programming interface and the software engine ForceSeatPM Platform Manager from Motion Systems [Mot], the dynamic platform's rotational (roll, pitch and yaw) and translational (heave, surge and sway) motions can be controlled. The Simucube steering wheel features a direct-drive force-feedback motor, realistically simulating forces from the road and steering linkages. Additionally, the Simucube True Drive software allows fine-tuning of the steering wheel and pedal settings to replicate the feel of a BEV. In this work, the virtual driving environment was modeled in IPG CarMaker, with the cockpit view displayed on a 65-inch LCD TV monitor using IPG Movie. Furthermore, views from side and rear mirrors were also included to ensure the driver remains aware of the surrounding environment. To connect external

controllers (steering and pedals) to CarMaker, the IPG Cockpit Package extension was utilized.

To achieve realistic driving behaviors of surrounding traffic in the virtual environment, the strengths of both traffic simulation and vehicle simulation tools were integrated. A co-simulation framework was developed, facilitating real-time information exchange between IPG CarMaker, a vehicle simulation software tailored for on-road passenger and commercial vehicles, and SUMO, an open-source microscopic traffic simulation software. Matlab is used as the primary interface tool for connecting both SUMO and the IPG CarMaker via the TraCI Python and CarMaker4Simulink (CM4SL) APIs, respectively. Additionally, the vehicle model in IPG CarMaker was parameterized to replicate a Nissan Leaf BEV. The driver-in-the-loop (DiL) simulation was conducted on a 64-bit Windows 10 PC equipped with an Intel Core i9-10850K CPU, 3.9 GHz clock frequency and 64 GB of RAM. Finally, the predictive EDAS algorithm computes optimal advisory speeds in real-time and provides continuous eco-driving speed recommendations to the driver via the HMI (Figure 6.4). Ultimately, the driver controls the host vehicle while following these recommendations, enabling an energy-efficient driving behavior.

3.1.5 Learning-Based Driver Behavior Modeling

One of the key contributions of this dissertation is the modeling of driver behavior to predict delays, referred to as speed-tracking errors, associated with drivers when following the advisory speed recommended by the proposed predictive EDAS (Section 3.1). The architecture of the proposed learning-based driver behavior modeling approach is illustrated in Figure 3.6. This architecture is structured into four main phases, each briefly described below:

Online Speed Suggestion and Actuation Phase

In this phase, advisory speed suggestions are provided to the driver as visual feedback through a HMI (Section 3.4). The driver then controls the host vehicle by attempting to follow these suggestions within the driver-in-the-loop setup (Section 3.1.4).

Offline Clustering and Training Phase

This phase utilizes historical driving data to perform clustering and model training. Initially, principal component analysis (PCA) is applied to identify key variables and reduce the dimensionality of the dataset. Subsequently, the reduced data is grouped into n distinct clusters using hierarchical cluster analysis (HCA).The resulting cluster IDs serve as labels for supervised learning, enabling the training of a temporal convolutional network (TCN) to classify driving styles. In addition, both the stochastic volatility (SV) model and deterministic gated recurrent unit (GRU) model are trained separately for each cluster to predict the driver's speed-tracking error.

Online Classification Phase

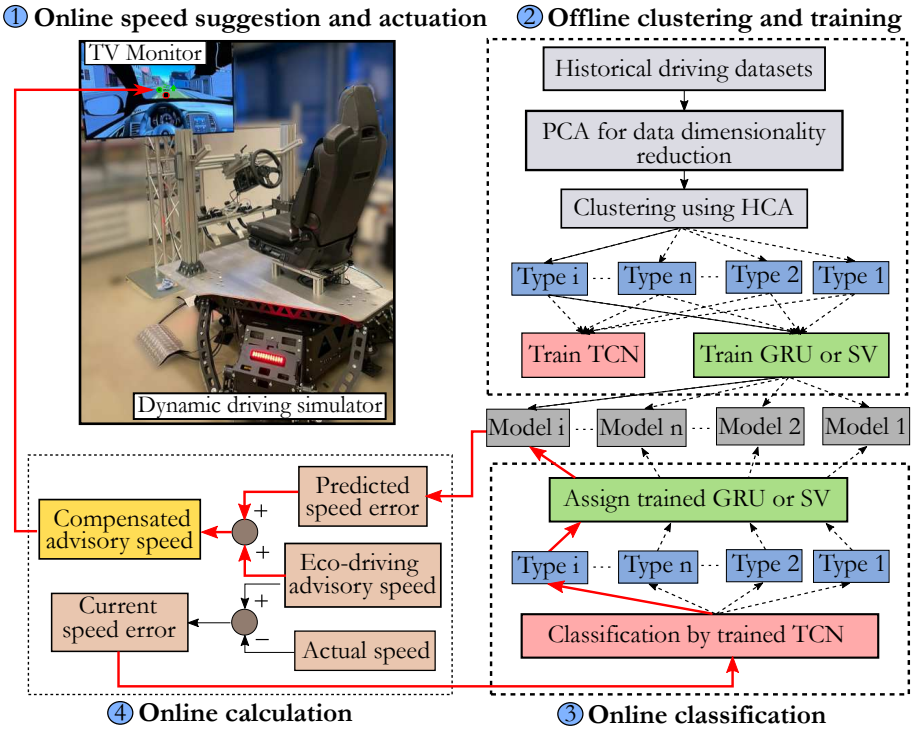


Figure 3.6: Architecture of learning-based driver behavior modeling approach

In this phase, the trained TCN model is used to classify a new driver's behavior into one of the previously identified clusters based on current and past driving data. Once the appropriate cluster ID is assigned, the corresponding trained prediction model (stochastic or deterministic) is elected to forecast the driver's speed-tracking error.

Online Calculation Phase

Here, the selected prediction model is employed to forecast the driver's speed-tracking error in real-time at every sampling interval (0.2 s). This predicted error is then used to compensate the eco-driving advisory speed. Simultaneously, the actual speed-tracking error is calculated as the difference between the recommended advisory speed and the measured host vehicle speed. This current error, along with past speed-tracking errors, forms the input for the online classification phase at the subsequent time step.

3.2 Summary

This chapter presented the architecture of the predictive eco-driving assistance system (EDAS), along with its key components, including the predictive information, switching logic, eco-driving feedback system, and the driver-in-the-loop setup. The primary objective of the proposed predictive EDAS is to support drivers in enhancing their driving style by providing optimal speed recommendations across various driving scenarios,

thereby reducing the energy consumption in vehicles. Furthermore, a learning-based driver behavior modeling architecture was introduced, incorporating both unsupervised and supervised learning techniques for clustering and classification of driving data, respectively. In addition, a driver speed-tracking error compensation approach was integrated into the system architecture to further improve the efficiency and accuracy of the predictive EDAS. The following chapter will focus on the problem formulations of eco-driving model predictive controllers (Eco-MPCs) for battery electric vehicles (BEVs) and internal combustion engine (ICE) vehicles, along with their validation through simulation studies.

4 Eco-Driving Model Predictive Controllers

This chapter discusses the problem formulations for eco-driving model predictive controllers (Eco-MPCs) in detail, with the objective of minimizing energy consumption in battery electric vehicles (BEVs) while ensuring safety and driving comfort. To develop such controllers capable of planning energy-optimal speed trajectories for the host vehicle in various traffic scenarios, this chapter emphasizes the accurate modeling of longitudinal vehicle dynamics, and provides detailed insights into the modeling of BEV efficiency maps. Additionally, this chapter focuses on adapting the MPC problem formulations developed for BEVs to make them applicable to internal combustion engine (ICE) vehicles. For this purpose, a city bus model is briefly introduced. Finally, the chapter presents the performance evaluation results of the proposed Eco-MPCs in simulation.

4.1 Modeling of the Longitudinal Dynamics

4.1.1 System Dynamics

This dissertation focuses on eco-driving assistance systems with an emphasis on longitudinal control. Therefore, this section specifically examines longitudinal vehicle dynamics. Consider a host vehicle with mass m travelling at a speed v_h on an inclined road as depicted in the Figure 4.1. The longitudinal dynamics of the host vehicle can be mathematically expressed as,

$$a_\mathrm{h} = \frac{dv_\mathrm{h}}{dt} = \underbrace{\frac{1}{(m + m_\mathrm{r})}}_{m_\mathrm{eq}} (\underbrace{F_\mathrm{t} - F_\mathrm{b}}_{F_\mathrm{w}} - \underbrace{F_\mathrm{a} - F_\mathrm{roll} - F_\mathrm{g}}_{F_\mathrm{r}}) \tag{4.1}$$

where a_h and v_h are the host vehicle acceleration and velocity respectively, m is the mass of the vehicle with driver and cargo, m_r is the equivalent mass of the rotational components in the vehicle, m_eq is the eqivalent mass of the vehicle, F_w is the force exerted by the wheel, and is equal to the difference of the traction force F_t and braking force F_b. Moreover, F_r represents the combined resistance forces, including aerodynamic resistance F_a, rolling resistance F_roll and gradient resistance F_g forces.

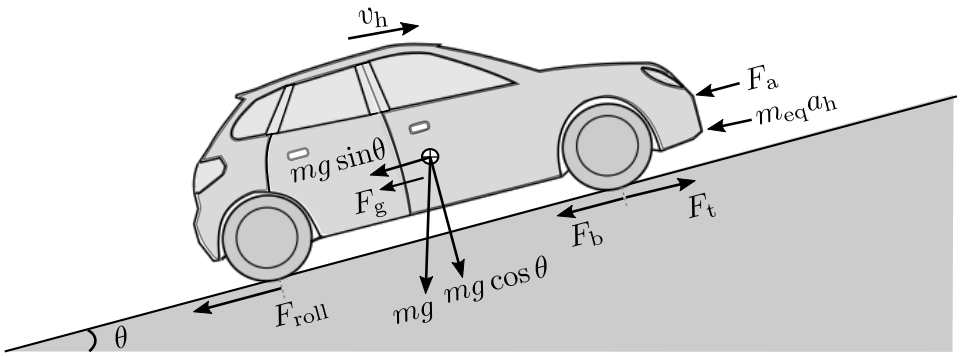


Figure 4.1: Schematic representation of the forces acting on a host vehicle during driving

The aerodynamic drag force acting on the vehicle in motion is caused by two components, mainly the pressure difference due to air flow separation between the front and rear of the vehicle and also the viscous friction due to the surrounding air on the vehicle surface. The aerodynamic resistance force F_a is described by

$$F_a = \frac{1}{2}\rho A_f c_a v_h^2 \tag{4.2}$$

where c_a is the drag coefficient, A_f is the frontal cross-sectional area of the vehicle, ρ is the density of the air. To handle the nonlinearity occuring due to the term v_h^2 in F_a, an approximation of the aerodynamic resistance $F_a \approx \frac{1}{2}\rho A_f c_a(p_1 v_h + p_2)$ is considered in this work as illustrated in the Figure 4.2, where p_1 and p_2 are the coefficients obtained through linear fitting. The parameters for the BEV and the city bus equipped with conventional ICE are listed in Table A.2 and Table A.3 in the Appendix, respectively. The rolling friction force is expressed as

$$F_{roll} = c_r mg\cos\theta \tag{4.3}$$

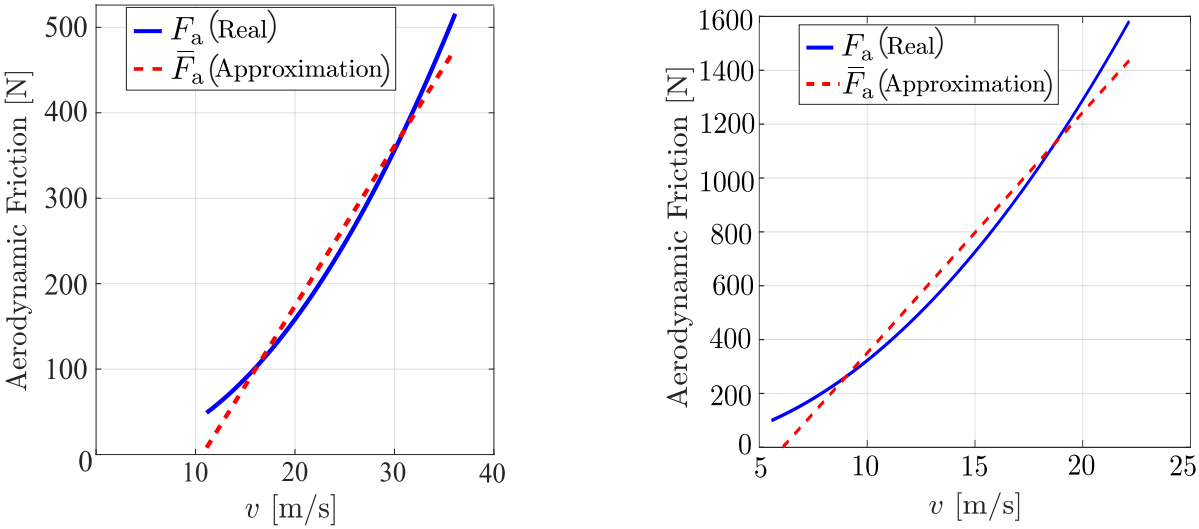
where g is the acceleration due to gravity, c_r is the rolling friction coefficient and θ is the road slope.

The force due to gravity when driving on a road with slope θ has considerable influence on the vehicle behavior. The gradient force is expressed mathematically as

$$F_g = mg\sin\theta \tag{4.4}$$

In conclusion, the longitudinal dynamics of the host vehicle can be written by combining the equations from (4.1) to (4.4) as

$$a_h = \frac{dv}{dt} = \frac{1}{(m+m_r)}\left(F_t - F_b - \frac{1}{2}c_d A_v \rho_a (p_1 v_h + p_2) - mg\sin\theta - c_r mg\cos\theta\right) \tag{4.5}$$



(a) Battery electric vehicle

(b) City bus with conventional ICE

Figure 4.2: Approximation of aerodynamic friction force

4.1.2 Prediction Model

The prediction model can be constructed by discretizing and reformulating the state equations into a state-space representation. The time-domain discretized form of (4.5) is expressed as

$$\frac{v_{\mathrm{h},k+1} - v_{\mathrm{h},k}}{\Delta t} = \frac{1}{m + m_{\mathrm{r}}} \left(F_{\mathrm{t},k} - \frac{1}{2} c_{\mathrm{d}} A_{\mathrm{v}} \rho_{\mathrm{a}} (p_1 v_{\mathrm{h}} + p_2) - mg \sin\theta_k - c_{\mathrm{r}} mg \cos\theta_k - F_{\mathrm{b},k} \right) \tag{4.6}$$

where Δt is the sampling time, k denotes the sample step, and v_{h} is the velocity of the host vehicle. The equation above can be rearranged into the form

$$v_{\mathrm{h},k+1} = \frac{\Delta t}{m + m_{\mathrm{r}}} \left(F_{\mathrm{t},k} - \frac{1}{2} c_{\mathrm{d}} A_{\mathrm{v}} \rho_{\mathrm{a}} (p_1 v_{\mathrm{h}} + p_2) - mg \sin\theta_k - c_{\mathrm{r}} mg \cos\theta_k - F_{\mathrm{b},k} \right) + v_{\mathrm{h},k} \tag{4.7}$$

Reordering similar terms gives

$$v_{\mathrm{h},k+1} = v_{\mathrm{h},k} \underbrace{\left(1 - \frac{\Delta t}{m + m_{\mathrm{r}}} p_1 \right)}_{A_{11}} + \underbrace{\left(\frac{\Delta t}{m + m_{\mathrm{r}}} \right)}_{B_{11}} F_{\mathrm{t},k} - \underbrace{\left(\frac{\Delta t}{m + m_{\mathrm{r}}} \right)}_{B_{12}} F_{\mathrm{b},k} - \underbrace{\left[\frac{\Delta t}{m + m_{\mathrm{r}}} (mg \sin\theta_k + c_{\mathrm{r}} mg \cos\theta_k + p_2) \right]}_{w_{1,\mathrm{k}}} \tag{4.8}$$

which can be written as

$$v_{\mathrm{h},k+1} = A_{11} v_{\mathrm{h},k} + B_{11} F_{\mathrm{t},k} - B_{12} F_{\mathrm{b},k} - w_{1,k} \tag{4.9}$$

The distance between the host vehicle and the leader vehicle can be calculated by

$$d_{\mathrm{rel},k+1} = d_{\mathrm{rel},k} + \left(\frac{v_{\mathrm{p},k} + v_{\mathrm{p},k+1}}{2} - \frac{v_{\mathrm{h},k} + v_{\mathrm{h},k+1}}{2} \right) \Delta t \tag{4.10}$$

where v_{p} is the leader vehicle velocity. By substituting $v_{\mathrm{h},k+1}$ from (4.9), and grouping similar terms gives the equation to calculate the relative distance between the host vehicle and leader vehicle as

$$\begin{aligned} d_{\mathrm{rel},k+1} = d_{\mathrm{rel},k} - \underbrace{\left[\frac{\Delta t}{2} \left(2 - \frac{\Delta t}{m + m_{\mathrm{r}}} p_1 \right) \right]}_{A_{21}} v_{\mathrm{h},k} - \underbrace{\left[\frac{(\Delta t)^2}{2\,(m + m_{\mathrm{r}})} \right]}_{B_{21}} F_{\mathrm{t},k} \\ + \underbrace{\left[\frac{(\Delta t)^2}{2\,(m + m_{\mathrm{r}})} \right]}_{B_{22}} F_{\mathrm{b},k} + \underbrace{\left[\frac{\Delta t}{2} \left(w_{1,k} + v_{\mathrm{p},k} + v_{\mathrm{p},k+1} \right) \right]}_{w_{2,\mathrm{k}}} \end{aligned} \tag{4.11}$$

which can be written as

$$d_{\mathrm{rel},k+1} = d_{\mathrm{rel},k} - A_{21} v_{\mathrm{h},k} - B_{21} F_{\mathrm{t},k} + B_{22} F_{\mathrm{b},k} + w_{2,k} \tag{4.12}$$

The relative distance between the host vehicle and the upcoming ITS (traffic light signal, stop sign or bus stop) at each time step is calculated using

$$d_{\mathrm{ITS},k+1} = d_{\mathrm{ITS},k} - v_{\mathrm{h},k} \Delta t \tag{4.13}$$

The state-space form using (4.9), (4.12) and (4.13) can be written as

$$\underbrace{\begin{pmatrix} v_{\mathrm{h},k+1} \\ d_{\mathrm{rel},k+1} \\ d_{\mathrm{ITS},k+1} \end{pmatrix}}_{x_{k+1}} = \underbrace{\begin{pmatrix} A_{11} & 0 & 0 \\ -A_{21} & 1 & 0 \\ -\Delta t & 0 & 1 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} v_{\mathrm{h},k} \\ d_{\mathrm{rel},k} \\ d_{\mathrm{ITS},k} \end{pmatrix}}_{x_k} + \underbrace{\begin{pmatrix} B_{11} & -B_{12} \\ -B_{21} & B_{22} \\ 0 & 0 \end{pmatrix}}_{B} \underbrace{\begin{pmatrix} F_{\mathrm{t},k} \\ F_{\mathrm{b},k} \end{pmatrix}}_{u_k} + \underbrace{\begin{pmatrix} -w_{1,k} \\ w_{2,k} \\ 0 \end{pmatrix}}_{w_k} \tag{4.14}$$

The discrete-time state-space representation of a linear time-invariant (LTI) system is expressed as

$$x_{k+1} = A x_k + B u_k + w_k \tag{4.15}$$

where A is the state matrix, B is the input matrix, x is the state vector, u is the control vector and w is the disturbance in the system.

Using (4.15), a prediction model over the prediction horizon H can be formulated in the matrix form as

$$X_k = \Phi x_k + \Gamma U_k + \Gamma_w W_k \tag{4.16}$$

where,

$$X_k = \begin{pmatrix} x_{k+1} \\ x_{k+2} \\ \vdots \\ x_{k+N} \end{pmatrix}, \; \Phi = \begin{pmatrix} A \\ A^2 \\ \vdots \\ A^N \end{pmatrix}, \; \Gamma = \begin{pmatrix} B & 0 & \dots & 0 \\ AB & B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \dots & B \end{pmatrix}$$

$$U_k = \begin{pmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+N-1} \end{pmatrix}, \; \Gamma_w = \begin{pmatrix} I & 0 & \dots & 0 \\ A & I & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1} & A^{N-2} & \dots & I \end{pmatrix}, \; W_k = \begin{pmatrix} w_k \\ w_{k+1} \\ \vdots \\ w_{k+N-1} \end{pmatrix}$$

4.2 Eco-MPCs for Battery Electric Vehicles

4.2.1 Model of the BEV Powertrain

Battery

A basic quasi-static Thevenin model of a battery, represented by the equivalent circuit shown in Figure 4.3, is adopted in this work. In this model, the battery is depicted as an ideal open-circuit voltage source U_{OC}, connected in series with the internal resistance R_{i}. Both R_{i} and U_{OC} are dependent on the battery's state of charge ζ and temperature ϑ_{bat}, and together they determine the terminal voltage U_{bat}. The load current I_{bat} is positive during discharging and negative during charging.

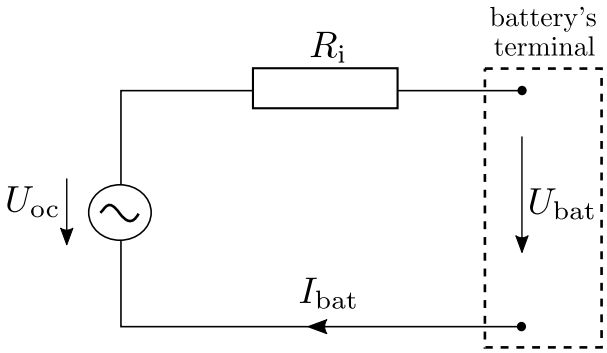


Figure 4.3: Quasi-static Thevenin model of a battery cell [KPKP21]

Using Kirchhoff's law, the governing equations for the cell are expressed as

$$U_{\mathrm{bat}} = U_{\mathrm{oc}}(\zeta, \vartheta_{\mathrm{bat}}) - R(\zeta, \vartheta_{\mathrm{bat}}) I_{\mathrm{bat}} \tag{4.17}$$

The output power P_{bat} at the battery cell terminal is given by

$$P_{\mathrm{bat}} = U_{\mathrm{bat}} I_{\mathrm{bat}} \tag{4.18}$$

Substituting (4.18) into (4.17), the terminal voltage U_{bat} can be expressed as

$$U_{\text{bat}}\left(\zeta, P_{\text{bat}}\right) = \begin{cases} \dfrac{U_{\text{oc}}(\zeta) - \sqrt{U_{\text{oc}}(\zeta)^2 - 4R_{\text{dis}}(\zeta)P_{\text{bat}}}}{2} & \text{if } P_{\text{bat}} \geq 0 \\ \dfrac{U_{\text{oc}}(\zeta) - \sqrt{U_{\text{oc}}(\zeta)^2 - 4R_{\text{chg}}(\zeta)P_{\text{bat}}}}{2} & \text{if } P_{\text{bat}} < 0 \end{cases} \tag{4.19}$$

where, a positive P_{bat} indicates battery discharge during driving mode with a resistance of R_{dis}, while a negative P_{bat} represents battery charging during regenerative braking mode with a resistance of R_{chg}.

The battery efficiency η_{bat} is defined as the ratio of the output power at the battery terminal P_{bat} to the total power input P_{in} during charging, and vice versa during discharging. It is expressed as

$$\eta_{\text{bat}}\left(\zeta, P_{\text{bat}}\right) = \begin{cases} \dfrac{P_{\text{bat}}}{P_{\text{in}}} = \dfrac{U_{\text{bat}}\left(\zeta, P_{\text{bat}}\right)}{U_{\text{oc}}(\zeta)} & \text{if } P_{\text{bat}} \geq 0 \\ \dfrac{P_{\text{in}}}{P_{\text{bat}}} = \dfrac{U_{\text{oc}}(\zeta)}{U_{\text{bat}}\left(\zeta, P_{\text{bat}}\right)} & \text{if } P_{\text{bat}} < 0 \end{cases} \tag{4.20}$$

The battery current I_{bat} is given by

$$I_{\text{bat}} = \frac{U_{\text{bat}}\left(\zeta, P_{\text{bat}}\right)}{R_{\text{i}}\left(\zeta\right)} \tag{4.21}$$

The rate of change in the battery's state of charge $\dot{\zeta}$ is calculated as

$$\dot{\zeta} = \frac{Q}{Q_0} = -\frac{\int I_{\text{bat}} dt}{Q_0} \tag{4.22}$$

where Q represents the charge capacity, Q_0 denotes the battery's full capacity, with a value of 32.8 Ah.

Electric Motor

The mechanical power output of the electric motor $P_{\text{m,motor}}$, is calculated as

$$P_{\text{m,motor}} = T_{\text{motor}}\omega_{\text{motor}} \tag{4.23}$$

The corresponding electrical power of the motor $P_{\text{e,motor}}$ is defined as

$$P_{\text{e,motor}} = \begin{cases} \dfrac{P_{\text{m,motor}}}{\eta_{\text{m}}(T_{\text{motor}}, \omega_{\text{motor}})}, & \text{if } T_{\text{motor}} \geq 0 \\ P_{\text{m,motor}}\ \eta_{\text{m}}(T_{\text{motor}}, \omega_{\text{motor}}), & \text{if } T_{\text{motor}} < 0 \end{cases} \tag{4.24}$$

Here, η_{m} represents the motor efficiency, which is a function of both motor torque T_{motor} and angular velocity ω_{motor}.

As illustrated in the discretized dynamics (4.7) , the BEV speed v_{h} can be updated at each time step by controlling the traction force F_{t} applied at the wheels. Consequently, the mechanical power required for propulsion is expressed as the product of the traction force and the vehicle velocity:

$$P_{\mathrm{m,BEV}} = F_{\mathrm{t}} v_{\mathrm{h}} \tag{4.25}$$

The relationship between the traction force F_{t} and the motor torque T_{motor} is given by

$$F_{\mathrm{t}} = \frac{T_{\mathrm{motor}} \eta \gamma}{R_{\mathrm{w}}} \tag{4.26}$$

where R_{w} is the wheel radius, η is the gearbox efficiency and γ denotes the gear ratio.

Additionally, the vehicle speed v_{h} is related to the motor's angular speed ω_{motor} through

$$v_{\mathrm{h}} = \frac{\omega_{\mathrm{motor}} R_{\mathrm{w}}}{\gamma} \tag{4.27}$$

Given the mechanical output power P_{out}, the total electrical power consumption P_{t} of the BEV is determined by accounting for the efficiencies of key powertrain components i.e., efficiency of inverter and motor η_{im}, and the efficiency of the battery η_{b}. The electrical power consumption for the discharge and charge modes is calculated as

$$P_{\mathrm{BEV}} = \begin{cases} \dfrac{F_{\mathrm{t}}.v}{\eta_{\mathrm{inv}}(F_{\mathrm{t}}, v)\eta_{\mathrm{bat}}(F_{\mathrm{t}}, v, \zeta)\eta_{\mathrm{mot}}(F_{\mathrm{t}}, v)\eta_{\mathrm{t}}} & \text{if } F_{\mathrm{t}} \geq 0 \\ F_{\mathrm{t}}.v.\eta_{\mathrm{inv}}(F_{\mathrm{t}}, v)\eta_{\mathrm{bat}}(F_{\mathrm{t}}, v, \zeta)\eta_{\mathrm{mot}}(F_{\mathrm{t}}, v)\eta_{\mathrm{t}}, & \text{if } F_{\mathrm{t}} < 0 \end{cases} \tag{4.28}$$

It is important to note that the influence of the state of charge ζ on the instantaneous power consumption $P_{\mathrm{BEV},t}$ is considered negligible over short prediction horizons. Therefore, for the purpose of model predictive control (MPC) design in the subsequent sections, $P_{\mathrm{BEV},t}$ is modeled solely as a function of F_t and v. The battery efficiency map η_{bat}, however, is updated whenever the accumulated SoC deviation exceeds 5%.

Approximation of the BEV Power Consumption Map

In this work, the power consumption map is approximated using a second-order polynomial function of velocity v_h and traction force F_t, expressed as

$$P_{\mathrm{app}}\left(v_{\mathrm{h}}, F_{\mathrm{t}}\right) = p_{00} + p_{10} v_{\mathrm{h}} + p_{01} F_{\mathrm{t}} + p_{11} v_{\mathrm{h}} F_{\mathrm{t}} + p_{20} v_{\mathrm{h}}^2 + p_{02} F_{\mathrm{t}}^2 \tag{4.29}$$

where the coefficients p_{00}, p_{10}, p_{01}, p_{11}, p_{20}, p_{02} are determined by solving a convex optimization problem as defined in (4.30a). The objective of the optimization is to minimize the approximation error between the polynomial function P_{app} and the original power

consumption map (4.28) across the operating range $i = 1, .., R$ for traction force and $j = 1, .., S$ for velocity. Additionally, (4.30b) ensures that P_{app} is positive semidefinite. The optimization problem is solved using CVX semi-definite programming toolbox.

$$\min_{p_{00},p_{10},p_{01},p_{11},p_{20},p_{02}} \sum_{i=1}^{R}\sum_{j=1}^{S} \|Ap - b\|^2 \tag{4.30a}$$

$$\text{s.t.} \quad \begin{bmatrix} p_{02} & p_{11} \\ p_{11} & p_{20} \end{bmatrix} \succcurlyeq 0 \tag{4.30b}$$

$$\text{where,} \quad A = \begin{bmatrix} 1 & v_{\text{h},j} & F_{\text{t},i} & v_{\text{h},j}F_{\text{t},i} & v_{\text{h},j}^2 & F_{\text{t},i}^2 \end{bmatrix}$$

$$b = P_{\text{BEV}}$$

$$x = [p_{00}, p_{10}, p_{01}, p_{11}, p_{20}, p_{02}]^T$$

Figure 4.4 illustrates the BEV's real power consumption map for the full operational range, encompassing both positive and negative traction forces. The same figure also presents the approximated power consumption values for the full map, calculated using coefficients optimized by solving (4.30). Since the real power consumption P_{BEV} varies significantly between positive and negative F_{t}, achieving an accurate approximation across the entire operational range is challenging.

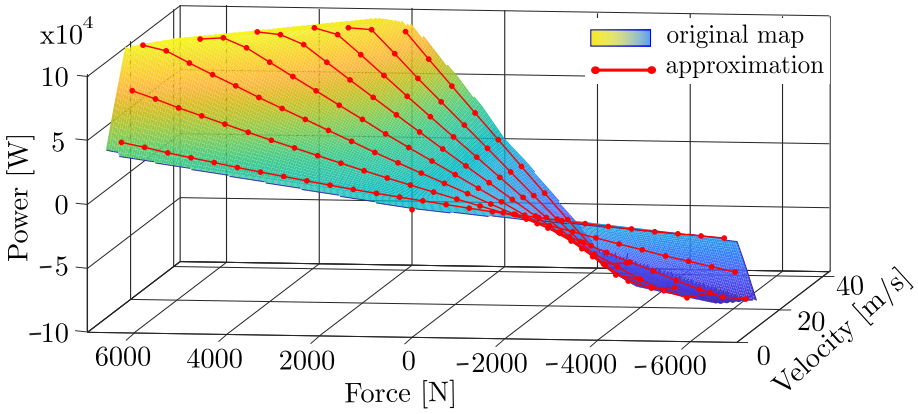


Figure 4.4: Original full map of BEV's power consumption (obtained from [LGW17]) and its approximated power consumption P_{app}.

Conversely, when the coefficients of P_{app} are optimized to approximate only the upper half of the BEV's operational range, where F_{t} is positive as shown in Figure 4.5, the approximation error becomes smaller as highlighted in [JJIG19]. Additionally, a limitation of the full-map approximation is that P_{app} may generate a negative cost for negative F_t in the cost function of the Eco-MPC, prompting it to prioritize recuperation for energy savings. However, excessive recuperation can result in energy losses due to the double utilization of the battery (i.e., repeated charging and discharging). For these

reasons, this work focuses on the upper half of the BEV's operational range (Figure 4.5) to develop the approximation function P_{app} (4.30).

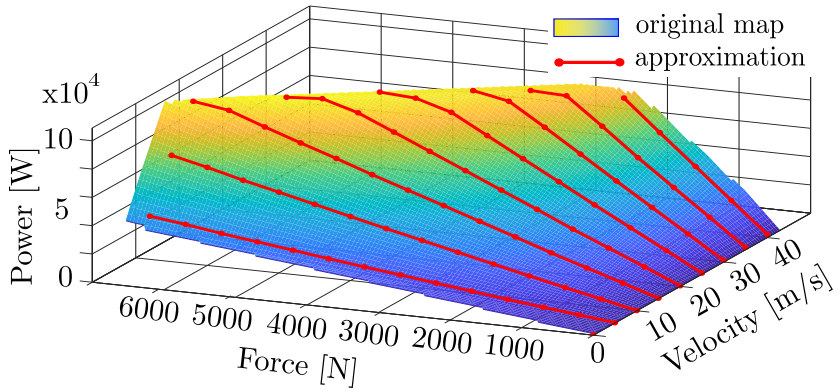


Figure 4.5: Original half map of BEV's power consumption and its approximated power consumption P_{app}.

4.2.2 Problem Formulations for Reference Tracking MPC

After introducing the BEV powertrain model and the approximation of the power consumption map in Section 4.2.1, this section presents the problem formulation for the reference tracking MPC.

Scenario I: Freeway Scenario

Consider a freeway scenario, illustrated in Figure 4.6, where the host vehicle travels in an urban environment, without any influence due to surrounding traffic. To control the host vehicle energy-efficiently in this scenario, the problem formulation for the reference tracking MPC is defined by

$$\min_{F_{\text{t},k},F_{\text{b},k}} \sum_{k=0}^{N-1} P_{\text{app}}\left(v_{\text{h},k}, F_{\text{t},k}\right) + \xi_1 (v_{\text{h},k} - v_{\text{ref},k})^2 + \xi_2 F_{\text{b},k}^2 \tag{4.31a}$$

$$\text{s.t.} \quad v_{\text{h},k+1} = v_{\text{h},k} + \frac{\Delta T}{m_{\text{eq}}}(F_{\text{t},k} - F_{\text{b},k} - F_{\text{r},k} - F_{\text{a},k} - F_{\text{g},k}) \tag{4.31b}$$

$$0 \leq F_{\text{t},k} \leq F_{\text{t,max}} \tag{4.31c}$$

$$0 \leq F_{\text{b},k} \leq F_{\text{b,max}} \tag{4.31d}$$

$$v_{\text{min},k} \leq v_{\text{h},k} \leq v_{\text{max},k} \tag{4.31e}$$

In the cost function (4.31a), the first term minimizes the BEV's power consumption, and is approximated using (4.29). However, relying solely on the first term would not incentivize the host vehicle to move forward, as energy consumption is lowest when the

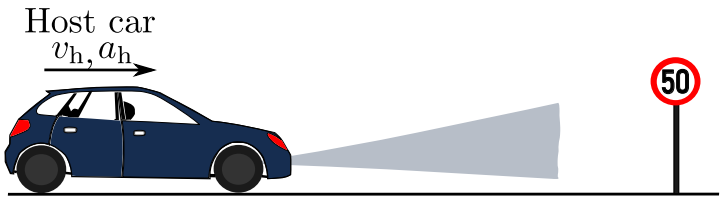


Figure 4.6: Schematic of the host vehicle in a freeway scenario

vehicle is stationary. To address this, the second term introduces a penalty, weighted by ξ_1, for deviations of the host vehicle's velocity v_h from the reference velocity v_ref. In Scenario I, the reference velocity v_ref is determined using the dynamic programming algorithm described in (4.32), which aims to plan an energy efficient optimal velocity profile for the host vehicle based on the route information. The third term penalizes excessive braking force F_b using the weighting factor ξ_2. Additionally, the host vehicle's velocity is updated at each time step using (4.31b). Constraint (4.31c) ensures that the traction force F_t remains non-negative and within its maximum limit $F_\mathrm{t,max}$, which is approximated as $p_3 v_{\mathrm{h},k} + p_4$, where p_3 and p_4 are parameters obtained through line approximation. Similarly, the braking force F_b and host vehicle velocity v_h are bounded by (4.31d) and (4.31e) enforcing the vehicle's physical limitations and road speed limits, respectively.

Reference Speed Generation using Dynamic Programming

In Scenario I, the optimal reference speed v_ref is determined by solving the dynamic programming (DP) optimization problem defined in (4.32) [WGL18]. The objective of this optimization is to balance three competing goals: minimizing energy consumption, reducing trip time, and enhancing driving comfort.

$$\min_{v_{\mathrm{ref},k}} \sum_{k=0}^{N-1} \left\{ t_k P_{\mathrm{bat},k} + \beta t_k + \lambda (T_{\mathrm{motor},k})^2 \right\} \tag{4.32a}$$

$$\text{s.t.} \quad v_0 = v_\mathrm{init} \tag{4.32b}$$

$$\zeta_0 = \zeta_\mathrm{init} \tag{4.32c}$$

$$v_N \in [v_{\mathrm{min},N}, v_{\mathrm{max},N}] \tag{4.32d}$$

$$\zeta_N \in [\zeta_{\mathrm{min},N}, \zeta_{\mathrm{max},N}] \tag{4.32e}$$

$$v_k \in [v_{\mathrm{min},k}, v_{\mathrm{max},k}] \tag{4.32f}$$

$$\zeta_k \in [\zeta_{\mathrm{min},k}, \zeta_{\mathrm{max},k}] \tag{4.32g}$$

$$T_{\mathrm{motor},k} \in [T_\mathrm{min}(v_k), T_\mathrm{max}(v_k)] \tag{4.32h}$$

In the objective function (4.32a), the first term $t_k P_{\mathrm{bat},k}$ represents the energy consumed in segment k, where the total battery power P_bat is calculated as

$$P_\mathrm{bat} = P_\mathrm{e,motor} + P_\mathrm{aux} + P_\mathrm{hydr} + P_\mathrm{cell} \tag{4.33}$$

Here, $P_{\text{e,motor}}$ is the electrical power required by the motor, P_{aux} accounts for the power consumed by auxiliary systems (e.g., pumps, lighting, air conditioning), P_{hydr} denotes the power used for hydraulic braking, and $P_{\text{cell}} = N_{\text{cell}} R_{\text{cell}} i_{\text{cell}}^2$ represents the power losses within the battery due to internal resistance R_{cell}, where N_{cell} is the number of battery cells. The second term βt_k in the cost function represents the cost associated with trip time, while the third term $\lambda (T_{\text{motor},k})^2$ penalizes high motor torque to ensure smooth driving and improve comfort. The weighting parameters β and λ can be adjusted based on driver preferences. Increasing β prioritizes shorter travel time at the expense of higher energy consumption, whereas increasing λ promotes smoother driving by discouraging excessive acceleration and deceleration.

Equations (4.32b) and (4.32c) define the initial conditions of the trip, namely the initial vehicle speed v_{init} and the initial battery state of charge ζ_{init}. The constraints in (4.32d) and (4.32e) ensure that the final speed and battery charge remain within predefined bounds, which are determined by regulatory speed limits $v_{\max}$ and maximum allowable battery state of charge $\zeta_{\max}$, respectively. The constraint in (4.32f) enforces compliance with the speed limits throughout the trip, while (4.32g) prevents both deep discharge and overcharging of the battery. A lower speed bound $v_{\min}$ is also imposed to avoid unreasonably low driving speeds of the host vehicle. Finally, (4.32h) introduces a speed-dependent inequality constraint on the motor torque to reflect the physical limitations of the powertrain, including maximum traction and braking capabilities. It is important to note that the resulting optimal speed profile v_{ref} is generated based on a predefined desired trip time, which may incorporate aspects of the current traffic flow. However, this speed profile does not necessarily guarantee optimal performance under all traffic conditions, as it may not fully capture real-time variations and uncertainties in traffic dynamics.

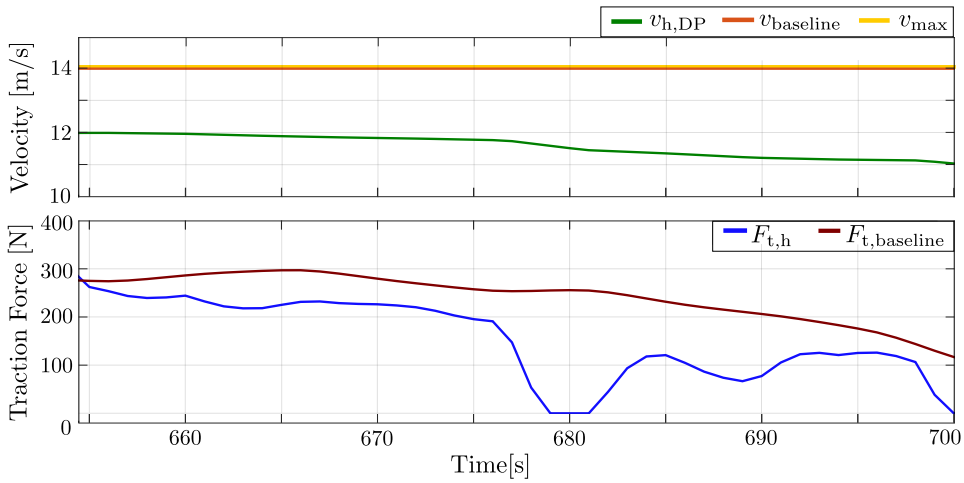


Figure 4.7: Comparison of vehicle behavior under reference tracking MPC and set-speed control in a freeway scenario with reference speed trajectory tracking

To provide a clearer understanding of the host vehicle's behavior under reference tracking

MPC, as formulated in (4.31), a short driving segment from an extended test scenario is selected and illustrated in Figure 4.7. For comparison, a baseline vehicle is considered, which operates in the same freeway scenario using a constant set-speed input corresponding to the maximum allowable road speed $v_{\max}$. As depicted in Figure 4.7, the host vehicle guided by an optimal speed profile computed offline via dynamic programming adopts lower speeds than the baseline. This speed profile is derived by solving (4.32). Consequently, the host vehicle's traction force $F_{\mathrm{t,h}}$ is consistently lower than that of the baseline vehicle $F_{\mathrm{t,baseline}}$, leading to reduced overall energy consumption. These findings underscore the effectiveness of combining offline dynamic programming for determining optimal speed trajectories offline using dynamic programming and tracking these in real-time using reference tracking MPC, particularly in freeway driving scenarios.

Scenario II: Freeway Scenario in the Presence of Traffic Light Signals

Consider a freeway scenario, illustrated in Figure 4.8, where the host vehicle travels in an urban environment with traffic light (TL) signals, without any influence due to surrounding traffic. To control the host vehicle energy-efficiently in this scenario, the problem formulation for the reference tracking MPC is defined by

$$\min_{F_{\mathrm{t},k},F_{\mathrm{b},k},\varepsilon_{1,k}} \sum_{k=0}^{N-1} P_{\mathrm{app}}\left(v_{\mathrm{h},k}, F_{\mathrm{t},k}\right) + \xi_1 (v_{\mathrm{h},k} - v_{\mathrm{ref},k})^2 + \xi_2 F_{\mathrm{b},k}^2 + \xi_3 \varepsilon_{1,k}^2 \tag{4.34a}$$

$$\text{s.t.} \quad (4.31\text{b}), (4.31\text{c}), (4.31\text{d}), (4.31\text{e})$$

$$d_{\mathrm{ITS},k+1} = d_{\mathrm{ITS},k} - \Delta T \left(\tfrac{v_{\mathrm{h},k}+v_{\mathrm{h},k+1}}{2}\right) \tag{4.34b}$$

$$|F_{\mathrm{t},k} - F_{\mathrm{t},k+1}| - \varepsilon_{1,k} \leq \Delta F_{\mathrm{t,max}} \tag{4.34c}$$

In the cost function (4.34a), the first term minimizes the BEV's power consumption, and is approximated using (4.29). To address this, the second term introduces a penalty, weighted by ξ_1, for deviations of the host vehicle's velocity v_{h} from the reference velocity v_{ref}. In Scenario II, the reference velocity v_{ref} is determined using the green wave optimal speed (GWOS) Algorithm 3.1, which aims to assist the host vehicle in crossing signalized intersections during the green phase, as discussed in Section 3.1.2. The third term penalizes excessive braking force F_{b} using the weighting factor ξ_2. It is important to note that the non-negative optimal braking force $F_{\mathrm{b},k}$ consists of both the regenerative and friction braking forces. If $F_{\mathrm{b},k} \leq |-F_{\mathrm{t,max}}|$, then $F_{\mathrm{b},k}$ corresponds entirely to regenerative braking. However, if $F_{\mathrm{b},k} > |-F_{\mathrm{t,max}}|$, then $|-F_{\mathrm{t,max}}|$ represents the regenerative braking component, while the remaining force $F_{\mathrm{b},k} - |-F_{\mathrm{t,max}}|$ corresponds to friction braking. In BEVs, excessive braking-induced deceleration can lead to energy loss, as it contributes to friction braking in which the kinetic energy is converted into heat and dissipated into the environment, which is undesirable for energy efficiency. The fourth term incorporates a soft constraint using a slack variable ε_1 and a weighting factor ξ_3. This term penalizes large variations in traction force F_{t} between successive time steps, thereby reducing jerks and enhancing driving comfort of the passenger in the host vehicle. (4.34c) ensures that

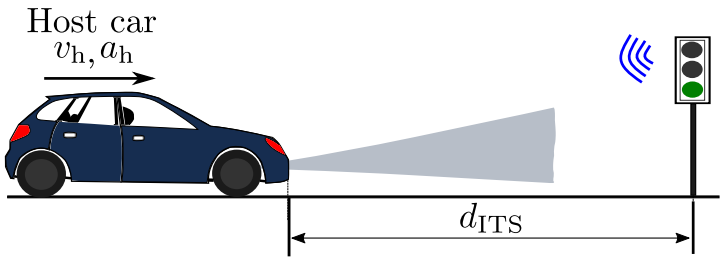


Figure 4.8: Schematic of the host vehicle in a freeway scenario approaching a TL signal

the traction force variation is within a maximum allowable limit $\Delta F_{\mathrm{t,max}}$. Additionally, the host vehicle's velocity is updated at each time step using (4.31b). Constraint (4.31c) ensures that the traction force F_t remains non-negative and within its maximum limit $F_{\mathrm{t,max}}$, which is approximated as $p_3 v_{\mathrm{h},k} + p_4$, where p_3 and p_4 are parameters obtained through line approximation. Similarly, the braking force F_b and host vehicle velocity v_h are bounded by (4.31d) and (4.31e) enforcing the vehicle's physical limitations and road speed limits, respectively.

To evaluate the problem formulation (4.34), a short simulated urban driving route of 2.6 km, featuring four signalized intersections with TL signals $\mathrm{TS}_{\{1-4\}}$ was selected, as shown in Figure 4.9. While $\mathrm{TS}_{\{1,3,4\}}$ operate as deterministic TL signals with fixed cycle timings, $\mathrm{TS}_{\{2\}}$ functions as an actuated TL signal, where the duration of the red phase is dynamically adjusted based on detected traffic conditions. Figure 4.9 illustrates the behavior of the host vehicle equipped with reference tracking MPC while receiving the SPaT information in real-time, and following a green wave optimal speed (GWOS) determined using Algorithm 3.1. Its performance is compared against five baseline BEVs $\mathrm{BS}_{\{1-5\}}$, each traveling at a fixed driver-set speed of 14 m/s, 13 m/s, 12 m/s, 11 m/s, and 10 m/s, respectively. It is observed that the host vehicle equipped with the reference tracking MPC (depicted in blue) effectively plans speed trajectories and successfully crosses both deterministic and actuated TL signals during the green phase. This is achieved by dynamically optimizing the velocity profile based on updated SPaT information. In contrast, the baseline vehicles, which operate at a fixed driver-defined set-speed without access to SPaT data, experience more frequent stops at signalized intersections. Notably, baseline vehicles BS_4, BS_5, and BS_6 required an additional 64, 65, and 66 seconds, respectively, to reach the target distance compared to the host vehicle. To assess the energy efficiency of the proposed control strategy, an accurate electric vehicle longitudinal dynamics model, as described in [LGL14], is employed. The energy savings, denoted as E_S, are computed based on the relative difference in energy consumption between the host vehicle using the reference tracking MPC and a baseline vehicle using a conventional set-speed controller.

$$E_\mathrm{S} = \frac{E_\mathrm{MPC} - E_{\text{set-speed}}}{E_{\text{set-speed}}} \tag{4.35}$$

The results demonstrate that the host vehicle equipped with the reference tracking MPC achieves a maximum energy savings of 30.78% compared to BS_1. Additionally,

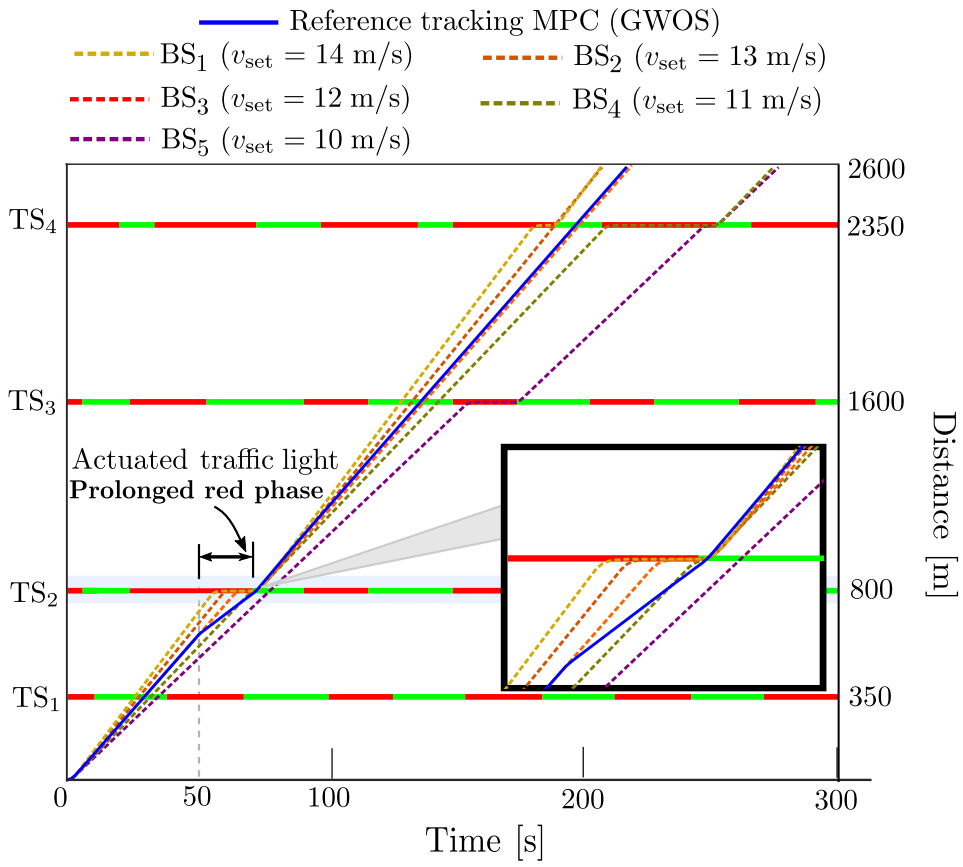


Figure 4.9: Space-time diagram comparing Reference tracking MPC and set-speed controllers

the controller outperforms BS_2, BS_3, and BS_5, yielding energy savings of up to 17.37%, 10.94% and 16.6%, respectively. These improvements are attributed to the MPC's ability to track an optimal speed designed to approach TL signals at green phases, thereby minimizing unnecessary stops and improving energy efficiency. Although the minimum observed energy savings is only 0.3% when compared to BS_3 (driven at a set-speed of 12 m/s), this still highlights a key limitation of conventional set-speed controllers: they do not account for traffic signal timing and thus cannot guarantee uninterrupted passage through signalized intersections. Overall, the findings suggest that the reference tracking MPC is a suitable and energy-efficient control strategy for freeway driving scenarios involving TL signals.

Challenge in Generating Green Wave Optimal Speeds (GWOS)

The results presented in Figure 4.9 highlight the significance of the Algorithm 3.1 and the reference tracking MPC in enabling host vehicle to pass through signalized intersections during green phases. Consider the freeway scenario depicted in Figure 4.10 where

the host vehicle approaches a TL signal with longer red phase. Although, in theory, it is possible to compute a GWOS for such cases often by adopting very low driving speeds, this approach may disrupt overall traffic flow and is therefore undesirable. To mitigate this issue, a minimum allowable road speed $v_{\min}$ is imposed within the green wave algorithm (Algorithm 3.1). However, this constraint may result in situations where no feasible green wave speed can be found, especially for long red light durations. In such cases, the host vehicle is required to stop at the upcoming TL signal.

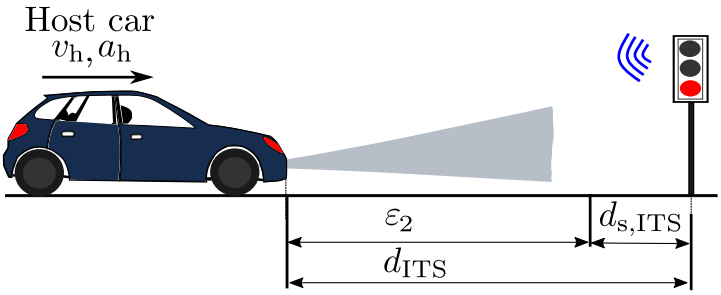


Figure 4.10: Schematic of the host vehicle in a freeway scenario approaching a red TL signal

To address this, the previous reference tracking MPC formulation (4.34) is adapted as follows:

$$\min_{F_{\mathrm{t},k},F_{\mathrm{b},k},\varepsilon_{1,k},\varepsilon_{2,k}} \sum_{k=0}^{N-1} P\left(v_{\mathrm{h},k}, F_{\mathrm{t},k}\right) + \xi_1 (v_{\mathrm{h},k} - v_{\mathrm{ref},k})^2 + \xi_2 F_{\mathrm{b},k}^2 + \xi_3 \varepsilon_{1,k}^2 + \xi_4 \varepsilon_{2,k}^2 \quad (4.36a)$$

$$\text{s.t.} \quad (4.31b), (4.31c), (4.31d), (4.31e), (4.34b)$$

$$d_{\mathrm{ITS},k} \geq 0 \quad (4.36b)$$

$$d_{\mathrm{ITS},k} - \varepsilon_{2,k} \leq d_{\mathrm{s,ITS}} \quad (4.36c)$$

The key modification in this formulation lies in how the reference velocity v_{ref} is handled within the tracking term $\xi_1(v_{\mathrm{h},k} - v_{\mathrm{ref},k})^2$. Specifically, v_{ref} is generated via the dynamic programming (DP) algorithm (4.32), but only when the host vehicle is located outside the predefined intersection horizon $d_{\mathrm{ITS,horizon}}$, as outlined in the switching criteria shown in Figure 3.3. Once within the intersection horizon, the vehicle transitions to a controlled deceleration mode aimed at stopping in an energy-efficient manner. To facilitate this behavior, an additional penalization term $\xi_4 \varepsilon_{2,k}^2$ is introduced in the objective function (4.36a), where $\varepsilon_{2,k}$ represents a slack variable measuring the distance between the host vehicle and the stop line of the TL signal. The associated soft constraint (4.36c) encourages the vehicle to come to a stop as close as possible to the TL stop line $d_{\mathrm{s,ITS}}$. Furthermore, the safety constraint (4.36b) ensures that the vehicle does not cross TL signal during a red signal phase. It is important to note that the reference tracking MPC formulation in (4.36a) simultaneously manages two potentially conflicting objectives: (i) tracking the DP-generated speed profile and (ii) safely stopping at the red light. These objectives are balanced dynamically by activating or deactivating the corresponding weighting parameters ξ_1 and ξ_4, depending on the traffic context.

4.2.3 Quadratic Programming Form with Linear Inequality Constraints

In this subsection, the optimization problem for the reference tracking MPC (4.36) is reformulated as a quadratic programming (QP) problem, consisting of a quadratic objective function subject to linear constraints.

The quadratic objective function J for the reference-tracking MPC is reformulated as

$$J = \min_{u_k} \sum_{k=i}^{i+N-1} P_{\text{app}}\left(v_{\text{h},k}, F_{\text{t},k}\right) + \xi_1 \left(v_{\text{h},k} - v_{\text{ref},k}\right)^2 + \xi_2 F_{\text{b},k}^2 + \xi_3 \varepsilon_{1,k}^2 + \xi_4 \varepsilon_{2,k}^2 \tag{4.37}$$

where, the state and control variables are chosen as $x_k = [v_{\text{h},k}, d_{\text{ITS},k}]^\intercal$ and $u_k = [F_{\text{t},k}, F_{\text{b},k}, \varepsilon_{1,k}, \varepsilon_{2,k}]^\intercal$, respectively. Substituting (4.29) in (4.37) yields

$$\begin{aligned} J = N p_{00} + p_{10} \sum_{k=i}^{i+N-1} v_{\text{h},k} + p_{01} \sum_{k=i}^{i+N-1} F_{t,k} + p_{20} \sum_{k=i}^{i+N-1} v_{\text{h},k}^2 + p_{02} \sum_{k=i}^{i+N-1} F_{\text{t},k}^2 \\ + p_{11} \sum_{k=i}^{i+N-1} F_{t,k} v_{\text{h},k} + \xi_1 \sum_{k=i+1}^{i+N} v_{\text{h},k}^2 - 2\xi_1 \sum_{k=i+1}^{i+N} v_{\text{h},k} v_{\text{ref},k} + \xi_1 \sum_{k=i+1}^{i+N} v_{\text{ref},k}^2 \\ + \xi_2 \sum_{k=i}^{i+N-1} F_{\text{b},k}^2 + \xi_3 \sum_{k=i}^{i+N-1} \varepsilon_{1,k}^2 + \xi_4 \sum_{k=i}^{i+N-1} \varepsilon_{2,k}^2 \end{aligned} \tag{4.38}$$

The quadratic objective function J at the time step i is expressed as

$$\begin{aligned} J_i = \underbrace{p_{00}}_{c_1(i)} + \underbrace{\begin{pmatrix} p_{10} & 0 \end{pmatrix}}_{c_2(i)} \underbrace{\begin{pmatrix} v_{\text{h},i} \\ d_{\text{ITS},i} \end{pmatrix}}_{x_i} + \underbrace{\begin{pmatrix} p_{01} & 0 & 0 & 0 \end{pmatrix}}_{c_3(i)} \underbrace{\begin{pmatrix} F_{t,i} \\ F_{\text{b},i} \\ \varepsilon_{1,i} \\ \varepsilon_{2,i} \end{pmatrix}}_{u_i} \\ + \begin{pmatrix} v_{\text{h},i} \\ d_{\text{ITS},i} \end{pmatrix}^T \underbrace{\begin{pmatrix} p_{20} & 0 \\ 0 & 0 \end{pmatrix}}_{c_4(i)} \begin{pmatrix} v_{\text{h},i} \\ d_{\text{ITS},i} \end{pmatrix} + \begin{pmatrix} F_{\text{t},i} \\ F_{\text{b},i} \\ \varepsilon_{1,i} \\ \varepsilon_{2,i} \end{pmatrix}^T \underbrace{\begin{pmatrix} p_{02} & 0 & 0 & 0 \\ 0 & \xi_2 & 0 & 0 \\ 0 & 0 & \xi_3 & 0 \\ 0 & 0 & 0 & \xi_4 \end{pmatrix}}_{c_5(i)} \begin{pmatrix} F_{\text{t},i} \\ F_{\text{b},i} \\ \varepsilon_{1,i} \\ \varepsilon_{2,i} \end{pmatrix} \\ + \begin{pmatrix} v_{\text{h},i} \\ d_{\text{ITS},i} \end{pmatrix}^T \underbrace{\begin{pmatrix} p_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}}_{c_6(i)} \begin{pmatrix} F_{\text{t},i} \\ F_{\text{b},i} \\ \varepsilon_{1,i} \\ \varepsilon_{2,i} \end{pmatrix} \end{aligned} \tag{4.39}$$

The quadratic objective function J at the time step $i+N$ is given by

$$J_{i+N} = \underbrace{\xi_1 v_{\mathrm{ref},i+N}^2}_{c_1(i+N)} + \underbrace{\begin{pmatrix} -2\xi_1 v_{\mathrm{ref},i+N} & 0 \end{pmatrix}}_{c_2(i+N)} \underbrace{\begin{pmatrix} v_{\mathrm{h},i+N} \\ d_{i+N} \end{pmatrix}}_{x_{i+N}} + \begin{pmatrix} v_{\mathrm{h},i+N} \\ d_{\mathrm{ITS},i+N} \end{pmatrix}^T \underbrace{\begin{pmatrix} p_{20} & 0 \\ 0 & 0 \end{pmatrix}}_{c_4(i+N)} \begin{pmatrix} v_{\mathrm{h},i+N} \\ d_{\mathrm{ITS},i+N} \end{pmatrix} \tag{4.40}$$

The quadratic objective function J for the intermediate time steps $k = (i+1), \ldots, (i+N-1)$ is expressed as

$$\begin{aligned} J_i = {} & \underbrace{p_{00} + \xi_1 v_{\mathrm{ref},k}^2}_{c_1(k)} + \underbrace{\begin{pmatrix} p_{10} - 2\xi_1 v_{\mathrm{ref},k} & 0 \end{pmatrix}}_{c_2(k)} \underbrace{\begin{pmatrix} v_{\mathrm{h},k} \\ d_{\mathrm{ITS},k} \end{pmatrix}}_{x_k} + \underbrace{\begin{pmatrix} p_{01} & 0 & 0 & 0 \end{pmatrix}}_{c_3(k)} \underbrace{\begin{pmatrix} F_{t,k} \\ F_{\mathrm{b},k} \\ \varepsilon_{1,k} \\ \varepsilon_{2,k} \end{pmatrix}}_{u_k} \\ & + \begin{pmatrix} v_{\mathrm{h},k} \\ d_{\mathrm{ITS},k} \end{pmatrix}^T \underbrace{\begin{pmatrix} p_{20} + \xi_1 & 0 \\ 0 & 0 \end{pmatrix}}_{c_4(k)} \begin{pmatrix} v_{\mathrm{h},k} \\ d_{\mathrm{ITS},k} \end{pmatrix} + \begin{pmatrix} F_{\mathrm{t},k} \\ F_{\mathrm{b},k} \\ \varepsilon_{1,k} \\ \varepsilon_{2,k} \end{pmatrix}^T \underbrace{\begin{pmatrix} p_{02} & 0 & 0 & 0 \\ 0 & \xi_2 & 0 & 0 \\ 0 & 0 & \xi_3 & 0 \\ 0 & 0 & 0 & \xi_4 \end{pmatrix}}_{c_5(k)} \begin{pmatrix} F_{\mathrm{t},k} \\ F_{\mathrm{b},k} \\ \varepsilon_{1,k} \\ \varepsilon_{2,k} \end{pmatrix} \\ & + \begin{pmatrix} v_{\mathrm{h},k} \\ d_{\mathrm{ITS},k} \end{pmatrix}^T \underbrace{\begin{pmatrix} p_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}}_{c_6(k)} \begin{pmatrix} F_{\mathrm{t},k} \\ F_{\mathrm{b},k} \\ \varepsilon_{1,k} \\ \varepsilon_{2,k} \end{pmatrix} \end{aligned} \tag{4.41}$$

The quadratic objective function J for all time steps can be written as

$$\begin{aligned} J = {} & \textstyle\sum_{k=i}^{i+N} c_1(k) + c_2(i)x_i + x_i^T c_4(i) x_i + \underbrace{\begin{pmatrix} c_2(i+1) & \ldots & c_2(i+N) \end{pmatrix}}_{C_2} \begin{pmatrix} x_{i+1} \\ \vdots \\ x_{i+N} \end{pmatrix} \\ & + \begin{pmatrix} x_{i+1} \\ \vdots \\ x_{i+N} \end{pmatrix}^T \underbrace{\begin{pmatrix} c_4(i+1) & & \\ & \ddots & \\ & & c_4(i+N) \end{pmatrix}}_{C_4} \begin{pmatrix} x_{i+1} \\ \vdots \\ x_{i+N} \end{pmatrix} \\ & + \underbrace{\begin{pmatrix} c_3(i) + x_i^T c_6(i) & c_3(i+1) & \ldots & c_3(i+N-1) \end{pmatrix}}_{C_3} \begin{pmatrix} u_i \\ \vdots \\ u_{i+N-1} \end{pmatrix} \\ & + \begin{pmatrix} u_i \\ \vdots \\ u_{i+N-1} \end{pmatrix}^T \underbrace{\begin{pmatrix} c_5(i) & & \\ & \ddots & \\ & & c_5(i+N-1) \end{pmatrix}}_{C_5} \begin{pmatrix} u_i \\ \vdots \\ u_{i+N-1} \end{pmatrix} \\ & + \begin{pmatrix} x_{i+1} \\ \vdots \\ x_{i+N} \end{pmatrix}^T \underbrace{\begin{pmatrix} 0 & c_6(i+1) & & \\ & & \ddots & c_6(i+N-1) \\ & & & 0 \end{pmatrix}}_{C_6} \begin{pmatrix} u_i \\ \vdots \\ u_{i+N-1} \end{pmatrix} \end{aligned} \tag{4.42}$$

$$= \sum_{k=i}^{i+N} c_1(k) + c_2(i)x_i + x_i^T c_4(i)x_i + C_2X_i + X_i^T C_4 X_i + C_3U_i + U_i^T C_5 U_i + X_i^T C_6 U_i \tag{4.43}$$

From the prediction model (4.16), substituting X_i in (4.43) yields

$$\begin{aligned} J = \underbrace{\sum_{k=i}^{i+N} c_1(k) + c_2(i)x_i + x_i^T c_4(i)x_i}_{f_c} &+ C_2\left(\Phi x_i + \Gamma U_i + \Gamma_w W_i\right) \\ &+ \left(\Phi x_i + \Gamma U_i + \Gamma_w W_i\right)^T C_4 \left(\Phi x_i + \Gamma U_i + \Gamma_w W_i\right) \\ &+ C_3U_i + U_i^T C_5 U_i + \left(\Phi x_i + \Gamma U_i + \Gamma_w W_i\right)^T C_6 U_i \end{aligned} \tag{4.44}$$

Furthermore, substituing $L = \Phi x_i + \Gamma_w W_i$ into (4.44) yields

$$\begin{aligned} J = f_c + C_2\left(\Gamma U_i + L\right) + \left(\Gamma U_i + L\right)^T C_4 \left(\Gamma U_i + L\right) \\ + C_3U_i + U_i^T C_5 U_i + \left(\Gamma U_i + L\right)^T C_6 U_i \end{aligned} \tag{4.45}$$

Rearranging the terms in (4.45) results in

$$\begin{aligned} J = f_c + L^T C_4 L + C_2 L + \left(2L^T C_4 \Gamma + C_2\Gamma + C_3 + L^T C_6\right) U_i \\ + U_i^T \left(C_5 + \Gamma^T C_4 \Gamma + \Gamma^T C_6\right) U_i \end{aligned} \tag{4.46}$$

Equation (4.46) can now be expressed in the standard quadratic programming (QP) objective function form as follows

$$\min_{U_k} J = \min_{U_k} \frac{1}{2} U_k^T H U_k + f^T U_k \tag{4.47}$$

where $H = 2\left(C_5 + \Gamma^T C_4 \Gamma + \Gamma^T C_6\right)$ and $f^T = 2L^T C_4 \Gamma + C_2\Gamma + C_3 + L^T C_6$.

Linear Inequality MPC Constraints

The constraints discussed in (4.36) can be rearranged into a standard form as

$$\begin{aligned} -v_{\mathrm{h},k} &\leq -v_{\min,k} \\ v_{\mathrm{h},k} &\leq v_{\max,k} \\ -p_3 v_{\mathrm{h},k} + F_{\mathrm{t},k} &\leq p_4 \\ -F_{\mathrm{t},k} &\leq 0 \\ F_{\mathrm{b},k} &\leq F_{\mathrm{b,max}} \\ -F_{\mathrm{b},k} &\leq 0 \\ \Delta F_{\mathrm{t},k} - \varepsilon_{1,k} &\leq \Delta F_{\max} \\ -\Delta F_{\mathrm{t},k} - \varepsilon_{1,k} &\leq \Delta F_{\max} \\ -d_{\mathrm{ITS},k} &\leq 0 \\ d_{\mathrm{ITS},k+1} - \varepsilon_{2,k} &\leq d_{\mathrm{s,ITS}} \end{aligned} \tag{4.48}$$

where, $\Delta F_{\mathrm{t},k} = F_{\mathrm{t},k} - F_{\mathrm{t},k-1}$.

It is important to note that the state constraints at time step i and the control constraints at time step $i+N$ are not necessary and can therefore be omitted. The constraints at time step i can be written as

$$\underbrace{\begin{pmatrix} 0 & 0 \\ -p_3 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}}_{M_i} \underbrace{\begin{pmatrix} v_{\mathrm{h},i} \\ d_{\mathrm{ITS},i} \end{pmatrix}}_{x_i} + \underbrace{\begin{pmatrix} -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}}_{E_i} \underbrace{\begin{pmatrix} F_{\mathrm{t},i} \\ F_{\mathrm{b},i} \\ \varepsilon_{1,i} \\ \varepsilon_{2,i} \end{pmatrix}}_{u_i} \leq \underbrace{\begin{pmatrix} 0 \\ p_4 \\ 0 \\ F_{\mathrm{b,max}} \end{pmatrix}}_{b_i} \tag{4.49}$$

The constraint on the variation of the traction force is considered separately and is further presented in (4.53). The constraints for the time steps $k = i+1, \ldots, i+N-1$ are formulated as

$$\underbrace{\begin{pmatrix} -1 & 0 \\ 1 & 0 \\ 0 & 0 \\ -p_3 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -1 \\ 0 & 1 \end{pmatrix}}_{M_k} \underbrace{\begin{pmatrix} v_{\mathrm{h},k} \\ d_{\mathrm{ITS},k} \end{pmatrix}}_{x_k} + \underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{pmatrix}}_{E_k} \underbrace{\begin{pmatrix} F_{\mathrm{t},k} \\ F_{\mathrm{b},k} \\ \varepsilon_{1,k} \\ \varepsilon_{2,k} \end{pmatrix}}_{u_k} \leq \underbrace{\begin{pmatrix} -v_{\min} \\ v_{\max} \\ 0 \\ p_4 \\ 0 \\ F_{\mathrm{b,max}} \\ 0 \\ d_{s,\mathrm{ITS}} \end{pmatrix}}_{b_k} \tag{4.50}$$

The constraints for the final time step $i+N$ are given by

$$\underbrace{\begin{pmatrix} -1 & 0 \\ 1 & 0 \\ 0 & -1 \\ 0 & 1 \end{pmatrix}}_{M_{i+N}} \underbrace{\begin{pmatrix} v_{\mathrm{h},i+N} \\ d_{\mathrm{ITS},i+N} \end{pmatrix}}_{x_{i+N}} \leq \underbrace{\begin{pmatrix} -v_{\min} \\ v_{\max} \\ 0 \\ d_{s,\mathrm{ITS}} \end{pmatrix}}_{b_{i+N}} \tag{4.51}$$

For the prediction horizon N, the constraint equations can be written as

$$\underbrace{\begin{pmatrix} M_i \\ 0 \\ \vdots \\ 0 \end{pmatrix}}_{D'_k} x_i + \underbrace{\begin{pmatrix} 0 & \ldots & 0 \\ M_{i+1} & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & M_{i+N} \end{pmatrix}}_{\mathcal{M}'_k} \underbrace{\begin{pmatrix} x_{i+1} \\ x_{i+2} \\ \vdots \\ x_{i+N} \end{pmatrix}}_{X_k}$$
$$+ \underbrace{\begin{pmatrix} E_i & \ldots & 0 \\ \vdots & \ddots & \ldots \\ 0 & \ldots & E_{i+N-1} \\ 0 & 0 & 0 \end{pmatrix}}_{\mathcal{E}'_k} \underbrace{\begin{pmatrix} u_i \\ u_{i+1} \\ \vdots \\ u_{i+N-1} \end{pmatrix}}_{U_k} \leq \underbrace{\begin{pmatrix} b_i \\ b_{i+1} \\ \vdots \\ b_{i+N} \end{pmatrix}}_{\mathcal{B}'_k} \tag{4.52}$$

There are $2n$ constraints governing the variation in traction force, given by

$$\underbrace{\begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}}_{D_k''} x_i + \underbrace{\begin{pmatrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{pmatrix}}_{\mathcal{M}_k''} \underbrace{\begin{pmatrix} x_{i+1} \\ \vdots \\ x_{i+N} \end{pmatrix}}_{X_k} + \underbrace{\begin{pmatrix} h_1 & 0 & 0 & \dots & 0 \\ h_2 & 0 & 0 & \dots & 0 \\ 0 & h_2 & 0 & \dots & 0 \\ \vdots & \dots & \ddots & \dots & \vdots \\ 0 & \dots & 0 & h_2 & 0 \\ 0 & \dots & 0 & 0 & h_2 \end{pmatrix}}_{\varepsilon_k''} \underbrace{\begin{pmatrix} u_i \\ u_{i+1} \\ \vdots \\ u_{i+N-1} \end{pmatrix}}_{U_k} \leq \underbrace{\begin{pmatrix} F_{\mathrm{t},i-1} + \Delta F_{\max} \\ -F_{\mathrm{t},i-1} + \Delta F_{\max} \\ \Delta F_{\max} \\ \Delta F_{\max} \\ \vdots \\ \Delta F_{\max} \\ \Delta F_{\max} \end{pmatrix}}_{\mathcal{B}_k''} \tag{4.53}$$

where the matrices $h1$ and $h2$ are defined as

$$h_1 = \begin{pmatrix} 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{4.54}$$

$$h_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \end{pmatrix} \tag{4.55}$$

and the applied traction force $F_{\mathrm{t},i-1}$ is known from the previous time step. In the case when $k = 1$, the initial traction force is $F_{\mathrm{t},0} = 0$.

The total set of constraints for the optimization problem is obtained by combining (4.52) and (4.53) and rewriting them as

$$D_k x_k + \mathcal{M}_k X_k + \mathcal{E}_k U_k \leq \mathcal{B}_k \tag{4.56}$$

where,

$$D_k = \begin{pmatrix} D_k' \\ D_k'' \end{pmatrix}, \mathcal{M}_k = \begin{pmatrix} \mathcal{M}_k' \\ \mathcal{M}_k'' \end{pmatrix}, \mathcal{E}_k = \begin{pmatrix} \mathcal{E}_k' \\ \mathcal{E}_k'' \end{pmatrix} \text{ and } \mathcal{B}_k = \begin{pmatrix} \mathcal{B}_k' \\ \mathcal{B}_k'' \end{pmatrix} \tag{4.57}$$

Now, substituting the prediction model equation from (4.16) yields,

$$D_k x_k + \mathcal{M}_k \left(\Phi x_k + \Gamma U_k + \Gamma_w W_k\right) + \mathcal{E}_k U_k \leq \mathcal{B}_k \tag{4.58}$$

Equation (4.58) must be reformulated to obtain the standard QP inequality constraint form as follows

$$\underbrace{\left(\mathcal{M}_k \Gamma + \mathcal{E}_k\right)}_{A_{\mathrm{ieq}}} U_k \leq \underbrace{\mathcal{B}_k - D_k x_k - \mathcal{M}_k \left(\Phi x_k + \Gamma_w W_k\right)}_{b_{\mathrm{ieq}}} \tag{4.59}$$

Finally, combining the quadratic objective function formulated in (4.47) with the inequality constraint form in (4.59) results in the quadratic programming (QP) problem, expressed as

$$\begin{aligned} \min_{U_k} \quad & \frac{1}{2} U_k^T H U_k + f^T U_k \\ \text{subject to} \quad & A_{\text{ieq}} U_k \leq b_{\text{ieq}} \end{aligned} \tag{4.60}$$

4.2.4 Problem Formulations for Car-Following MPC

The previous subsections introduced the problem formulations for reference tracking MPC, with a particular focus on freeway driving in urban environments without surrounding traffic. In this subsection, the focus shifts to the formulation of the car-following MPC problem, specifically addressing scenarios in which the host vehicle encounters a leader vehicle in urban settings.

Scenario I: Typical Car-Following Scenario

As illustrated in Figure 4.11, in a typical car-following scenario, a host car tracks a leader vehicle. To control the host car in an energy-efficient manner, an optimization problem

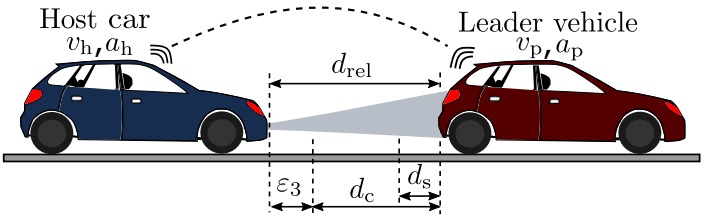


Figure 4.11: Schematic of a typical car-following scenario

in the time domain, as described in (4.61), is proposed.

$$\min_{F_{t,k}, F_{b,k}, \varepsilon_{1,k}, \varepsilon_{3,k}} \sum_{k=i}^{i+N-1} P_{\text{app}}\left(v_{\text{h},k}, F_{\text{t},k}\right) + \xi_1 \varepsilon_{3,k}^2 + \xi_2 F_{\text{b},k}^2 + \xi_3 \varepsilon_{1,k}^2 \tag{4.61a}$$

$$\text{s.t.} \qquad (4.31\text{b}), (4.31\text{c}), (4.31\text{d}), (4.31\text{e}), (4.34\text{c})$$

$$d_{\text{rel},k+1} = d_{\text{rel},k} + \Delta T \left(\frac{v_{\text{p},k} + v_{\text{p},k+1}}{2} - \frac{v_{\text{h},k} + v_{\text{h},k+1}}{2} \right) \tag{4.61b}$$

$$d_{\text{rel},k} \geq \underbrace{d_{\min} + h_{\min} v_{\text{h},k}}_{d_\text{s}} \tag{4.61c}$$

$$d_{\text{rel},k} \leq \underbrace{d_{\min} + h_{\text{comfort}} v_{\text{h},k}}_{d_\text{c}} + \varepsilon_{3,k} \tag{4.61d}$$

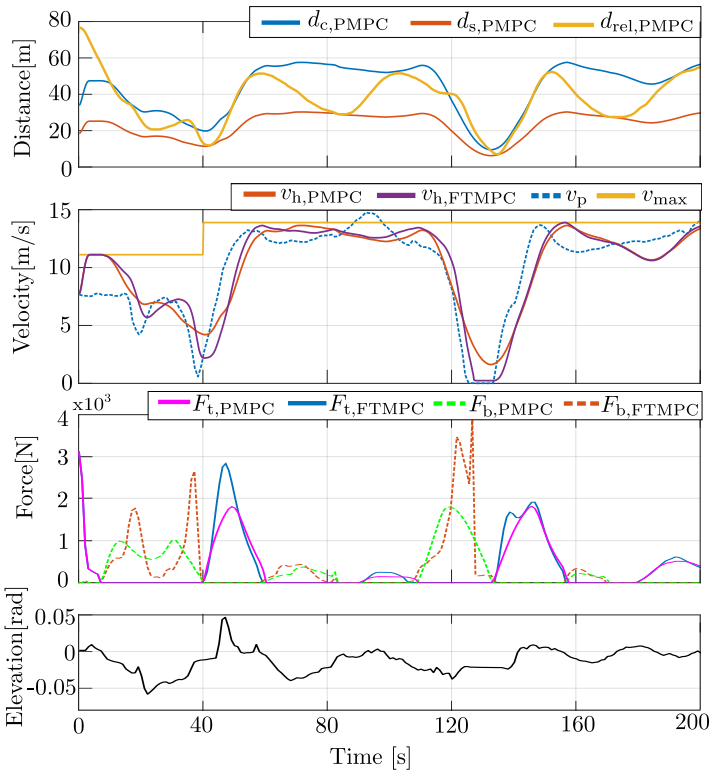


Figure 4.12: Performance of the car-following MPC while tracking a leader vehicle

The objective function aims to minimize energy consumption (term 1), penalize excessive braking (term 3), and reduce excessive jerk (term 4). In the second term, the slack variable $\varepsilon_{3,k}$ is penalized to propel the host car while ensuring the host car remains within the comfort region. The relative distance to the lead car is calculated using (4.36b), and the host car's velocity is updated at each time step based on (4.31b), while the soft constraint (4.36c) encourages the host vehicle to move forward and stay within the comfort zone behind the leader. The remaining constraints, (4.31c), (4.31d), (4.31e), and (4.34c), have been discussed in previous problem formulations.

In this section, the performance of the car-following MPC with problem formulation (4.61) is evaluated while tracking a leader vehicle, whose velocity v_p was recorded from a real-world urban driving test. For comparison, two established benchmark control approaches, namely prescient model predictive control (PMPC) and frozen-time model predictive control (FTMPC) are considered, as described in [RBC+10]. The key distinction between these approaches lies in their handling of the leader vehicle's velocity: PMPC has perfect knowledge of the leader vehicle's future velocities within the prediction horizon, whereas FTMPC assumes that the leader vehicle maintains a constant velocity throughout the horizon. The performance of both PMPC and FTMPC is evaluated for a prediction horizon N=25 (5 s), and their behavior is illustrated in Figure 4.12. The results indicate that the proposed controller with PMPC and FTMPC, success-

fully tracks the leader vehicle's velocity v_p while adhering to the road speed limit v_max. Furthermore, the inter-vehicle distance d_rel to the leader vehicle is controlled robustly, ensuring that the host vehicle remains within the comfort region d_c while strictly maintaining the required safe distance d_s from the leader vehicle. As illustrated in the figure, the host vehicle operating under PMPC applies lower traction force F_t and braking force F_b compared to the FTMPC approach.

Moreover, the simulation results indicated that car-following MPC with PMPC can achieve energy savings of up to 9.2 %, whereas FTMPC achieves up to 7.8 %, both relative to the leader vehicle. The observed improvement in energy efficiency with PMPC is primarily attributed to its ability to utilize previewed velocity information of the leader vehicle. This enables the host vehicle to generate an energy-optimal velocity trajectory characterized by smoother acceleration and deceleration patterns, thereby outperforming the leader vehicle in terms of energy consumption.

Scenario II: Car-Following in the Presence of Traffic Light Signals

As illustrated in Figure 4.13, in urban environments, the speed trajectory of the host vehicle is influenced not only by the behavior of the leader vehicle but also by traffic light (TL) signals. Consider a scenario where the host vehicle is following a leader vehicle that crosses a TL signal during its green phase, while the signal turns red before the host vehicle reaches the intersection. In this case, the host vehicle must discontinue tracking the leader and decelerate to stop at the TL signal. To effectively handle such a scenario, the car-following MPC formulation presented in (4.61) must be extended to include additional terms that account for the presence of TL signals.

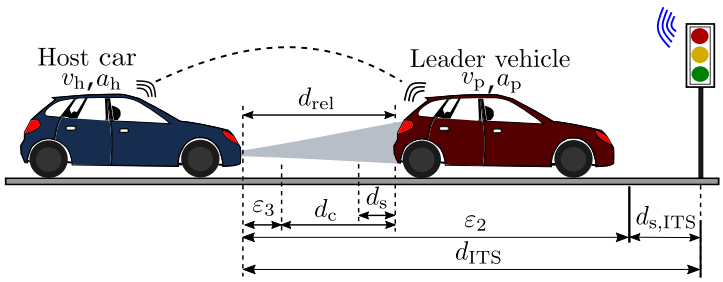


Figure 4.13: Schematic of a car-following scenario in the presence of TL signals

The optimization problem for the aforementioned scenario is formulated as

$$\min_{F_{t,k},F_{b,k},\varepsilon_{1,k},\varepsilon_{2,k},\varepsilon_{3,k}} \sum_{k=i}^{i+N-1} P_\mathrm{app}\left(v_{\mathrm{h},k}, F_{\mathrm{t},k}\right) + \xi_1\varepsilon_{3,k}^2 + \xi_2 F_{\mathrm{b},k}^2 + \xi_3\varepsilon_{1,k}^2 + \xi_4\varepsilon_{2,k}^2 \tag{4.62a}$$

$$\text{s.t.} \quad (4.31\text{b}), (4.34\text{b}), (4.61\text{b}), (4.31\text{c}), (4.31\text{d}), (4.34\text{c}), (4.36\text{b}), (4.36\text{c}), (4.61\text{c}), (4.61\text{d})$$

$$v_{\mathrm{min},k} \leq v_{\mathrm{h},k} \leq \min(v_{\mathrm{max},k}, v_{\mathrm{ref},k}) \tag{4.62b}$$

Here, the optimization problem minimizes an objective function similar to that in (4.61a). However, account for the presence of TL signals, an additional penalty term $\xi_4 \varepsilon_{2,k}^2$, is introduced in the objective function (4.62a). Here, $\varepsilon_{2,k}$ is a slack variable representing the distance between the host vehicle and the stop line of the TL signal. The associated constraints are incorporated into the optimization problem in the same manner as in the previous formulations. Additionally, the physical limitation on the vehicle speed v_{h} is enforced by constraint (4.62b) to promote conservative car-following, where v_{ref} is the reference speed computed using the dynamic programming algorithm (4.32). In the following section, the performance of the car-following MPC will be evaluated in conjunction with the reference tracking MPC.

4.2.5 Performance Evaluation of Eco-MPCs

In this subsection, the performance of the proposed Eco-MPCs, namely, the reference tracking MPC and the car-following MPC, designed for battery electric vehicles is evaluated through simulations under realistic urban driving conditions, including surrounding traffic and TL signals. The switching logic described in Chapter 3 (Section 3.1.2) is employed to dynamically select the appropriate controller based on the driving situation. For benchmarking purposes, a baseline vehicle is introduced with three notable distinctions: (i) While the host vehicle equipped with Eco-MPCs receives real-time SPaT data for upcoming TL signals, the baseline controller has access only to the current signal phase. (ii) The baseline vehicle employs a FTMPC approach, whereas the host vehicle assumes perfect knowledge of the leader vehicle's future velocities using PMPC approach. (iii) In freeway scenarios, the baseline vehicle operates with a constant set speed equal to the maximum allowable speed limit, $v_{\max}$, whereas the host vehicle determines an energy-optimal speed profile using dynamic programming.

As shown in Figure 4.14, both the host and baseline vehicles initially track the same leader vehicle velocity profile v_{p}, which was recorded from a real-world urban driving test. The host vehicle begins by employing the car-following MPC and successfully tracks the leader vehicle until approximately 108 s. At this point, the leader crosses a TL signal during its green phase, while the signal turns red before the host vehicle reaches the intersection. Consequently, the car-following MPC stops tracking the leader vehicle and decelerates to bring the vehicle to a stop, which increases the inter-vehicle distance d_{rel}. Once d_{rel} exceeds the sensor range threshold of 100 m at around 130 s, the controller switches to the reference tracking MPC. Upon the TL signal turning green, the host vehicle resumes travel using the optimal speed trajectory generated by dynamic programming until the leader vehicle re-enters the sensor range at 170 s, there afer switches to car-following MPC. Meanwhile, the baseline vehicle continues traveling at a constant speed $v_{\max}$ and encounters the leader vehicle, and slows down to a complete stop. (5 s).

Between 190 s and 260 s, d_{rel} frequently exceeds the desired following region d_{c}, primarily

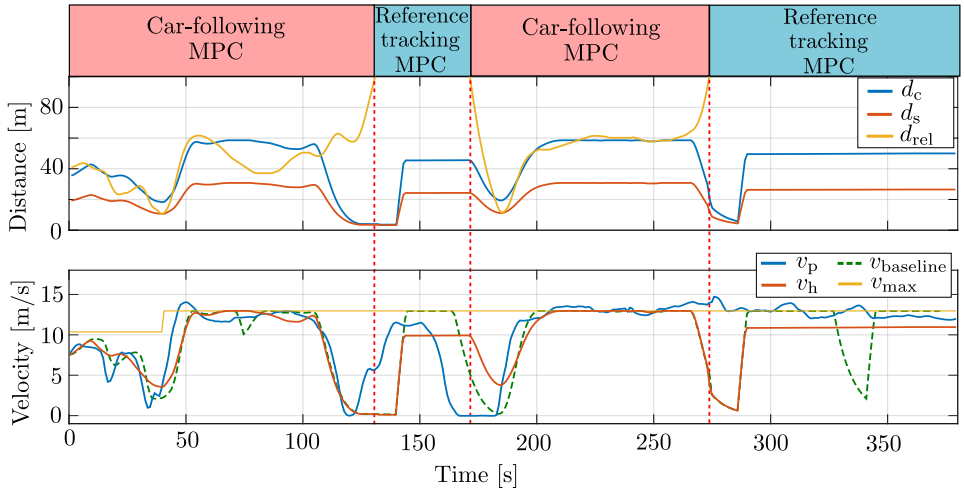


Figure 4.14: Comparison of the simulation results between Eco-MPCs and baseline vehicle

due to the leader vehicle frequently exceeding the speed limit $v_{\max}$. The host vehicle, in contrast, consistently respects the speed limit. At 266 s, the leader vehicle again crosses a TL signal during the green phase, while the host vehicle is forced to decelerate for the red TL signal. The resulting increase in d_{rel} prompts another switch to the reference tracking MPC at 274 s. After the signal turns green, the host vehicle follows a green-wave optimized velocity to pass the next intersection during the green phase. In contrast, the baseline vehicle travels at $v_{\max}$, and is required to decelerate at the TL signal at 330 s due to lacking of SPaT data. Overall, the host vehicle equipped with Eco-MPCs demonstrated superior performance, achieving a 12.8% reduction in energy consumption compared to the baseline controller for a prediction horizon of $N = 25$ steps

The execution times of the proposed Eco-MPC strategies were evaluated using the MATLAB R2016a profiler. The evaluations were performed on a Windows 10 PC equipped with an Intel Core i7-7500U CPU running at 2.70 GHz clock frequency, with 12 GB of RAM. The average computation times per control step were measured as 15.2 ms for reference tracking MPC and 14.9 ms for the car-following MPC. Given that these values are significantly lower than the selected sampling interval (Δ_T=0.2 s), the results indicate the potential for real-time implementation, even on embedded hardware.

4.3 Eco-HMPC for City Buses with Conventional ICE

This section focuses on extending the previously formulated problem formulations for conventional ICE vehicles, especially city buses. A particularly effective fuel-saving

technique for such vehicles is the pulse and glide (PnG) strategy. In this approach, the vehicle accelerates to a speed above the desired cruising speed during the pulse phase and then coasts down to a lower speed during the glide phase, with the engine turned off. When the engine is off, no power is transmitted, effectively reducing fuel consumption. Consequently, the objective is to implement an optimal engine switching control strategy that turns the engine off in scenarios such as coasting or standstill conditions (e.g., during a red traffic light phase). To realize this control logic, the reference tracking and car-following MPCs introduced earlier must be extended by incorporating a binary state variable that governs the engine's ON and OFF status. This results in a hybrid model predictive control (HMPC) framework, which is well-suited for managing the interaction between continuous vehicle dynamics and discrete engine states.

4.3.1 Approximation of ICE Power Consumption Map

To represent the engine speed-torque characteristics of a city bus, a reference engine map OM668 of a passenger vehicle from the ADVISOR library is chosen and scaled using a simplified procedure. This scaling process involves determining two scaling factors: the torque scaling factor $f_{\mathrm{T,scale}}$ and the angular velocity scaling factor $f_{\mathrm{w,scale}}$. First, $f_{\mathrm{T,scale}}$ is computed by dividing the maximum torque of the target (scaled) engine $T_{\mathrm{e,max,new}}$ by the maximum torque of the reference engine $T_{\mathrm{e,max,ref}}$, as defined in Equation (4.63a). The scaled torque values $T_{\mathrm{e,new}}$ are then obtained by multiplying the reference torque values $T_{\mathrm{e,ref}}$ by $f_{\mathrm{T,scale}}$. Similarly, the angular velocity scaling factor $f_{\mathrm{w,scale}}$ is calculated by dividing the maximum angular velocity of the reference engine map $w_{\mathrm{e,max,ref}}$ by that of the scaled engine $w_{\mathrm{e,max,new}}$, as shown in Equation (4.63b). The scaled angular velocity values $w_{\mathrm{e,new}}$ are subsequently determined by dividing the reference angular velocities $w_{\mathrm{e,ref}}$ by $f_{\mathrm{w,scale}}$.

$$f_{\mathrm{T,scale}} = \frac{T_{\mathrm{e,max,new}}}{T_{\mathrm{e,max,ref}}} \implies T_{\mathrm{e,new}} = f_{\mathrm{T,scale}} T_{\mathrm{e,ref}} \tag{4.63a}$$

$$f_{\mathrm{w,scale}} = \frac{w_{\mathrm{e,max,ref}}}{w_{\mathrm{e,max,new}}} \implies w_{\mathrm{e,new}} = \frac{w_{\mathrm{e,ref}}}{f_{\mathrm{w,scale}}} \tag{4.63b}$$

The updated engine efficiency $\eta_{\mathrm{e,new}}$ is computed using

$$\eta_{\mathrm{e,new}} = \frac{T_{\mathrm{e,new}} w_{\mathrm{e,new}}}{\dot{m}_{\mathrm{e,ref}}(w, T)\ H_{\mathrm{u}}} \tag{4.64}$$

Here, H_{u} represents the lower heating value of diesel fuel, while $T_{\mathrm{e,new}}$ and $w_{\mathrm{e,new}}$ are the scaled engine torque and angular velocity, respectively, obtained from equations (4.63a) and (4.63b). The fuel flow rate $\dot{m}_{\mathrm{e,ref}}(w, T)$ is taken from the reference engine's fuel consumption map.

Subsequently, the engine power map P_{e} is calculated using

$$P_{\mathrm{e}} = \frac{F_{\mathrm{t}} v_{\mathrm{h}}}{\eta_{\mathrm{e,new}} \eta_{\mathrm{t}}} \tag{4.65}$$

where η_t denotes the transmission efficiency, and F_t and v_h represent the traction force and vehicle velocity, respectively.

The city bus is equipped with a 4-speed automatic transmission. In this study, a rule-based gear selection strategy is implemented with the primary objective of operating the internal combustion engine (ICE) within its most efficient region. This strategy divides the power consumption map into four distinct regions, as illustrated in Figure 4.15.

To approximate the ICE power consumption for each gear, a convex optimization problem is formulated and solved, as defined in (4.30a). In this formulation, power consumption is expressed as a function of the vehicle velocity v_h and traction force F_t, given by

$$f_{app}(v_h, F_t) = x_{00} + x_{10}v_h + x_{01}F_t + x_{20}v_h^2 + x_{02}F_t^2 + x_{11}v_hF_t \tag{4.66}$$

where x_{00}, x_{10}, x_{01}, x_{20}, x_{02}, x_{11} are the polynomial coefficients determined during optimization. The resulting approximation of the power consumption map is depicted in Figure 4.15.

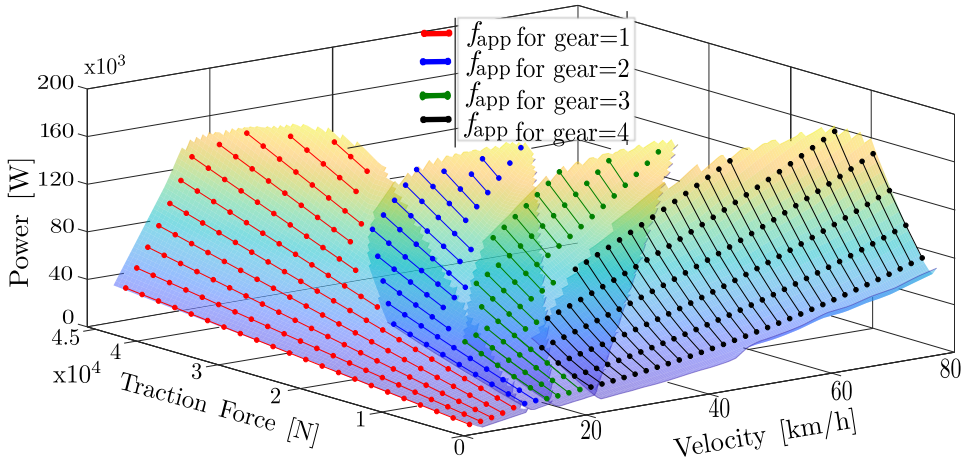


Figure 4.15: Approximation of ICE power consumption map

It is important to note that approximating the nonlinear power consumption map using (4.66) introduces some inaccuracies, particularly at low traction force values, as also highlighted in [JCG20]. For example, when the traction force F_t is zero, the approximated power consumption f_{app} should ideally be zero. This corresponds to the glide phase of PnG strategy, during which the engine is turned off and energy consumption should theoretically be zero. However, the value of f_{app} computed from (4.66) at $F_t = 0$ is found to be greater than zero. This inaccurately indicates a non-zero power demand, which may result in the engine remaining on unnecessarily. To address this issue, an additional error compensation term is required. This will be discussed in more detail in Section 4.3.2.

4.3.2 Problem Formulation for Eco-HMPC

Consider a typical urban driving scenario for a city bus, as illustrated in Figure 4.16, where the host vehicle encounters various elements of real-world traffic, including surrounding vehicles, bus stops, and signalized intersections during its daily operation.

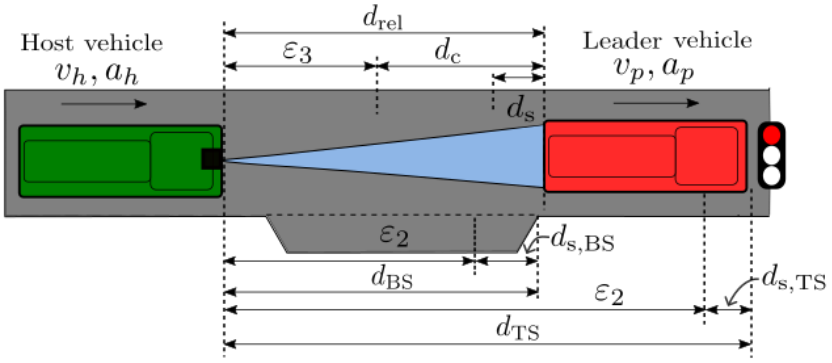


Figure 4.16: Schematic of a typical urban driving scenario encountered by city buses

A new problem formulation (4.67) is proposed that integrates both the reference tracking MPC (4.36) and the car-following MPC (4.62). Additionally, it introduces a binary state variable to capture engine's ON and OFF status. This results in a hybrid optimization problem formulated in the time domain, which is mathematically described as

$$\min_{P_{\text{ICE},k},F_{\text{t},k},F_{\text{b},k},n_k,\varepsilon_{1,k},\varepsilon_{2,k},\varepsilon_{3,k},\varepsilon_{4,k}} \sum_{k=0}^{N-1} P_{\text{ICE},k}\Delta T + \xi_1(v_{\text{h},k} - v_{\text{ref},k})^2 + \xi_2 F_{\text{b},k}^2 + \xi_3\varepsilon_{1,k}^2 + \xi_4\varepsilon_{2,k}^2 + \xi_5\varepsilon_{3,k}^2 + \xi_6\varepsilon_{4,k}^2 \tag{4.67a}$$

s.t. (4.31b), (4.61b), (4.31d), (4.31e), (4.34c), (4.36b), (4.36c), (4.61c), (4.61d)

$$d_{\{\text{TS,BS}\},k+1} = d_{\{\text{TS,BS}\},k} - \Delta T\left(\tfrac{v_{\text{h},k}+v_{\text{h},k+1}}{2}\right) \tag{4.67b}$$

$$n_k - n_{k-1} = \varepsilon_{4,k}; \quad n \in \mathbb{B} := \{0,1\} \tag{4.67c}$$

$$0 \le F_{\text{t},k} \le F_{\text{t,max}} n_k \tag{4.67d}$$

$$f_{\text{app}}(v_{\text{h},k}, F_{\text{t},k}) - P_{\max}(1-n_k) \le P_{\text{ICE},k} \tag{4.67e}$$

$$P_{\text{ICE},k} \le P_{\max} n_k \tag{4.67f}$$

$$d_{\{\text{TS,BS}\},k} \ge 0 \tag{4.67g}$$

$$d_{\{\text{TS,BS}\},k} - \varepsilon_{2,k} \le d_{\text{s},\{\text{TS,BS}\}} \tag{4.67h}$$

The first term in the cost function (4.67a) represents the energy consumption of the host vehicle calculated as the product of the engine power P_{ICE} and the sampling time ΔT, with P_{ICE} constrained by (4.67e) and (4.67f). To address the approximation error in the power consumption model discussed in Section 4.3.1, an error compensation term $P_{\max}(1-n_k)$ is subtracted from the approximated power f_{app} in (4.67e). Here, $P_{\max}$ is

the maximum available power, computed offline from the original engine map, and n_k is a binary variable indicating the engine state:

$$n_k = \begin{cases} 0 & : \text{Engine OFF} \\ 1 & : \text{Engine ON} \end{cases} \tag{4.68}$$

When the engine is OFF ($n_k = 0$), constraints (4.67e) and (4.67f) force P_{ICE} to zero. Substituting the expression for f_{app} from (4.66) into (4.67e) introduces a quadratic inequality to the problem. The second term in the cost function penalizes the deviation of the host vehicle's velocity v_{h} from a reference speed v_{ref}, which is either generated using the GWOS algorithm (3.1) or determined via a dynamic programming approach (4.32) in freeway driving scenarios. To reduce frequent switching between engine ON and OFF states, a slack variable ε_4 is introduced in the last term of the cost function (4.67a). A linear constraint describing ε_4 is given in (4.67b), where n_k is the current and n_{k-1} is the previous engine state. Additionally, the constraint in (4.67h) introduces slack variable ε_2 to encourage the host vehicle to stop as close as possible to the stop line $d_{\text{s},\{\text{TS,BS}\}}$, while the safety constraint in (4.67g) ensures that the vehicle does not violate traffic signals or fail to stop at bus stops. Constraint (4.67d) limits the traction force F_{t} to a maximum value $F_{\text{t,max}}$, which is computed offline for each gear. This force is multiplied by n_k to ensure F_{t} becomes zero when the engine is OFF. Furthermore, the penalty weight ξ_4 is activated only when approaching a red traffic signal, while ξ_1 is deactivated in vehicle-following scenarios and ξ_5 is deactivated in freeway scenarios by setting them to zero. All other terms in the cost function and constraints follow from the earlier problem formulations.

In summary, due to the objective function in (4.67a), and the quadratic inequality constraints, particularly in (4.67e), the problem is classified as a mixed-integer quadratically constrained quadratic programming (MIQCQP) problem. It is solved efficiently using the Gurobi optimizer.

4.3.3 Performance Evaluation of Eco-HMPC

Results for the Vehicle-following Scenario

To evaluate the performance of the proposed Eco-HMPC problem formulation (4.67) in a vehicle-following scenario, a real-world velocity profile of a city bus is substituted as the actual speed profile of the leader vehicle v_{p} (red dotted line), as illustrated in Figure 4.17. In this simulation study, the host vehicle aims to follow the leader vehicle velocity profile, without any influence due to bus stops or TL signals. The results demonstrate that the host vehicle, controlled by both PMPC and FTMPC strategies, maintains smoother velocity trajectories $v_{\text{h,PMPC}}$ and $v_{\text{h,FTMPC}}$ compared to the lead vehicle.Throughout the scenario, the host vehicle complies with the road speed limit v_{max} and successfully maintains an inter-vehicle distance d_{rel} within the desired following

region d_c, without violating the minimum safety distance d_s. The Eco-HMPC controller applied PnG strategy at various instances, effectively switching the engine OFF and reducing the traction force to zero. Minor fluctuations in engine ON/OFF transitions, such as around 1550 s are observed, suggesting that penalizing frequent state changes over multiple timesteps could improve stability–this is identified as a potential direction for future work.

To assess energy efficiency, a longitudinal vehicle dynamics model of the city bus was implemented using the QSS toolbox [GA99]. Simulation results reveal that the Eco-HMPC approach achieves notable reductions in fuel consumption of up to 12.11% with PMPC and 7.11% with FTMPC as compared to the leader vehicle, for a prediction horizon of $N = 8$ steps. These savings stem from the controller's ability to leverage preview information, such as route elevation and the leader vehicle's future velocities, to compute optimal control inputs. The PnG strategy, which enables engine OFF while gliding, further contributes to the observed improvements in fuel efficiency. The higher fuel savings achieved with PMPC relative to FTMPC are attributed to its access to perfect knowledge of the leader vehicle's future velocities. As shown in the zoomed-in view of Figure 4.17, the FTMPC-controlled host vehicle exhibits larger fluctuations in velocity and correspondingly higher peaks in traction force, indicating less optimal control compared to PMPC.

Baseline Controller for a Typical Urban Scenario

To further assess the effectiveness of the proposed Eco-HMPC in typical urban driving scenarios (see Figure 4.16), it is benchmarked against a newly formulated baseline controller. The mathematical formulation of the baseline controller is described using

$$\min_{F_{t,k},F_{b,k},\varepsilon_{1,k},\varepsilon_{2,k},\varepsilon_{3,k}} \sum_{k=0}^{N-1} f_{app}(v_{h,k}, F_{t,k}) + \xi_1 F_{b,k}^2 + \xi_2 \varepsilon_{1,k}^2 + \xi_3 \varepsilon_{2,k}^2 + \xi_4 \varepsilon_{3,k}^2 \tag{4.69a}$$

$$\text{s.t.} \quad (4.31b), (4.61b), (4.67b), (4.31c), (4.31d), (4.31e), (4.34c), (4.67g), (4.67h), (4.61c), (4.61d)$$

Compared to the Eco-HMPC formulation in (4.67), a key distinction in the baseline controller's cost function (4.69a) is the use of an approximated power consumption term derived from (4.66). As a result, the controller does not support the implementation of the PnG strategy. Consequently, the slack variable ε_4, used to facilitate engine switching in Eco-HMPC, is omitted in this formulation. Another significant difference is the baseline controller's lack of capability to track a reference velocity v_{ref} associated with green-wave, instead, it strictly follows the leader vehicle's velocity profile throughout the trip. Therefore, the reference tracking term is excluded from the objective function. Additional constraints associated with traffic light signals and bus stop events are retained, as previously defined in the Eco-HMPC problem formulation.

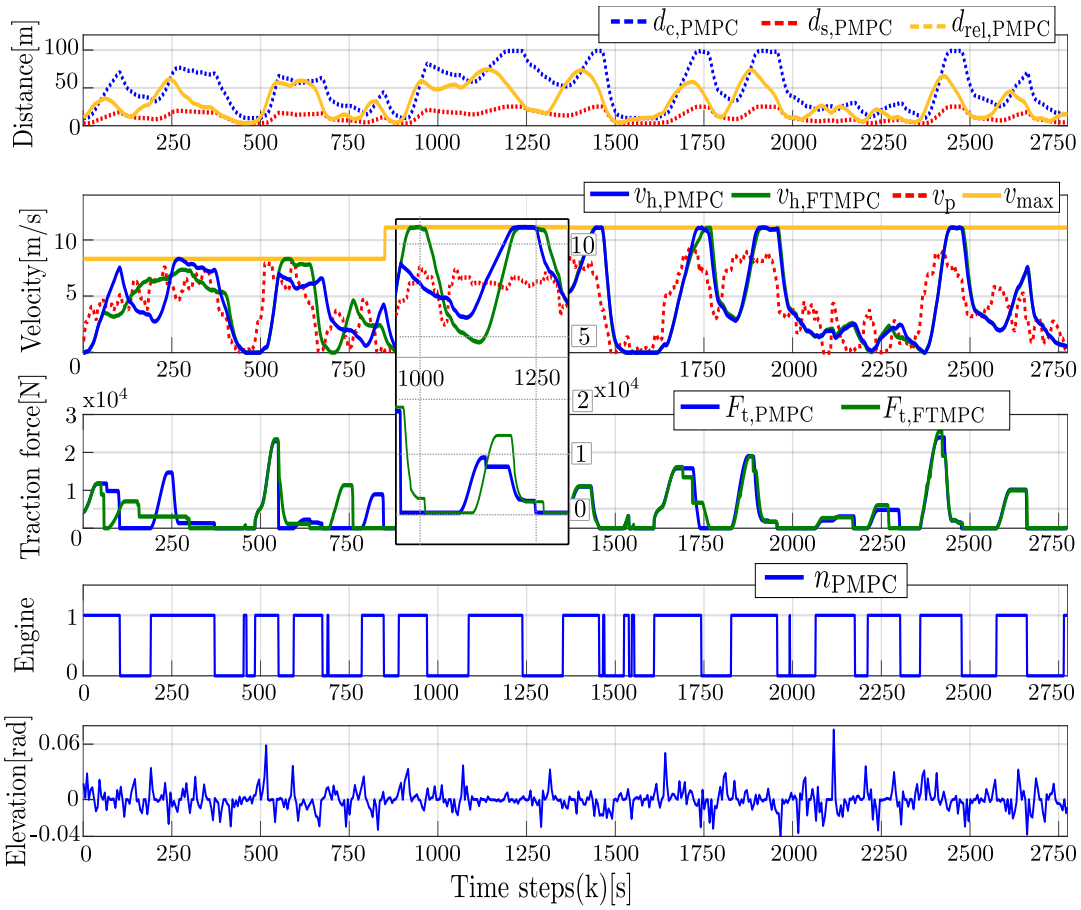


Figure 4.17: Results of Eco-HMPC for a vehicle-following scenario

Results for the Urban Driving Scenario

This section compares the performance of the proposed Eco-HMPC controller (4.67) with a baseline controller (4.69) in a representative urban driving scenario that includes a leading vehicle, signalized intersections, and designated bus stops. It is assumed that both controllers have perfect knowledge of the leader vehicle's future velocity profiles and the locations of upcoming bus stops. The key difference is that, while the host vehicle equipped with Eco-HMPC receives SPaT data for upcoming TL signals, the baseline controller has access only to the current signal phase.

Both controllers were evaluated on a simulated urban route approximately 2.5 km in length, featuring signalized intersections $\mathrm{TS}_{\{1,3,5\}}$ and bus stops $\mathrm{BS}_{\{2,4,6\}}$, as depicted in the space-time diagram in Figure 4.18a. Initially, the Eco-HMPC controller follows the same leader vehicle velocity profile v_p introduced in Figure 4.17. At the first TL signal (TS_1) around 460 s, the host vehicle comes to a halt behind the leader due to a red TL phase. Later, at 1000 s, the host vehicle makes a mandatory stop at bus stop BS_2 for a fixed dwell time st, which increases the inter-vehicle distance d_rel between the

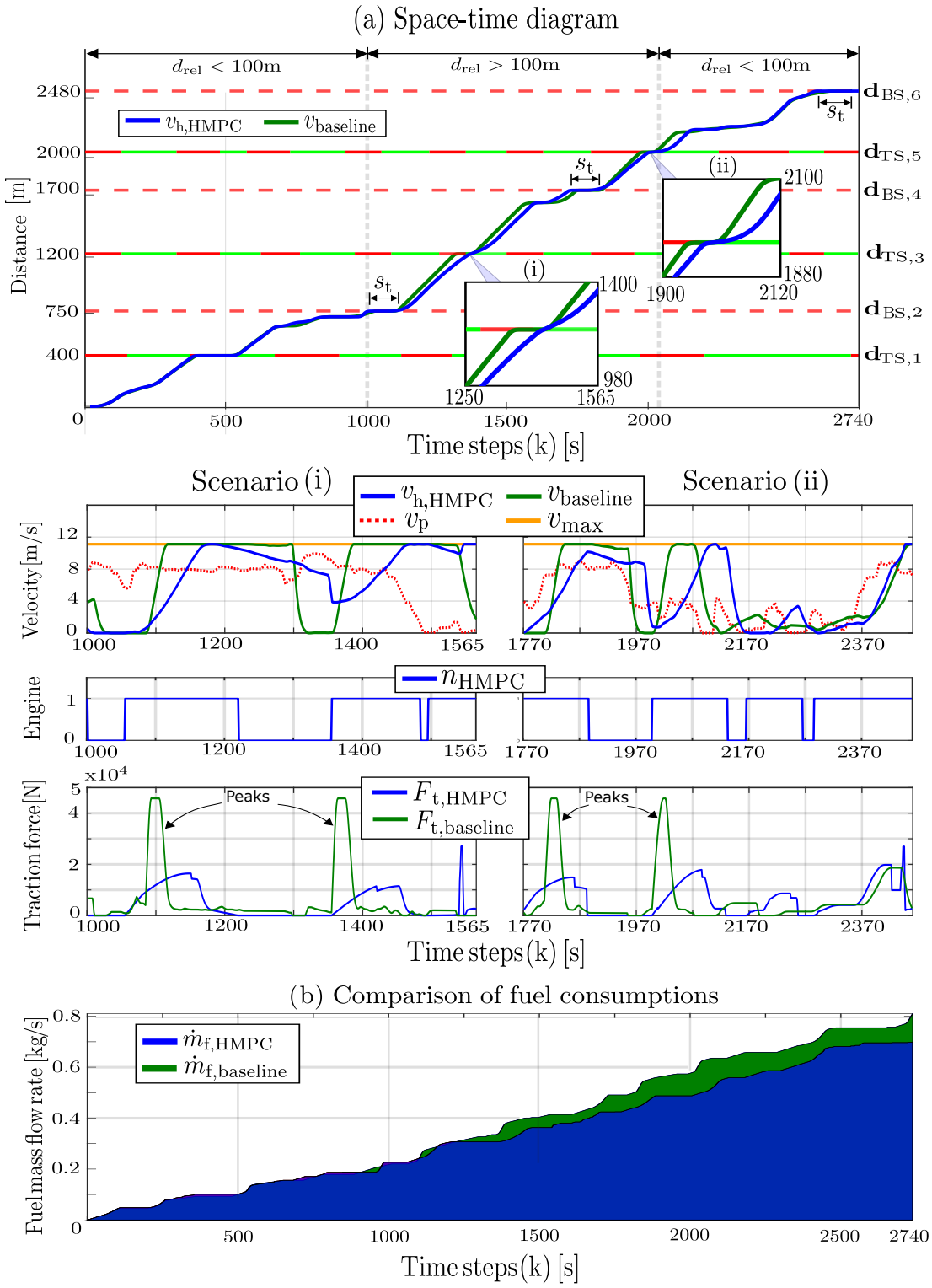


Figure 4.18: Comparison of the simulation results between Eco-HMPC and baseline controllers

host vehicle and leader vehicle. When the relative distance d_{rel} exceeds 100 m (beyond sensor's range) the Eco-HMPC stops tracking the leader vehicle and follows a GWOS to pass subsequent TL signals during green phases.

Enlarged views of two scenarios encountered at signalized intersections TS_3 and TS_5 are depicted in Figures 4.18a(i) and 4.18a(ii), respectively. At 1322 s and 1986 s, the baseline vehicle decelerates and stops at red TL signals due to the lack of SPaT information. In contrast, the host vehicle equipped with Eco-HMPC has the perfect knowledge of SPaT data, adjusts its speed to pass through these signalized intersections during green phases. Additionally, the traction force required by the baseline controller $F_{\text{t,baseline}}$ exhibits significantly higher peaks compared to that of the Eco-HMPC controller $F_{\text{t,HMPC}}$, indicating more abrupt acceleration and deceleration maneuvers. Figure 4.18b shows the fuel mass flow rate over the entire simulation duration for both controllers. It is evident that the baseline controller consumes more fuel in comparison with the Eco-HMPC controller, primarily due to unnecessary stops and aggressive throttle behavior near traffic signals.

A summary of the simulation results is presented in Table 4.1. The Eco-HMPC controller outperforms the baseline, achieving fuel savings of 10.21% and 12% for prediction horizons of $N = 8$ and $N = 15$ steps, respectively. The execution time for the pro-

Prediction Horizon	Controller	Distance [km]	Trip time [s]	Fuel savings
N=8	Baseline	2.48	2739	-
	Eco-HMPC	2.48	2737	+10.21%
N=15	Baseline	2.48	2740	-
	Eco-HMPC	2.48	2735	+12%

Table 4.1: Simulation results for the comparison of the Eco-HMPC with the baseline controller

Prediction horizon	N=8	N=15	N=25
Computation time	23 ms	71 ms	224 ms

Table 4.2: Computation time for different prediction horizons

posed Eco-HMPC strategy for varying prediction horizons is summarized in Table 4.2. The evaluation was conducted using the Matlab® R2020a profiler on a Windows 10 PC equipped with an Intel® Core™ i7-7500U processor operating at 2.70 GHz and 12 GB of RAM. For prediction horizons of $N = 8$, $N = 15$, and $N = 25$, the average computation times were measured to be 23 ms, 71 ms, and 224 ms, respectively. Given a controller sampling time of $\Delta T = 200$ ms, the results indicate that real-time implementation of

the Eco-HMPC is feasible for prediction horizons shorter than $N = 25$ steps (equivalent to 5 s).

4.4 Summary

This chapter proposed problem formulations for eco-driving model predictive controllers (Eco-MPCs) for battery electric vehicles (BEVs) and internal combustion engine (ICE) vehicles, which serve as fundamental components of the predictive EDAS. First, the longitudinal system dynamics and prediction model were developed as the foundation for the mathematical formulation of the MPCs. Then, a BEV powertrain model was introduced, along with an approximation of the BEV power consumption map. The problem formulation for reference-tracking MPC in typical freeway scenarios and at signalized intersections was then discussed. Additionally, the derivation of the MPC optimization problem in quadratic programming (QP) form with linear inequality constraints was presented. For car-following scenarios, problem formulations for the car-following MPC were introduced, considering a leader vehicle within sensor range and the influence of traffic light signals. The simulation-based validation of these controllers demonstrated not only significant energy-saving benefits for BEVs but also an emphasis on driving comfort and safety. Moreover, the seamless switching between controllers was demonstrated. To extend the applicability of predictive EDAS to other powertrains, an Eco-(H)MPC (hybrid model predictive control) approach was proposed for a city bus equipped with a conventional ICE. The Eco-HMPC was formulated as a mixed-integer quadratically constrained quadratic programming (MIQCQP) problem, optimizing both the continuous power from the ICE and the engine ON/OFF state. A comparative evaluation between the proposed Eco-HMPC and a baseline controller revealed energy-saving benefits due to the implementation of a pulse-and-glide (PnG) strategy, in which the vehicle frequently coasts, disabling the ICE and decoupling the powertrain. Finally, an analysis of execution times for the proposed Eco-MPCs across different prediction horizons demonstrated their feasibility for real-time implementation. To enhance the efficiency of the car-following MPC proposed in this chapter, the subsequent chapter will focus on accurately forecasting the leader vehicle's speed using prediction models.

5 Leader Vehicle Speed Prediction

Findings from the Subsections 4.2.4 and 4.3.2 of the previous chapter revealed that the performance of the car-following MPC is significantly influenced by the accuracy of the predicted future velocities of the leader vehicle. Therefore, this chapter focuses on developing an accurate speed prediction model, investigating the relevant environment information to improve the accuracy of the prediction models, and ultimately enhancing the overall efficiency of the car-following MPC.

5.1 Methodology

5.1.1 Data Preparation

In contrary to the other approaches which considered a fixed driving route for data collection [SKSH19, LAG+19, JCG20, WHK+23], this work focuses on network-based data collection in simulation as it is laborious and time-consuming to extract V2V and V2I information in real world. The goal here is to gather rich driving datasets from both urban and highway networks that are route-independent under varied traffic conditions. Moreover, the data must enable developing scalable prediction models and promote generalizability. Therefore, to extract time-series information in this work, the open-source microscopic traffic simulation tool SUMO [LBBw+18] was used. The road networks in SUMO were generated using the open-source map database OpenStreetMap (OSM) [Ope17], and the elevation data was embedded into the maps using shuttle radar topography mission (SRTM) data. To reproduce the real world traffic flow in the SUMO simulation environment, a random traffic is generated with each traffic object in the network having a random starting and destination points. A flowchart in Figure 5.1 provides an overview of the process involved in generating road networks and stochastic traffic in SUMO.

In a highway driving scenario, the speed of the leader vehicle is primarily influenced by the regulatory road speed limits and the velocities of a preceding vehicle. Besides these factors, the presence of traffic light signals in the urban environments influence the leader vehicle speed as well. By leveraging the V2V and V2I information, additional inputs from multiple vehicles ahead of the leader vehicle can be obtained. Moreover, the future signal phase and timing (SPaT) information can be used to improve the predictions for the leader vehicle.

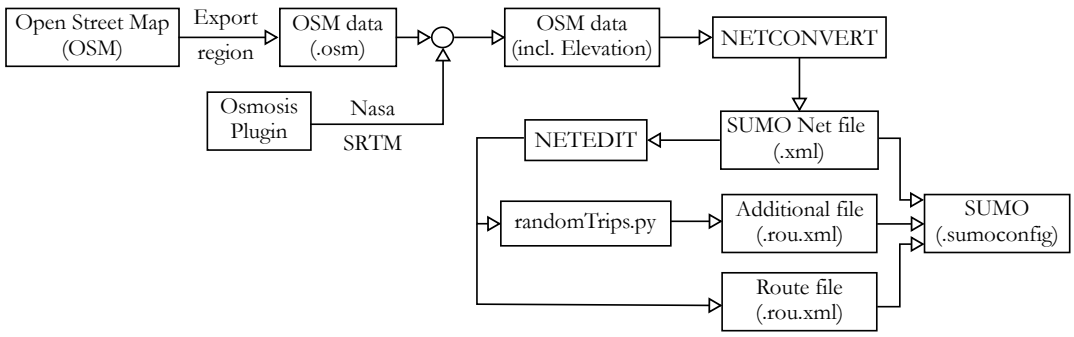


Figure 5.1: Procedure to generate a road network and stochastic traffic in SUMO

In this work, two feature groups FG1 and FG2 are investigated. As listed in Table 5.1 (description given in Table 5.2) and illustrated in Figure 5.2, the feature group FG1 consists of six input features, namely, velocity of the leader vehicle v_{p}, velocity of the first preceding vehicle v_{p_1}, relative distance between the leader vehicle and the first preceding vehicle d_{rel_1}, traffic light signal current state s_{TL}, relative distance between the leader vehicle and the traffic light signal d_{TL}, and the maximum road speed limit $v_{\max}$.

Table 5.1: Feature groups and their corresponding input variables

Feature group	**Input Features**
FG1	$v_{\mathrm{p},k}, v_{\mathrm{p}_1,k}, d_{\mathrm{rel}_1,k}, s_{\mathrm{TL},k}, d_{\mathrm{TL},k}, v_{\max,k}$
FG2	$v_{\mathrm{p},k}, v_{\mathrm{p}_1,k}, d_{\mathrm{rel}_1,k}, s_{\mathrm{TL},k}, d_{\mathrm{TL},k}, v_{\max,k}$ $v_{\mathrm{p}_2,k}, d_{\mathrm{rel}_2,k}, s_{\mathrm{TL},k+1}, \ldots, \ s_{\mathrm{TL},k+H}$

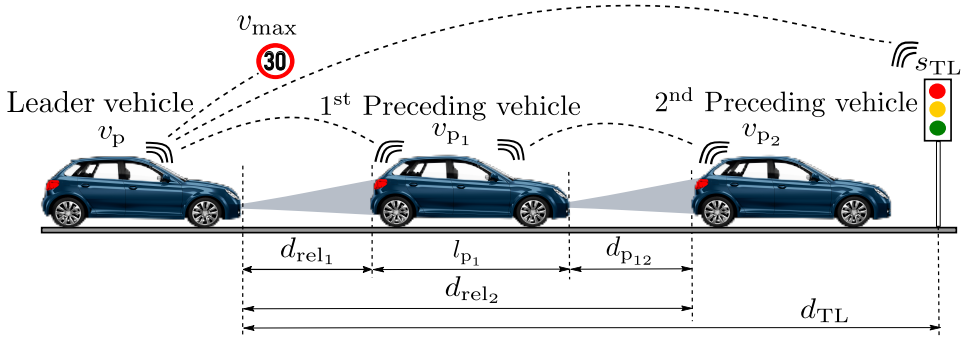


Figure 5.2: Schematic of V2V and V2I enabled traffic scenario

In addition to the input features described in FG1, the feature group FG2 considers the influence of the second preceding vehicle with velocity v_{p_2}, and the relative distance between the leader vehicle and the second preceding vehicle d_{rel_2} that is calculated using

$$d_{\mathrm{rel}_2} = d_{\mathrm{rel}_1} + l_{\mathrm{p}_1} + d_{\mathrm{p}_{12}} \tag{5.1}$$

Table 5.2: Various driving and traffic input data for the prediction models

Input variables	Description
v_{p}	Velocity of leader vehicle speed
v_{p1}	Velocity of first preceding vehicle
v_{p2}	Velocity of second preceding vehicle
$d_{\mathrm{rel_1}}$	Relative distance between the leader and the first preceding vehicle
$d_{\mathrm{rel_2}}$	Relative distance between the leader and the second preceding vehicle
$v_{\max}$	Maximum road speed limit
s_{TL}	Traffic light signal current state
d_{TL}	Relative distance between the leader vehicle and the traffic light signal

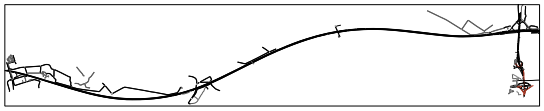
where $l_{\mathrm{p_1}}$ is the length of the first preceding vehicle and $d_{\mathrm{p_{12}}}$ is the relative distance between the first and second preceding vehicles. Besides, FG2 also considers the future traffic light signal state information $s_{\mathrm{TL},k+1}, ..., \ s_{\mathrm{TL},k+H}$ until the prediction horizon H as input features. During the data collection process, each traffic object in the network is considered to be a leader vehicle and the input variables as given in Table 5.1 are gathered. To access the above mentioned variables from the simulation, a traffic control interface known as TraCI4Matlab is used [AEE15] with a sample rate of 1 s. Furthermore, preprocessing techniques such as data cleaning, normalization, and data splitting into training, validation and testing are performed to handle the data in an efficient manner.

In this work, the training datasets for the prediction models were generated using the road networks of the Landstuhl highway (Figure 5.3a) and Kaiserslautern city (Figure 5.3b). To enhance generalizability, test datasets were created using a different city and its connecting highway (Nieder-Olm), as illustrated in Figure 5.3c.

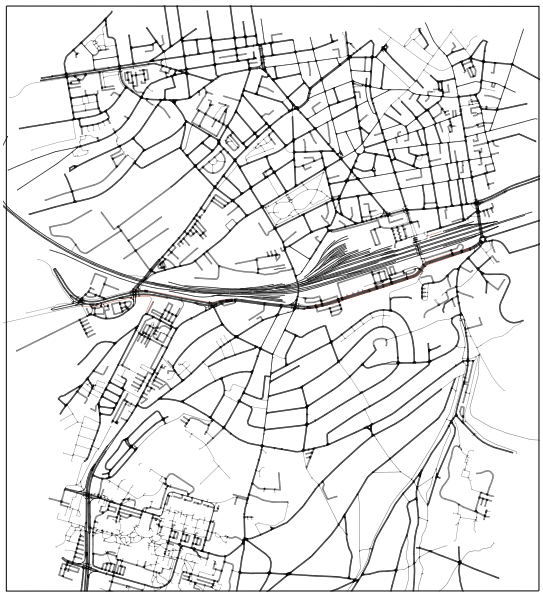
5.1.2 Prediction Models

In this work, both physics-based prediction methods [LVL14] and deep recurrent neural networks (RNNs) are explored to develop a leader vehicle speed predictor. Among the physics-based approaches, two commonly used models are studied: the Constant Velocity (CV) model, which assumes that the leader vehicle maintains a constant speed in the future, and the Constant Acceleration (CA) model, which assumes that the vehicle's future speed increases at a constant acceleration. In parallel, two RNN architectures, namely, Gated Recurrent Units (GRU) and Long Short-Term Memory (LSTM) networks are investigated.

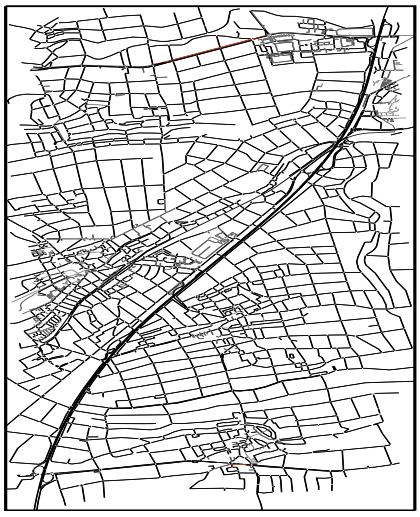
The internal cell structure of LSTM and GRU is discussed in Chapter 2 (Section 2.2.4). To process the sequential data effeciently and obtain a greater level of abstraction, the hidden layers in RNNs are arranged together in stacked fashion as illustrated in Figure 5.4. The architecture of a stacked RNN consists of three segments, described from bottom to top: (i) an input layer, (ii) stacked LSTM or GRU layers, (iii) an output layer. The input layer consists of multiple input features (marked in green) as described



(a) Landstuhl highway



(b) Kaiserslautern city



(c) Nieder-Olm

Figure 5.3: SUMO networks to generate training (a & b) and testing (c) datasets

in Table 5.1. In Figure 5.4 from left to right, the feature values of past, current and future timesteps are provided as inputs to the stacked layers. On the right, along with the current traffic light state, future state information up to a prediction horizon H is additionally provided. The combined input sequence is fed into the stacked hidden layers. Each hidden layer has memory cells associated with it, otherwise known as neurons. The lower layers are used to capture the low level representations and the higher layers can learn deeper levels of abstraction. The third segment (in yellow) outputs multi-step sequence predictions up to a prediction horizon H, in this case the future velocities of the leader vehicle.

To model the RNNs in this work, the keras deep learning library was used [Bro18] in python. The training of the deep learning prediction models was performed on a GPU cluster. The hyperparameters for both the LSTM and GRU have been tuned using the Keras Bayesian optimization function [OBL+19]. The tuned hyperparameters for the feature groups (FG1 and FG2) for a prediction horizon of 5 s are illustrated in the Table 5.3. To model the non-linear input data representations to outputs, four stacked layers were implemented. Moreover, to reduce overfitting of the data during the training process, a regularization technique called dropout was used. Furthermore, a rectified linear activation function was used to train the network faster, and a mean squared error was used as the loss function to estimate the loss of the model while training.

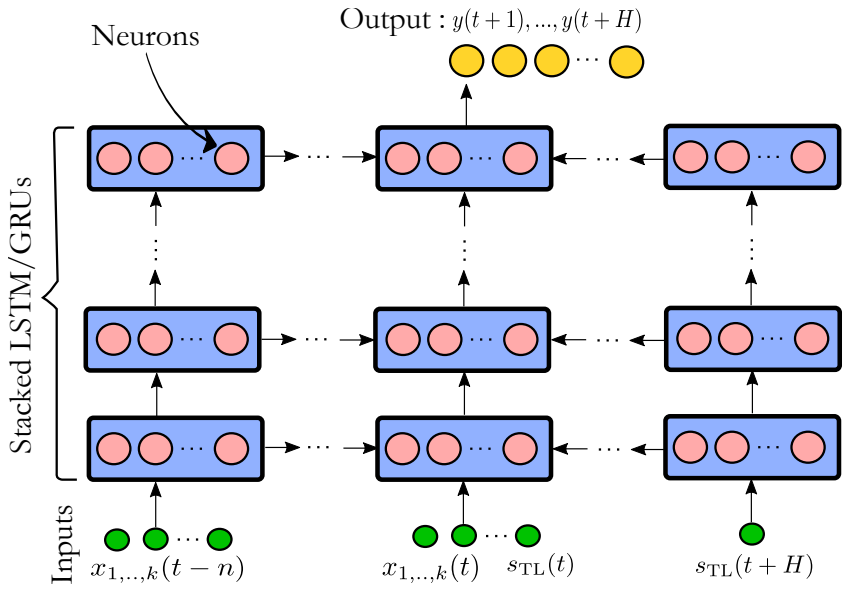


Figure 5.4: Architecture of stacked RNN for leader vehicle speed prediction

Hyperparameters	FG1		FG2	
	LSTM	GRU	LSTM	GRU
Batch size	32		512	
Input features	6		13	
Stacked layer 1	90	450	600	300
Dropout 1	0	0.3	0	0.2
Stacked layer 2	60	600	420	600
Dropout 2	0	0.3	0.25	0
Stacked layer 3	600	60	450	600
Dropout 3	0.3	0	0.3	0
Stacked layer 4	600	60	480	180
Dropout 4	0	0.3	0.3	0.1
Dense layer	30	570	60	30

Table 5.3: Hyperparameters for LSTM and GRU models with feature groups (FG1 and FG2) for a prediction horizon of 5 s

Additionally, the ADAM optimizer was used to efficiently update the network weights while training. For the simulations, the length of the past sequence was maintained to be equivalent to the prediction horizon H, and the learning rate was chosen as $1e^{-}3$. Moreover, the number of epochs was chosen as 25, in addition to which an early stopping was implemented to be able to stop the training process when a desired metric has stopped to improve.

5.2 Results and Discussion

5.2.1 Performance Evaluation of the Prediction Models

The performance of the prediction models discussed in Section 5.1.2 is evaluated on 5 test datasets from the road network depicted in Figure 5.3c. The average MAE and RMSE for each prediction model across different prediction horizons (5 s and 10 s) with respect to the feature groups (FG1 and FG2) are summarized in Table 5.4. It can be noticed that the prediction error increases for all the models as the prediction horizon increases due to the increasing uncertainties. Moreover, the LSTM model with input features FG2 has outperformed remaining methods, and showcased lower average prediction error as compared to feature group FG1.

Prediction Models	**MAE [m/s]**				**RMSE [m/s]**			
	5s		**10s**		**5s**		**10s**	
	FG1	FG2	FG1	FG2	FG1	FG2	FG1	FG2
CV	2.12		3.16		3.79		5.43	
CA	2.75		5.26		4.77		8.68	
LSTM	2.06	**1.75**	4.26	**2.92**	3.19	**2.84**	5.63	**4.61**
GRU	2.02	2.51	3.19	3.3	3.29	3.68	4.58	4.7

Table 5.4: Comparison of MAE and RMSE for various prediction models

The prediction results for the LSTM-FG2 model with a prediction horizon of 5 s after evaluating it on one of the test datasets that consists of various traffic scenarios are shown in Figure 5.5. Furthermore, the prediction results of $\hat{v}_{\text{LSTM-FG1}}$ are compared against $\hat{v}_{\text{LSTM-FG2}}$ in Figure 5.6 under two scenarios. In Figure 5.6a, the leader vehicle is under the influence of two preceding vehicles ahead (exemplary scenario in Figure 5.2). As presented in Table 5.1, the FG1 uses the inputs from the first preceding vehicle alone, whereas the FG2 uses both the preceding vehicle information as inputs with an assumption that this information can be obtained from V2V communication. The results illustrate that the $\hat{v}_{\text{LSTM-FG2}}$ has better tracking ability of the leader vehicle v_{t} as compared to $\hat{v}_{\text{LSTM-FG1}}$. In the second scenario as illustrated in Figure 5.6b, the leader vehicle has stopped at a red phase of a traffic light signal. The predictions of $\hat{v}_{\text{LSTM-FG2}}$ match the expected behavior of the leader vehicle v_{t}, however, as the feature group FG1 did not take the future traffic light phase into account, the $\hat{v}_{\text{LSTM-FG1}}$ depict inaccurate predictions that the leader vehicle will move forward from standstill. Such a behavior may in turn lead to traffic signal violations which is not desirable.

Furthermore, a comparison of the prediction results for the methods discussed in Table 5.4 when evaluated under various traffic scenarios (maneuvering a right turn, freeway scenario, approaching a stopped preceding vehicle at a traffic light signal, navigating through a roundabout, and approaching a red traffic light signal) can be found in Figure 5.7. It can be observed that the LSTM-FG2 prediction model has demonstrated

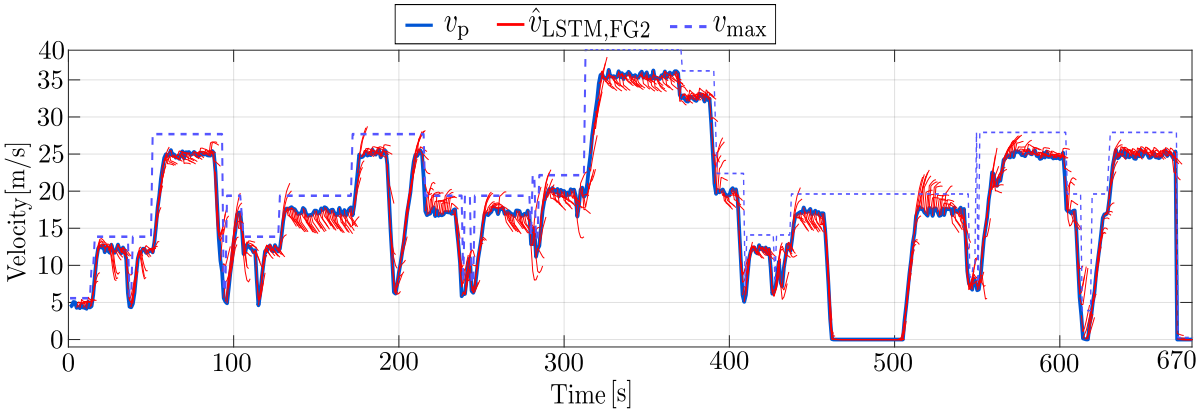


Figure 5.5: Prediction results of LSTM-FG2 on one of the test datasets for a prediction horizon of 5 s

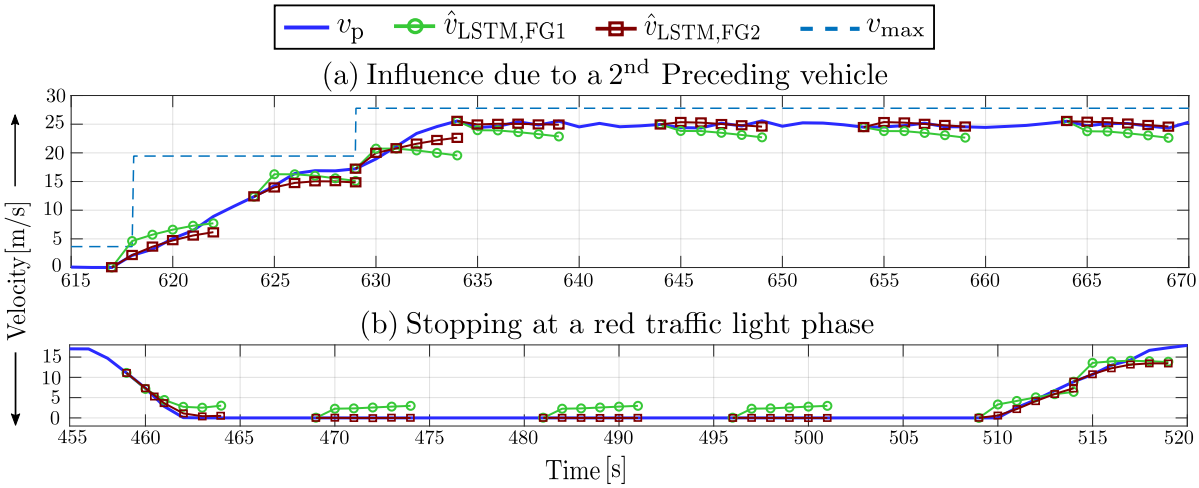


Figure 5.6: Comparison of leader vehicle speed predictions for LSTM-FG1 and FG2 models for a prediction horizon of 5 s

better prediction accuracy in all the scenarios as compared to GRU-FG2, CV and CA models. Although the prediction results of $\hat{v}_{GRU,FG2}$ matches with the predictions of $\hat{v}_{LSTM,FG2}$ at a few timestamps, its accuracy needs to be improved in a few scenarios (e.g. round-abouts). The predictions based on CV and CA models were not able to accurately predict the leader vehicle behavior due to the presence of abrupt speed variations in the leader vehicle. It can be noticed from Figure 5.7, that a prediction error exists with the $\hat{v}_{LSTM,FG2}$ as well. For instance, while maneuvering a right turn (Figure 5.7a), at 89 s the model did not predict the slow down at the curvature accurately. Similarly, in Figure 5.7d the model is able to predict the future speeds accurately in the round-about only until 3 s into the future.

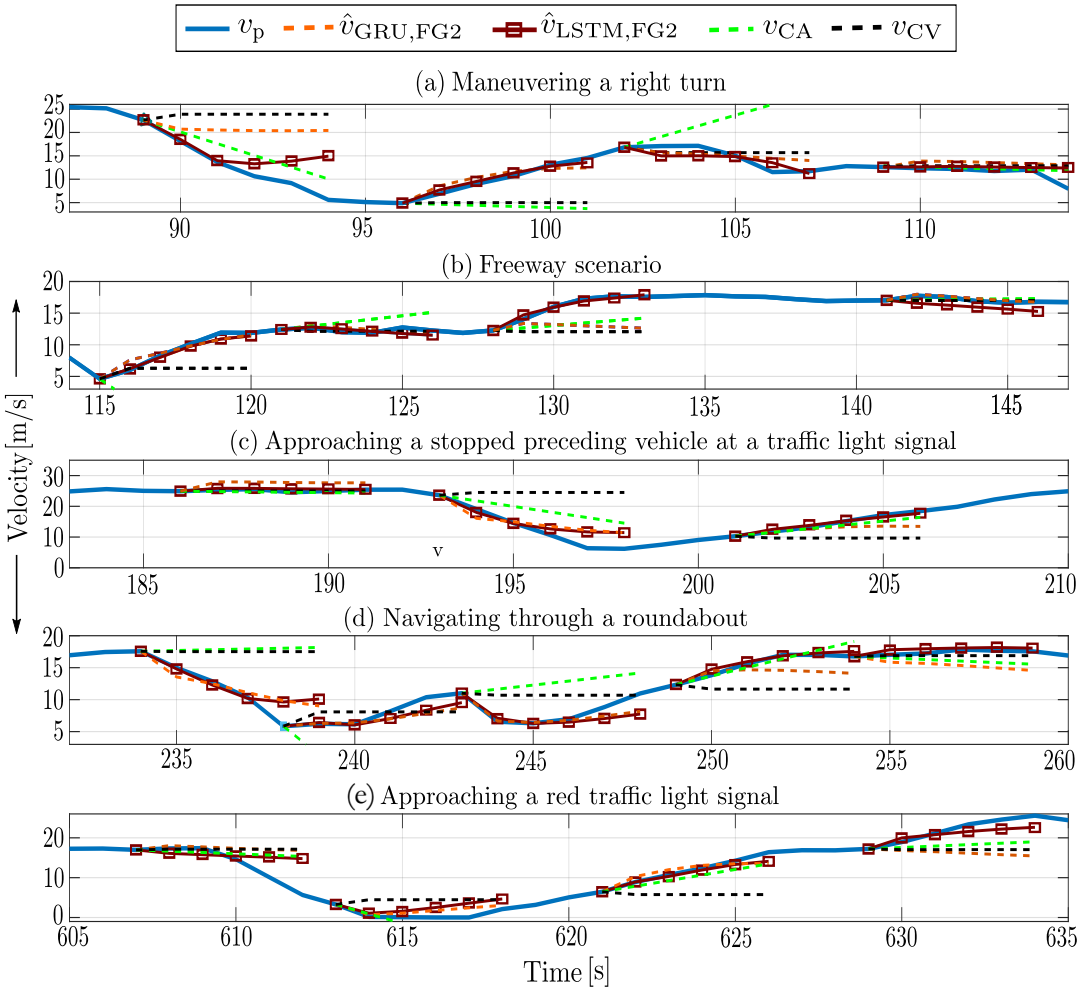


Figure 5.7: Speed forecasting results of $\hat{v}_{GRU,FG2}$, $\hat{v}_{LSTM,FG2}$, $\hat{v}_{CA}$ and $\hat{v}_{CV}$ for a prediction horizon of 5 s

5.2.2 Performance Evaluation of the Car-following MPC using Predicted Leader Vehicle Speeds

To analyze the performance of the host vehicle equipped with the car-following MPC while tracking a leader vehicle, an evaluation was conducted based on three setups in which the leader vehicle future velocities are (I) assumed to be constant due to the lack of a robust speed predictor and V2V communications, (II) predicted using the LSTM-based speed prediction model proposed in this work, and (III) assumed to be perfectly available. The speed profiles of the host vehicle for the three setups $v_{h,I}$, $v_{h,II}$ and $v_{h,III}$ are illustrated in Figure 5.8 while tracking a leader vehicle velocity profile v_t. In the distance plot in the above part of Figure 5.8, the host vehicle with condition II is seen performing a robust car-following and maintaining a good inter-vehicle distance to the leader vehicle $d_{rel,II}$. It can be noticed from the two enlarged sections in the below part of Figure 5.8,

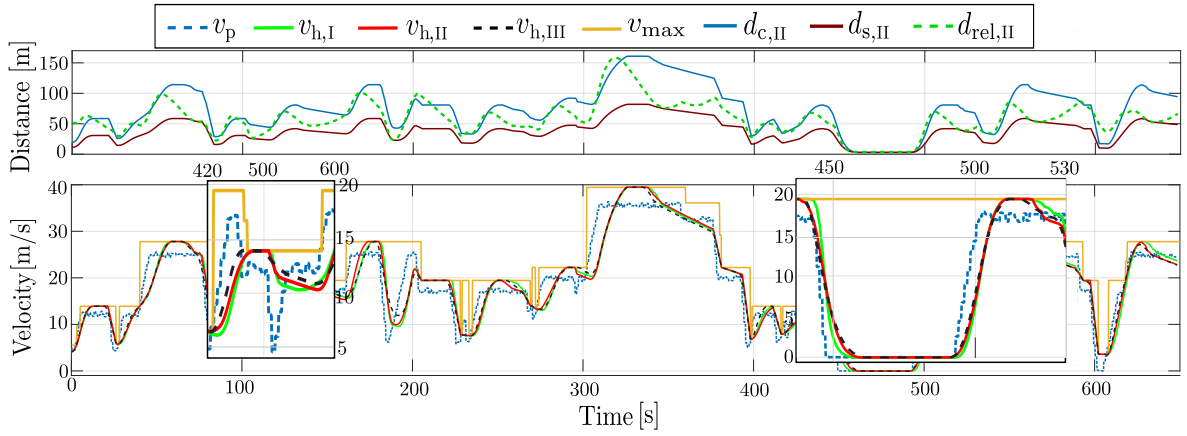


Figure 5.8: Speed profiles of the host vehicle equipped with car-following MPC while tracking leader vehicle velocity profiles under the setups I, II and III

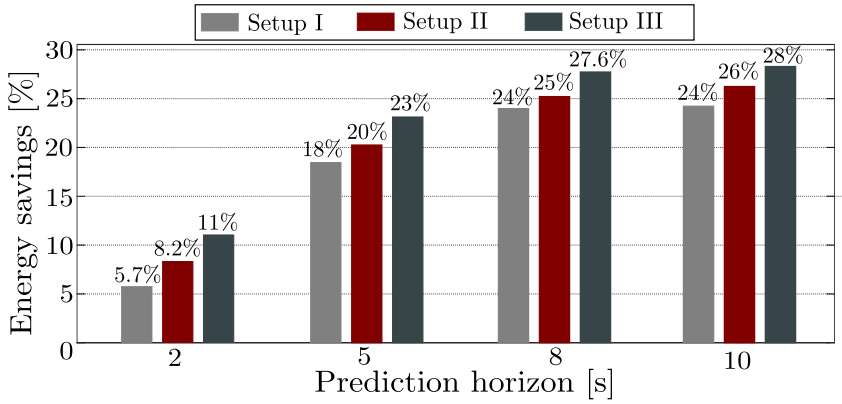


Figure 5.9: Energy savings for host vehicle equipped with car-following MPC while tracking the leader vehicle velocity profiles under the setups I, II and III at different prediction horizon lengths

that the $v_{h,II}$ is able to track the $v_{h,III}$ better than the $v_{h,I}$. Moreover, considering a constant velocity (CV) model in setup I has resulted in abrupt decelerations and sharp accelerations as compared to the II and III.

In Figure 5.9, a comparison of the host vehicle energy savings at different prediction horizon length while tracking the leader vehicle speed profiles obtained using the aforementioned setups. The results demonstrate that the energy savings can increase with the increase in the prediction horizon, i.e. the more the information about the leader vehicle is available into the future, the better that the host vehicle is able to plan its optimal trajectories. Moreover, the proposed LSTM-based car-following MPC (setup II) is able to achieve better energy savings as compared to the setup I and is close to setup III. The energy savings up to 26% can be realizable for the host vehicle equipped with the proposed LSTM-based car-following MPC while tracking the leader vehicle predicted velocities. A performance increment in terms of additional average energy savings of up

to 2.5% with the proposed LSTM-based car-following MPC can be achieved as compared to the constant velocity (CV) model.

5.3 Summary

To improve the efficiency of the car-following MPC proposed in Chapter 4, accurately predicting the leader vehicle speed is focused in this chapter. In this chapter, a scalable and more generalizable network-based time series data collection in both urban and highway environments using the microscopic traffic simulation tool SUMO is proposed. The input variables are grouped into two feature groups. The leader vehicle speed prediction using deep recurrent neural networks (GRU and LSTM) is performed in this work, and compared with the physics-based models (CV and CA). It can be observed that the LSTM-based prediction model outperformed remaining models (GRU, CV and CA) in terms of better prediction accuracy while forecasting the leader vehicle future velocities. Moreover, leveraging the critical environmental information (surrounding information) via V2V and V2I communication (e.g. multiple vehicles ahead of the leader vehicle, future SPaT information) has improved the prediction accuracy. Furthermore, the performance of the car-following MPC on the predicted leader vehicle velocities is evaluated, and the energy-saving potential for different prediction horizons are investigated. The results demonstrate that the energy savings can increase with the increase in the prediction horizon, i.e. the more the information about the leader vehicle is available into the future, the better that the host vehicle is able to plan its optimal trajectories. A performance increment in terms of additional average energy savings of up to 2.5% with the proposed LSTM-based car-following MPC can be achieved as compared to the constant velocity (CV) model. In the next chapter, results of the objective evaluation of predictive EDAS will be discussed.

6 Objective Evaluation of Predictive EDAS

The performance of eco-driving model predictive controllers (Eco-MPCs) has been extensively analyzed through simulations in Chapter 4. For battery electric vehicles (BEVs), problem formulations (4.36) and (4.62) have been identified as the most effective for reference tracking MPC and car-following MPC, respectively. In this chapter, the effectiveness of the predictive EDAS in promoting energy-efficient driving behavior is evaluated through driver-in-the-loop tests in complex driving scenarios. Additionally, the results of the objective evaluation are presented using key performance indicators in Section 6.1. Furthermore, the impact of driver behavior modeling on speed-tracking accuracy and its role in enhancing the efficiency of predictive EDAS is analyzed in Section 6.2.

6.1 Evaluation I: Performance Evaluation under Driver-in-the-Loop Tests

6.1.1 Simulation Test Conditions

To investigate potential improvements in drivers' driving styles and reductions in energy consumption in BEVs with the assistance of the predictive EDAS, a virtual driving environment was designed for driver-in-the-loop validations in this dissertation. To model and configure road networks, which are fundamental components of the static environment in a virtual driving simulation, including features such as lanes, junctions, elevation profiles, and road markings–IPG CarMaker's scenario editor was utilized. Additionally, to enhance realism in virtual environments, static objects such as buildings, trees, traffic barriers, road signs, and traffic light (TL) signals, which represent non-moving elements, were incorporated. Furthermore, to introduce dynamic objects (such as surrounding traffic) into the virtual environment, a co-simulation framework (Section 3.1.4) was developed in this dissertation to facilitate information exchange between IPG CarMaker and SUMO. To ensure consistency in the static environment across both tools, the road network created using the IPG Scenario Editor was exported to SUMO via the OpenDRIVE format (version 1.4). Since real-world traffic data was unavailable in this work, an alternative method for generating traffic demand using flow definitions was explored. The advantage of defining traffic via a probabilistic flow parameter is that traffic den-

sity can be controlled between a starting edge and an endpoint, allowing variations in different parts of the road network. For the road network used in this study, traffic flow was set to moderate in urban and motorway areas and low in suburban towns.

To validate the predictive EDAS in driver-in-the-loop tests, a 13.2 km driving route within the road network was selected for the host vehicle. The driving route consists of three distinct segments: (i) an urban area, serving as the starting point for the host vehicle, with variable speed limits of 30 km/h, 40 km/h and 50 km/h, as well as two stop signs and five signalized intersections, (ii) a highway (motorway) segment extending approximately 6.8 km, with a maximum speed limit of 120 kmph, and (iii) a sub-urban town, where the speed limit is restricted to 30 km/h, featuring two traffic lights and one stop sign. Additionally, to maintain smooth traffic flow and prevent excessively low driving speeds that could inconvenience following vehicles, a minimum road speed limit was introduced.

6.1.2 Experimental Procedure

A total of 44 participants (Male=37, Female=7) possessing a valid EU driving license and normal or corrected-to-normal visual acuity volunteered to test the predictive EDAS. However, three participants experienced motion sickness and were unable to complete the study. Additionally, seven participants encountered technical issues with the driving simulator during the test run, hence, their objective driving data could not be collected. As a result, the sample size for the objective analysis was reduced to 34 participants. Before the experimental study, participants completed a brief socio-demographic survey in which the information regarding their age, gender, driving experience, and driving style they possess was gathered. The responses indicated that 20.54% identified as cautious drivers, 58.82% as normal drivers, and 20.54% as sporty drivers. Furthermore, participants were surveyed on their familiarity with eco-driving concepts. The feedback revealed that 2.5% of the participants have never heard of eco-driving, 42.5% have heard about eco-driving but had never used it, 32.5% were moderately familiar with similar eco-driving systems and had used similar systems in few instances, 15% were quite familiar with similar eco-driving systems and had used such systems occasionally, and 7.5% regularly use or had used a similar system in their driving routine.

The timeline of the experimental study is illustrated in Figure 6.1. Upon arrival, participants were briefed about the study and its objectives. It is important to mention here that the predictive EDAS and its features were not introduced to the participants at this stage. In the first phase, participants were presented with a short practice route and multiple trial drives to familiarize themselves with the driving simulator and its controls. In the next phase, participants drove along a predetermined 13.2 km route consisting of urban and highway road segments with realistic traffic, deterministic TL signals, and stop signs. Importantly, during this phase, participants drove with their natural driving style without any assistance from the predictive EDAS. To ensure a fair

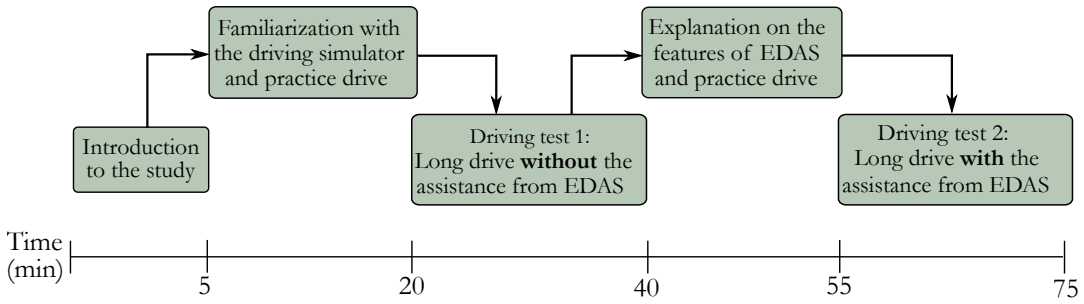


Figure 6.1: Experimental procedure

evaluation, participants were instructed not to overtake other vehicles and to maintain a minimum speed of 80 km/h on the highway segment. Following this drive, participants were introduced to the predictive EDAS and its features. It is then followed by practice test drives to familiarize themselves with the system's suggestions. Finally, in the last phase, participants drove the same predetermined route, this time with assistance from the predictive EDAS.

6.1.3 Driving Behavior Analysis

To analyze driving behavior before and after using the predictive EDAS, the driving data, including velocity, acceleration, and energy consumption profiles of a participant, were compared, as illustrated in Figure 6.2. The results indicate that the predictive EDAS assisted the driver in the following ways: (A) Maintaining compliance with speed limits by reducing instances of exceeding $v_{\max}$, (B) Crossing three signalized intersections TL_1, TL_5 and TL_7 during the green phase by tracking a green-wave optimal speed, (C) Reducing speed at curved roads to ensure safer driving, (D) Lowering overall average driving speeds, (E) Minimizing unnecessary accelerations and decelerations, and (F) Reducing energy consumption in the BEV compared to driving without predictive EDAS assistance. As a result, the participant achieved an overall energy savings up to 15%, with 7% savings in urban areas and 18.5% on highways. Additionally, as a direct consequence of smoother acceleration and deceleration patterns while following the predictive EDAS recommendations, the overall jerk was reduced by 12%. These findings demonstrate that the participant benefited significantly from adhering to the predictive EDAS suggestions, leading to an improved, more energy-efficient driving style.

To demonstrate the switching performance between reference tracking and car-following MPCs, a scenario encountered by the same participant while driving with the predictive EDAS is illustrated in Figure 6.3. The left illustration depicts a situation where, at time $t = t_1$ and distance $s = 1000\,\mathrm{m}$, the host vehicle, equipped with predictive EDAS, is driving on a freeway and following a green-wave optimal speed using the reference tracking MPC to cross the TL signal during the green phase.

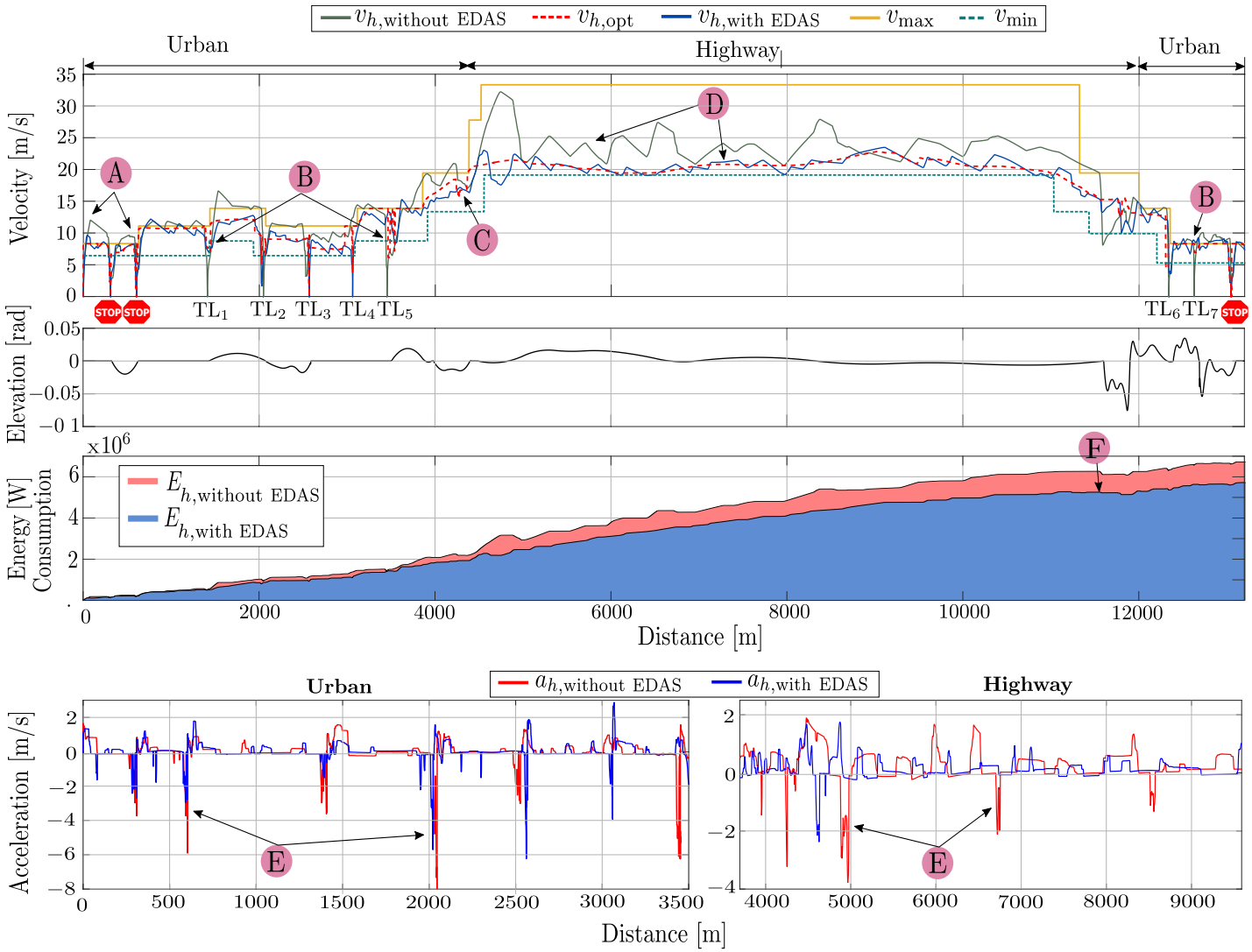


Figure 6.2: Driving behavior comparison of one of the participants before and after using predictive EDAS through velocity, energy consumption and acceleration/deceleration profiles

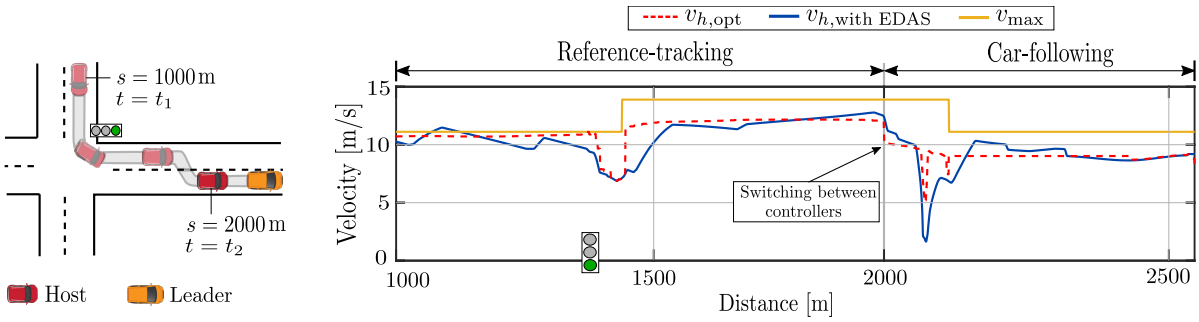
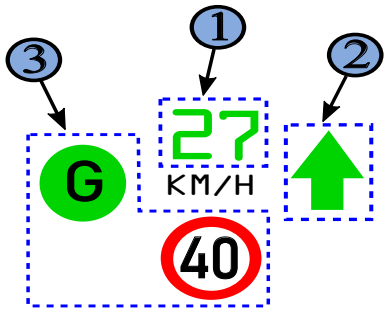


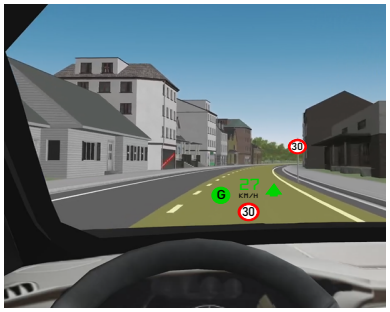
Figure 6.3: Scenario to demonstrate the switching between reference tracking and car-following MPCs

As shown in the right illustration in Figure 6.3, after passing the TL signal at $s = 1400\,\mathrm{m}$, the optimal host speed $v_{\mathrm{h,opt}}$ recommends a speed reduction to safely navigate a turn. Subsequently, at time $t = t_2$ and distance $s = 2000\,\mathrm{m}$, the host vehicle merges into an adjacent lane, triggering a switch to the car-following MPC due to the presence of a leader vehicle ahead. The $v_{\mathrm{h,opt}}$ then recommends a new optimal speed to maintain safe and efficient car-following behavior. The transition between the reference tracking and car-following MPCs is observed to be smooth and stable, demonstrating the effectiveness of the controller-switching mechanism in predictive EDAS.

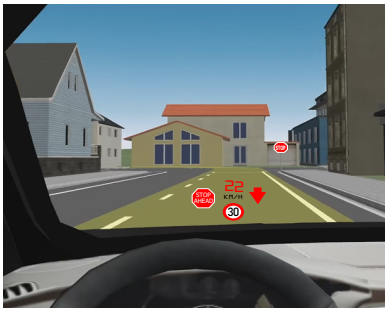
The suggestions obtained from the predictive EDAS in various driving scenarios during driver-in-the-loop tests are depicted in Figures 6.4a-6.4i. The speed advisory system, as illustrated in Figure 6.4a, presents information at three regions on the heads-up display (HUD). At the first region, the current speed of the host vehicle is displayed. The second region depicts an arrow, indicating the driver to either speed up (green, pointing upward) if the current speed is below the optimal speed or decelerate (red, pointing downward) if it exceeds the optimal speed. The length of the arrow represents the magnitude of speed adjustment required. In the third region, additional contextual information, such as the remaining time of a TL signal phase, green-wave advisories, and detected traffic signs is provided to aid the driver to better understand the recommendations provided by predictive EDAS. Additionally, the allowable road speed limit is displayed beneath the host vehicle's current speed. Figure 6.4b illustrates a green-wave speed advisory, where the system suggests an optimal speed to help the driver cross an approaching TL signal during the green phase. A visual icon with the letter G on a green background appears on the HUD, signaling the possibility of maintaining a green-wave. Predictive EDAS also alerts the driver about an upcoming stop or yield sign by displaying a corresponding visual icon on the HUD, as shown in Figure 6.4c. Simultaneously, it advises the driver to gradually reduce speed while approaching these signs, which helps minimize sudden deceleration and associated excessive jerks. Furthermore, in a freeway scenario, predictive EDAS recommends the driver to track a dynamic programming (DP)-optimized speed. Since the optimal speed may vary at each time step, maintaining an exact target speed could be challenging. Therefore, after repeated trials, an optimal speed band of $2\,\mathrm{km/h}$ was defined in this study.



(a) 1. Current speed of the host vehicle, 2. Speed advisory using arrow visualization, 3. Additional useful information

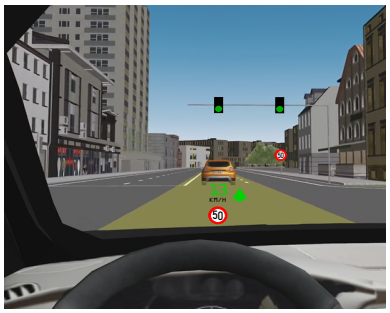


(b) Green-wave speed advisory to cross the immediate TL signal at green phase

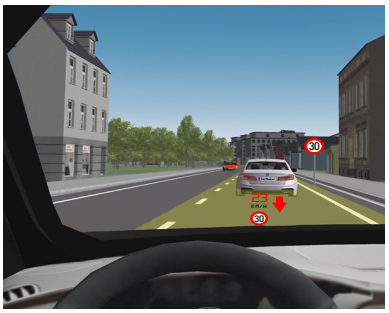


(c) Speed advisory to decelerate while approaching a stop sign

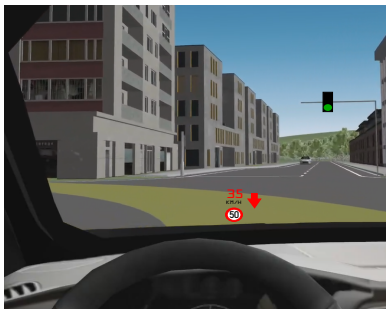
(d) Black digits indicate that current velocity is within the optimal speed band



(e) Indication to speed up to maintain a desired inter-vehicle distance to a leader vehicle



(f) Advisory to decelerate to maintain a safe distance to a leader vehicle



(g) Speed advisory to decelerate and travel at a safe speed while approaching a curved road

(h) Indication of the remaining time of green TL signal phase

(i) Indication of the remaining time of red TL signal phase

Figure 6.4: Recommendations from predictive EDAS at different driving scenarios in the test route

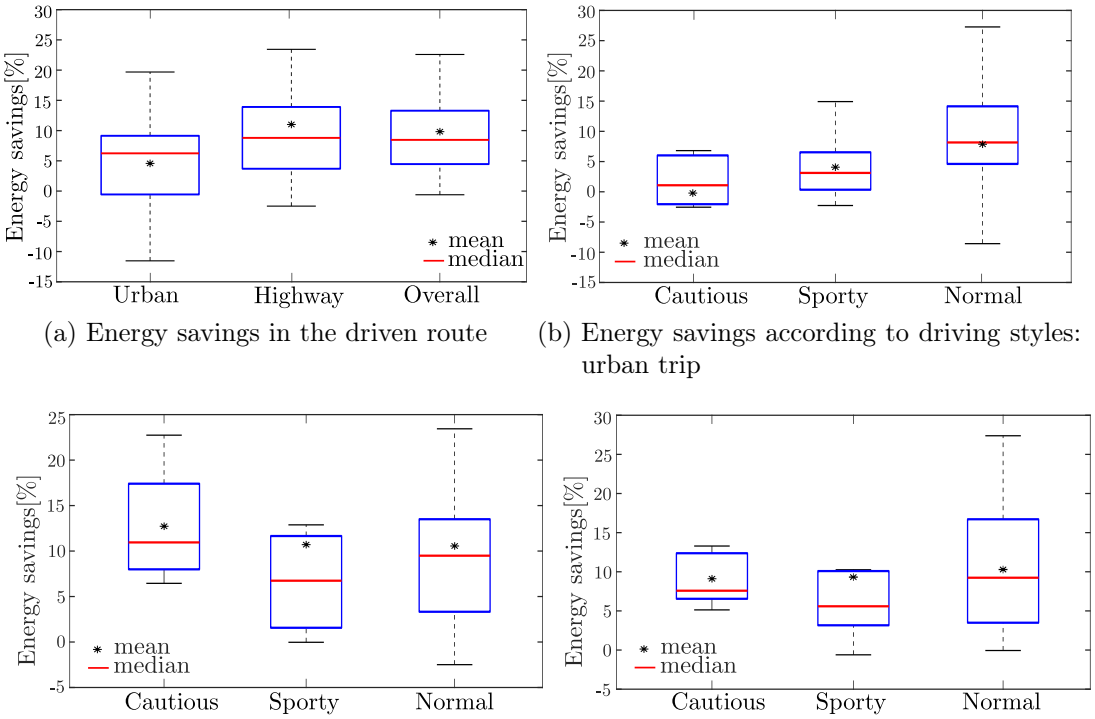
When the driver maintains speed within this band, the speed digit on the HUD remains black, as illustrated in Figure 6.4d. In such scenarios, the driver is advised to adopt a coasting strategy by avoiding throttle or brake input. Moreover, in car-following scenarios, predictive EDAS suggests maintaining a safe inter-vehicle distance from the leading vehicle. If the host vehicle falls below the comfort distance, the system advises the driver to accelerate (Figure 6.4e). Conversely, if the safe following distance is at risk of being violated, the system recommends slowing down (Figure 6.4f). In addition, the driver is alerted with a warning tone if the suggestion to slow down is not followed. Predictive EDAS also utilizes road curvature information to preemptively alert the driver to reduce speed before approaching a sharp curve, ensuring a safe turn. An example of such a scenario is depicted in Figure 6.4g. Additionally, when the host vehicle approaches a signalized intersection, the system displays the remaining time (less than 10 s) for the current TL phase on the HUD. This feature is demonstrated for green and red phases in Figures 6.4h and 6.4i, respectively. Since predictive EDAS leverages Signal Phase and Timing (SPaT) information from TL infrastructure, providing real-time countdown of the remaining TL signal phase information can assist drivers in making informed decisions about their next actions.

6.1.4 Objective Evaluation Results

To assess the driving performance of the proposed predictive EDAS, an objective evaluation was conducted in this dissertation. This evaluation was carried out by defining key performance indicators (KPIs), including energy savings, reduction in speed limit violations, and the number of TL signals crossed during the green phase. The results of this objective analysis are discussed in detail in this section.

Energy Savings

Figure 6.5a presents a summary of the energy savings achieved by participants in urban, highway, and overall trip using a box plot. Participants driving with the continuous feedback from the predictive EDAS achieved an average overall energy savings of 9.82% (median 8.46%) for the entire trip. In the urban segment, the average energy savings were 4.6% (median 6.24%), while the highest savings were observed in the highway segment, with an average of 11% (median 8.8%), compared to driving without the assistance from predictive EDAS. The lower energy savings in the urban segment as compared to the highway segment can be attributed to the dense nature of traffic in urban scenarios, which led to frequent acceleration and deceleration of the host vehicle, thereby reducing potential energy savings. The data also indicates that a few drivers could not save energy despite using the predictive EDAS. Further classifying the energy savings across urban, highway and overall trips based on driving styles (cautious, sporty and normal driving) provided insights into which driver categories benefited the most. As illustrated in Figures 6.5b, 6.5c, and 6.5d, the impact of predictive EDAS varied depending on



(a) Energy savings in the driven route

(b) Energy savings according to driving styles: urban trip

(c) Energy savings according to driving styles: highway trip

(d) Energy savings according to driving styles: overall trip

Figure 6.5: Comparison of energy savings of thirty-four participants

driving style. In the urban segment (Figure 6.5b), drivers with a normal driving style achieved the highest energy savings, with an average of 6.52% (median 7.8%), followed by drivers with a sporty style with an average of 4% (median 3.12%). Cautious drivers were able to save comparatively less since their natural driving behavior aligned closely with the recommendations of the predictive EDAS. In the highway segment (Figure 6.5c), predictive EDAS in general has helped drivers from all categories of driving styles to save energy, especially for drivers with a normal driving style who achieved an average savings of 10.28% (median 9.24%). Meanwhile, drivers with sporty and cautious styles saved comparatively less with average savings of 9.25% (median 5.59%) and 9.08% (median 7.58%), respectively.

Road Speed Limit Violations

The use of predictive EDAS resulted in a significant reduction in overall speed limit violations throughout the trip, with an average decrease of 46.18% (median 50.5%). The system effectively enhanced driver awareness of overspeeding, leading to a reduction in overall mean speed. This, in turn, contributed indirectly to lower energy consumption, highlighting the dual benefits of adhering to speed limits and enhanced energy efficiency.

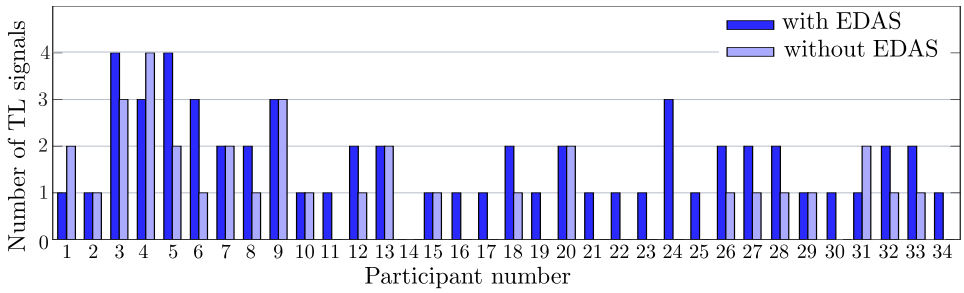


Figure 6.6: Comparison of traffic light signals crossed in green phase

Crossing Through Green Traffic Light Signals

Figure 6.6 compares the number of TL signals crossed during the green phase by all participants, both with and without assistance from predictive EDAS. The implementation of the green-wave algorithm 3.1, introduced in Section 3.1.2, generally increased the likelihood of crossing TL signals during the green phase. On average, the number of stops at red TL signals was reduced by 60% when using predictive EDAS. The empty entries in Figure 6.6 indicate cases where drivers stopped at all TL signals. In some instances, drivers also stopped at more TL signals despite using predictive EDAS. However, it is important to emphasize that this does not reflect the ineffectiveness of the green-wave algorithm but rather the stochastic nature of traffic, which can lead to unfavorable scenarios. For example, encountering a slower leader vehicle may deactivate the green-wave feature, thereby increasing the chances of stopping at a red light. Despite such occurrences, the results clearly demonstrate that under favorable conditions, the green-wave algorithm effectively enhances the chances of crossing TL signals during the green phase.

6.2 Evaluation II: Efficiency Improvement through Driver Behavior Modeling

The findings from the previous section demonstrated that predictive EDAS effectively encouraged drivers to adopt a more energy-efficient driving style by providing continuous feedback across various traffic conditions. In this section, the focus is on enhancing the effectiveness of predictive EDAS through driver behavior modeling using machine learning techniques. Additionally, the proposed learning-based driver speed-tracking error compensation approach, introduced in Chapter 3 (Section 3.1.5), is validated through driver-in-the-loop experiments.

6.2.1 Experimental Procedure

The procedure for the experimental study is illustrated in Figure 6.7. The study was conducted using a dynamic driving simulator and involved two participant groups. The first group, the offline training group, consisted of thirty-four volunteers whose driving data from Experiment I (Section 6.1.2) on an urban route of 3.55 km was utilized for developing clustering, classification, and prediction models, which are further discussed in Section 6.2.2. The second group, the online testing group, comprised six new volunteers aged between 22 and 28 years, all possessing valid driving experience. At the beginning of the experiment, participants were given a brief introduction to the study and its objectives, followed by a short route and multiple practice drives to familiarize themselves with the driving simulator and its controls. The experimental procedure consisted of four driving tests. In driving test I (*V1*), participants drove on the urban route without assistance from EDAS, navigating realistic traffic conditions, stop signs, and deterministic traffic light signals. After this test, participants were introduced to EDAS features through a presentation and practice test drives to become familiar with its recommendations. In driving test II (*V2*), participants repeated the same route with assistance from EDAS, but without driver error compensation. In the next phase, driving test III (*V3*) and Driving Test IV (*V4*) were conducted, where participants tested EDAS with stochastic and deterministic driver speed error compensation (cEDAS), respectively. To prevent potential bias from participants memorizing the driving route, the order of driving tests II, III, and IV was randomized for each driver.

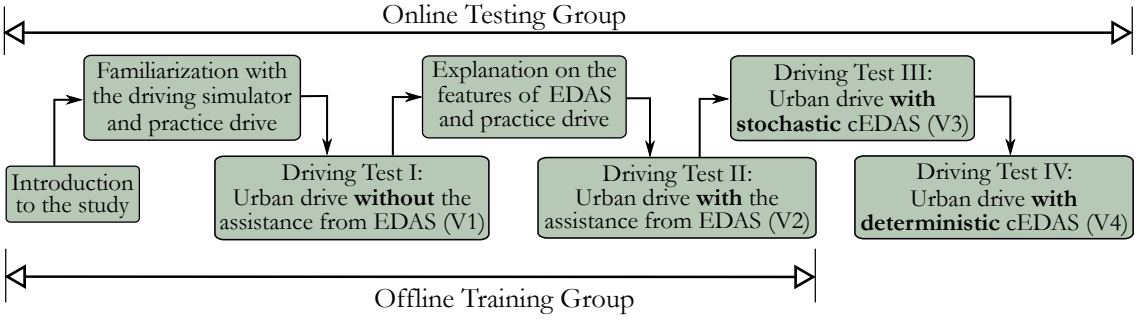


Figure 6.7: Experimental procedure

6.2.2 Offline Evaluation Results

Evaluation of Clustering Methods

For the offline dimentionality reduction, nine input variables from the offline testing group (34 participants) were measured, such as the variance of speed σ_v^2, mean speed error μ_{Δ_v}, the absolute mean speed error $\mu_{|\Delta_v|}$, the variance of speed error $\sigma_{\Delta_v}^2$, the acceleration variance σ_a^2, the mean error of acceleration μ_{Δ_a}, the absolute mean error

of acceleration $\mu_{|\Delta_a|}$, the acceleration error variance $\sigma^2_{\Delta_a}$ and the mean acceleration μ_a. To retain the most relevant information and enable dimensionality reduction, PCA was performed on the input variables, following the steps mentioned in Algorithm 2.1. The number of principal components to be retained in this study was determined based on the criteria established in [CMV10]: (i) identifying the 'elbow' point in the scree plot after plotting the eigenvalues, (ii) retaining principal components with eigenvalues greater than 1.0, (iii) selecting components that contribute to at least 75% of the cumulative explained variance [BG20], and (iv) keeping only interpretable components. As shown in the scree plot in Figure 6.8, three principal components (PC_1, PC_2, and PC_3) were selected based on these criteria, as they capture the highest proportion of variance in the dataset. Furthermore, the effect of different time-series window sizes on the cumulative percentage of explained variance was analyzed. The results indicated that a window size of 21 s was optimal for this study.

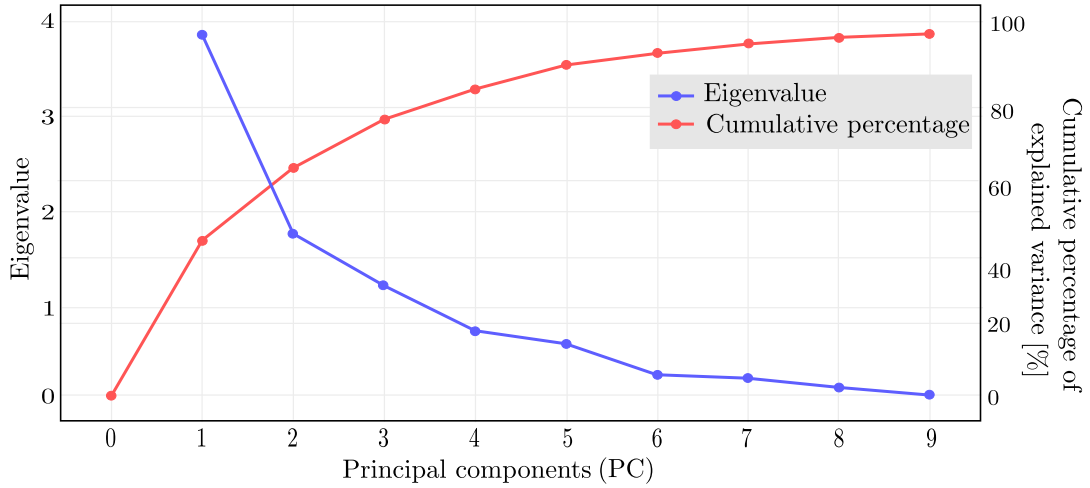


Figure 6.8: Scree plot with principal components

In the next step, the correlation between the original input variables and the principal components (PCs) was analyzed by examining the respective loadings, along with the corresponding rotated components (RCs) obtained after applying the varimax rotation technique. The results are summarized in Table 6.1. A key observation is that the varimax rotation improved the interpretability of the components by redistributing the loadings, making each component more distinctly representative of specific aspects of driving behavior. In the rotated component matrix, RC1 captures both speed and acceleration errors but does not distinguish between variance and mean. RC2, on the other hand, is strongly correlated with both actual speed and acceleration, highlighting the importance of variance in these parameters. Lastly, RC3 exhibits a strong positive correlation with the mean of acceleration error but a negative correlation with mean acceleration. This suggests that while both speed and acceleration errors are significant, variance plays a more critical role than the mean. Consequently, after the dimensionality reduction process, the variables σ^2_v, $\sigma^2_{\Delta v}$, σ^2_a, and $\sigma^2_{\Delta a}$ were identified as the most relevant

Input variables	Principal components			Rotated components		
	PC1	PC2	PC3	RC1	RC2	RC3
σ_v^2	0.2	0.39	-0.54	0.1	**0.87**	-0.15
μ_{Δ_v}	0.4	-0.26	0.12	**0.84**	-0.07	-0.15
$\mu_{\lvert\Delta_v\rvert}$	0.43	-0.24	0.1	**0.90**	-0.02	-0.15
$\sigma^2_{\Delta_v}$	0.44	-0.1	0.1	**0.88**	0.13	-0.04
σ_a^2	0.21	0.57	-0.26	0.15	**0.87**	0.23
μ_{Δ_a}	0.02	0.33	0.65	0.12	-0.08	**0.83**
$\mu_{\lvert\Delta_a\rvert}$	0.44	0.12	-0.01	**0.77**	0.41	0.05
$\sigma^2_{\Delta_a}$	0.4	0.11	0.15	**0.76**	0.28	0.18
μ_a	0.11	-0.49	-0.4	0.25	-0.16	**-0.77**

Table 6.1: Principal component and rotated loadings for the input variables

input variables for further analysis.

The principal components identified through PCA can be used for clustering the driver types via hierarchical cluster analysis (HCA). In this study, agglomerative clustering with the merge criteria as Ward linkage has been adopted to iteratively merge clusters with the smallest euclidean distance. The HCA updates the distance matrix and a dendrogram in each iteration, ultimately determining the optimal number of clusters based on the generated dendrogram. Upon performing HCA on the principal components, two clusters were obtained, which are given the cluster ID as 0 and 1 (orange and green) in the dendrogram of Figure 6.9.

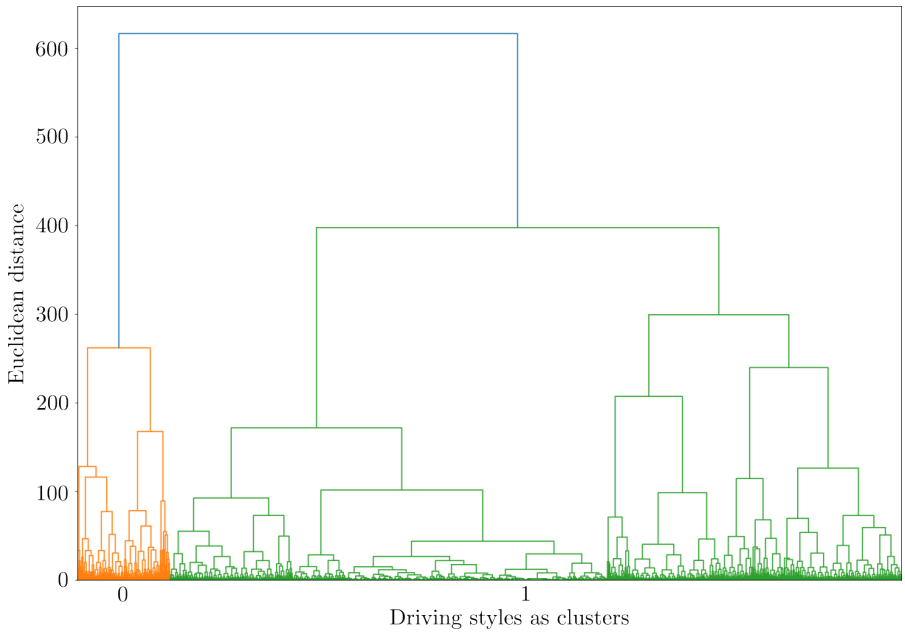


Figure 6.9: HCA dendogram with window size of 21 seconds

Evaluation of Classification Methods

In this subsection, the classification performance of a temporal convolutional network (TCN) is compared with that of a support vector machine (SVM) algorithm. The dataset for each cluster is split into 80% for training and 20% for testing both classifiers. To optimize the hyperparameters of the TCN, a Bayesian optimization approach using tree-structured parzen estimators is employed, whereas the hyperparameters for the SVM are manually tuned using a grid search. The optimized hyperparameters for both TCN and SVM are presented in Table 6.2. To enhance the training efficiency of the TCN, residual blocks were incorporated. Additionally, each temporal block consisted of two dilated convolutional layers with a ReLU as the activation function. To further improve model performance, weight normalization was applied to accelerate the training process, while dropout was used as a regularization technique to mitigate the risk of overfitting. The classification performance results for both TCN and SVM are summarized in Table 6.3. The results indicate that the proposed TCN achieves superior performance in terms of precision, recall, and F1 score on the test datasets for both clusters, thereby demonstrating a more effective classification capability compared to the SVM.

TCN		**SVM**	
Hyperparameters	**Values**	**Hyperparameters**	**Values**
Neurons in a hidden layer	79	Kernel	RBF
Batch size	$2^{12} = 4096$	Regularization	2.9724
Learning rate	$1.914e^{-}3$	gamma	0.25
Dropout rate	0.0716		
Temporal blocks	4		
Number of filters	5		
Optimizer	AdamW		

Table 6.2: Hyperparameters for TCN and SVM

Cluster	**TCN**			**SVM**		
	Precision	Recall	F1 score	Precision	Recall	F1 score
0	1.00	1.00	1.00	0.99	0.99	0.99
1	0.97	0.98	0.97	0.94	0.91	0.93

Table 6.3: Classification performance for TCN and SVM

Evaluation of Speed-Tracking Error Prediction Models

This study investigates both stochastic and deterministic approaches for predicting driver speed-tracking errors. In this subsection, the performance evaluation focuses

on comparing the stochastic volatility (SV) model with the deterministic stacked-GRU model.

(a) Stochastic Volatility (SV) Model

The SV model for the prediction of driver speed-tracking error $\Delta v_{\mathrm{h,t+1}}$ is modeled in this work through a uni-variate stochastic process. The stochastic differential equation (2.31) transformed using Euler-Maruyama scheme is given by

$$X_{t+1} = X_t + \gamma(\mu - X_t)\Delta t + \nu \Delta W_t \tag{6.1}$$

where, $X_t = \ln \frac{\Delta v_{\mathrm{h},t}}{\Delta v_{\mathrm{h},t-1}}$ is the rate of driver speed-tracking error variation. Posterior for all the parameters μ, γ and ν in (6.1) is obtained through No-U-Turn Sampler approach in monte carlo Markov chain (MCMC) simulation. An open-source probabilistic programming language called PyMC3 was used to run the MCMC simulation. The calculated posterior mean for the parameters is substituted in (6.1) to obtain X_{t+1}. Moreover, the speed-tracking error can be updated using

$$\Delta v_{\mathrm{h},t+1} = \Delta v_{\mathrm{h},t} \times \exp(\kappa_t) \tag{6.2}$$

where, κ represents a random variable that is used to model the stochastic component of the driver's speed-tracking error. Moreover, the logarithmic is introduced in (6.2), thus modifying the equation as

$$\begin{aligned} \ln(\Delta v_{\mathrm{h},t+1}) &= \ln(\Delta v_{\mathrm{h},t} \times \exp(\kappa_t)) \\ &= \ln(\Delta v_{\mathrm{h},t}) + \ln(\exp(\kappa_t)) \\ &= \ln(\Delta v_{\mathrm{h},t}) + \kappa_t \end{aligned} \tag{6.3}$$

The X_{t+1} from (6.1) is used to obtain $\kappa \sim \mathcal{N}(0, \exp(X_{t+1}))$. Substituting κ in (6.2), the driver speed-tracking error for the next step $\Delta v_{\mathrm{h},t+1}$ can be determined. The parameters in the proposed SV model presented in Table 6.4 have been tuned manually after carefully analyzing the prediction accuracy.

Variables	**Cluster 0**	**Cluster 1**
$\mathcal{N}(\mu_{\mathrm{Euler}} \mid \mu, \sigma^2)$	$\mu = 0$ $\sigma^2 = 0.1$	$\mu = 0$ $\sigma^2 = 0.1$
$\mathcal{U}(\lambda_{\mathrm{Euler}} \mid l, u)$	$l = 0.25$ $u = 1.0$	$l = 0.3$ $u = 1.0$
$\mathcal{U}(\sigma_{\mathrm{Euler}} \mid l, u)$	$l = 0.001$ $u = 0.25$	$l = 0.001$ $u = 0.3$

Table 6.4: Euler-Maruyama variables for Clusters 0 and 1

(b) Deterministic Stacked-GRU Model

To predict the driver speed-tracking error, in addition to the stochastic model, a deterministic stacked-GRU model is studied in this work. The internal cell structure of the deterministic GRU is discussed in Chapter 3 (Section 2.2.4). Figure 6.10 illustrates the architecture of the stacked GRU, which comprises an input layer, multiple stacked GRU layers, and an output layer. The input variables for the deterministic stacked-GRU model include the historical speed information of the host vehicle with assistance from predcitive EDAS ($v_{\mathrm{h,EDAS}}$), the optimal advisory speed ($v_{\mathrm{h,opt}}$), and the speed tracking error (Δv_{h}). Additionally, the hyperparameters of the stacked-GRU prediction models are optimized using tree Parzen estimators for each cluster type, as detailed in Table 6.5. In the output layer, the model predicts the driver's speed-tracking error for the next time step. The GRU model was implemented using the Keras deep learning library [Bro18] in Python, and the training of the deep learning prediction models was performed on a GPU cluster. In the preprocessing step, the training and test datasets for each cluster were prepared using an 80%-20% split within a 21 s window.

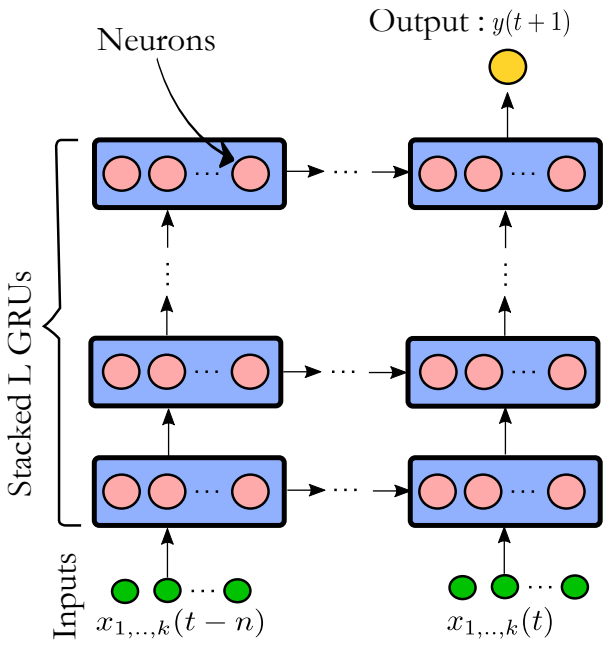


Figure 6.10: Architecture of deterministic stacked-GRU model

The prediction accuracy of the driver speed-tracking error, estimated using both the stochastic volatility (SV) model and the deterministic stacked-GRU model for Clusters 0 and 1, is summarized in Table 6.6. To assess prediction accuracy, mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE) were used as evaluation metrics. The results indicate that the deterministic stacked-GRU model achieves lower MAE, MSE, and RMSE (in m/s) compared to the stochastic volatility (SV) model, demonstrating its superior predictive performance.

Hyperparameters	GRU	
	Cluster 0	Cluster 1
Hidden layers	3	2
Neurons in a hidden layer	109	113
Batch size	$2^{13} = 8192$	$2^{11} = 2048$
Learning rate	$4.497e^{-}3$	$4.031e^{-}3$
Dropout rate	0.1351	0.166
Temporal blocks	-	-
Number of filters	-	-
Optimizer	AdamW	AdamW

Table 6.5: Hyperparameters of the GRU prediction models for Clusters 0 and 1

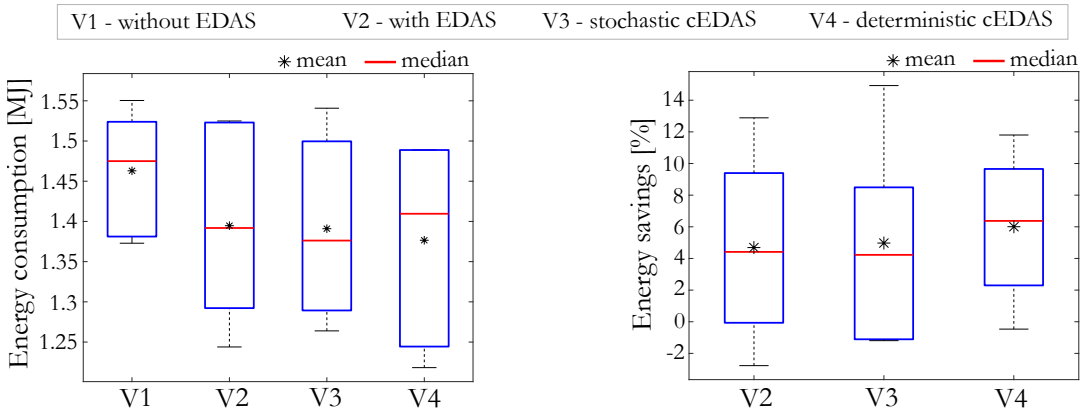
Cluster	GRU			SV		
	MAE	MSE	RMSE	MAE	MSE	RMSE
0	0.114	0.186	0.431	0.136	0.672	0.820
1	0.119	0.246	0.496	0.158	0.501	0.708

Table 6.6: Summary of prediction accuracy (in m/s) for both GRU and SV models for Clusters 0 and 1

6.2.3 Online Evaluation Results

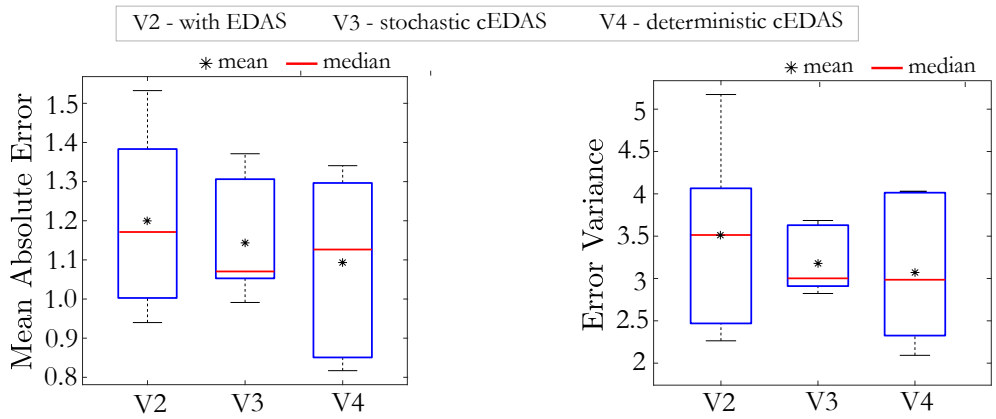
In this section, the evaluation results of the proposed EDAS considering the stochastic and deterministic driver speed error compensation (cEDAS) for six new drivers (online testing group) are presented. Figure 6.11a summarizes the overall energy consumption results for four driving tests *V1* (without EDAS), *V2* (with EDAS), *V3* (stochastic cEDAS) and *V4* (deterministic cEDAS). It can be noticed in general that participants driving with the assistance from EDAS (*V2*) and cEDAS (*V3* & *V4*) have consumed less energy as compared to without the assistance from EDAS (*V1*). It is evident from Figure 6.11b as well, that drivers in *V2*, *V3* and *V4* achieved average energy savings up to 4.7%, 4.92% and 6% respectively as compared to *V1*. It can be concluded that the drivers with deterministic cEDAS (*V4*) can achieve higher energy savings by considering the driver error. Moreover, the energy savings with stochastic cEDAS (*V3*) can be marginally higher compared to the EDAS without speed compensation (*V2*).

Furthermore, the absolute mean speed error $\mu_{|\Delta_v|}$ and the speed error variance $\sigma^2_{\Delta_v}$ of the online testing group are summarized in Figure 6.12a and 6.12b, respectively. The results demonstrate that the stochastic cEDAS (*V3*) reduced the absolute mean speed error of drivers' by 4.7% and the speed error variance by 9.25%, whereas, the deterministic cEDAS (*V4*) reduced the absolute mean speed error of drivers by 8.92% and the speed error variance by 12.23% as compared to EDAS without speed compensation (*V2*). This shows that considering the driver speed-tracking error while developing EDAS can aid



(a) Energy consumptions (b) Energy savings

Figure 6.11: Results of online testing group



(a) Absolute mean speed error (b) Variance speed error

Figure 6.12: Speed error comparison for online testing group

drivers to track the advisory speed more closely and thus correlate with the resulting energy saving benefits.

To give more insights into the speed tracking behaviors, a short stretch of speed profiles from one of the drivers is picked and illustrated in Figure 6.13. It can be noticed from Figure 6.13a that the driver with only EDAS (*V2*) has exhibited speed-tracking error while tracking an eco-driving advisory speed v_{EDAS} (blue). In the Figure 6.13b and 6.13c, the driver speed-tracking error is compensated by adjusting the advisory speed v_{cEDAS} at several sections (highlighted in grey) using the stochastic and deterministic approaches respectively (dotted green). In the Figure 6.13b between 3250-3340 m, the speed compensation does not take place due to a constraint on maximum road speed limit $v_{\max}$. Furthermore, an important observation can be made from 3430 m onwards, that the advisory speed v_{EDAS} in *V2* is seen to reduce as the host vehicle is approaching an upcoming traffic light red phase. In Figure 6.13b, besides implementing a speed-

tracking error compensation, *V3* was able to provide an early deceleration suggestion to the driver as compared to the *V4*.

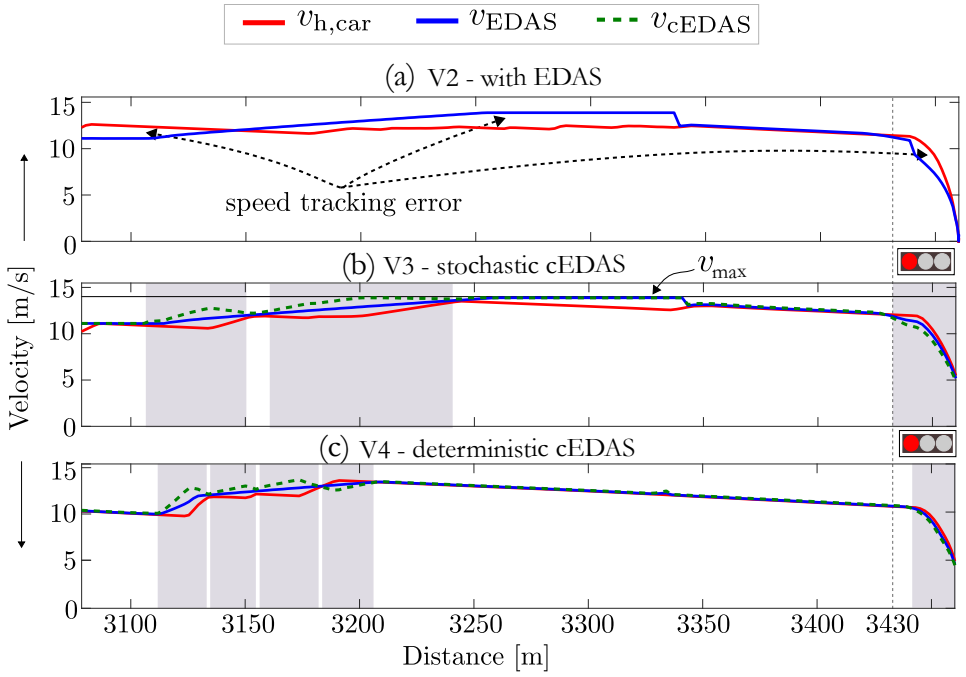


Figure 6.13: Speed profiles of one of the participants from online testing group

6.3 Summary

This chapter focuses on the objective evaluation of the predictive eco-driving assistance system (EDAS) through driver-in-the-loop tests. To validate and analyze its performance, two evaluations involving test subjects were conducted using a dynamic driving simulator. In Evaluation I, 34 participants drove the host vehicle both with and without assistance from the predictive EDAS in a virtual driving environment consisting of urban and highway road segments. The simulation included realistic traffic conditions, deterministic traffic light signals, and stop signs. The predictive EDAS provided continuous feedback, encouraging drivers to follow energy-optimal advisory speeds determined using dynamic programming in freeway scenarios, maintain a green-wave optimal speed to pass through signalized intersections during the green phase whenever possible, perform energy-efficient car-following while tracking a leader vehicle, and travel at safe speeds through turns and curved roads. The objective analysis revealed that drivers using the predictive EDAS achieved mean energy savings of up to 10%, reduced speed limit violations, and avoided unnecessary stops at signalized intersections. Additionally, the driver behavior analysis indicated that the transition between reference tracking and car-following MPCs was smooth and stable. In Evaluation II, historical driving data from the previous participants were analyzed to classify their driving styles based on

speed-tracking error. A learning-based approach was proposed to model driver behavior, employing unsupervised learning techniques such as principal component analysis (PCA) and hierarchical cluster analysis (HCA) for clustering driving data. Furthermore, a supervised deep-learning method, the temporal convolutional network (TCN), was used for driving style classification. To predict driver speed-tracking errors, both stochastic volatility (SV) and deterministic gated recurrent unit (GRU) models were compared, with results demonstrating superior prediction accuracy using the deterministic GRU model. Moreover, participants testing the predictive EDAS with driver error compensation exhibited improved advisory speed tracking and achieved higher energy savings, thereby enhancing the system's overall efficiency. In the next chapter, the predictive EDAS will be evaluated subjectively using state-of-the-art user acceptance models.

7 Subjective Evaluation of Predictive EDAS

To successfully incorporate predictive EDAS into the transportation sector, the user acceptance of such a technology is pivotal, which is the extent to which a user posseses a positive intention to use the system in their daily life if it is made accessible [Ade09]. In the previous chapter, while an objective evaluation of the predictive EDAS was carried out, this chapter primarily focuses on the evaluation in a subjective manner. The subjective evaluation refers to the assessment process in which the human drivers quantify the performance of the predictive EDAS based on individual experiences while testing the system and answering survey questionnaire. To identify and analyze the factors influencing user acceptance, this chapter employs the state-of-the-art acceptance models, namely the Technology Acceptance Model (TAM) and the Theory of Planned Behavior (TPB), as introduced in Chapter 2 (Section 2.3).

7.1 Experimental Procedure

A total of 41 participants (37 male, 4 female) answered a survey questionnaire after testing the predictive EDAS on a driving simulator, following the experimental procedure outlined in Figure 6.1. The survey questionnaire was designed based on the items of the constructs PU, PEoU, PBC, SN, BI (Table 7.2), and ATT (Table 7.3), along with a feedback questionnaire (Table 7.1). To measure the questionnaire items a 5-point Likert scale (1=strongly disagree, 3=neutral and 5=strongly agree) was used.

Table 7.1: Feedback questionnaire

Items	Feedback questions
Q1	Please give your feedback on whether the positioning/size of the pEDAS visualization on the heads-up display is appropriate
Q2	Are you able to guess the reasons behind the suggestions given by pEDAS? Were there any situations where the suggestions given by pEDAS were ambiguous?
Q3	Please give your feedback if you would like to have some additional suggestions to be given by pEDAS

pEDAS = predictive EDAS

Table 7.2: Survey questionnaire Items for assessing Perceived Usefulness, Perceived Ease of Use, Perceived Behavioral Control, Subjective Norms, and Behavioral Intention

Items	Survey Questionnaire
	<table><tr><td></td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td></td></tr><tr><td>Strongly disagree</td><td>◯</td><td>◯</td><td>◯</td><td>◯</td><td>◯</td><td>Strongly agree</td></tr></table>
Perceived Usefulness	
PU1	Using the pEDAS would improve my driving performance
PU2	Using the pEDAS would improve my energy savings
PU3	Using the pEDAS would enable me to react to unsafe driving situations more quickly
PU4	I assume that an pEDAS equipped car would be useful in my daily life
Perceived Ease of Use	
PEoU1	I find it easy to interpret the suggestions given by pEDAS
PEoU2	Interacting with the pEDAS would not require a lot of cognitive effort
PEoU3	I find the pEDAS to be distracting while driving
PEoU4	Learning to use the pEDAS would be easy for me
Perceived Behavioral Control	
PBC1	I have control over using pEDAS
PBC2	I do have the knowledge necessary to use pEDAS
PBC3	Given the resources, opportunities and knowledge it takes to use pEDAS, it would be easy for me to use it in my daily commute
Subjective Norms	
SN1	People whose opinions I value would think that I should use pEDAS
SN2	People who influence my behavior would think that I should pEDAS
Behavioral intention	
BI1	If the pEDAS is available in the market at an affordable price, I intend to purchase this feature
BI2	Assuming that the pEDAS is available in my car, I intend to use it regularly when I am driving

pEDAS = predictive EDAS

Table 7.3: Survey Questionnaire - Attitude toward behavior

I find predictive EDAS when I am driving to be:							
		1	2	3	4	5	
ATT1	Bad	◯	◯	◯	◯	◯	Good
ATT2	Useless	◯	◯	◯	◯	◯	Useful
ATT3	Undesirable	◯	◯	◯	◯	◯	Desirable
ATT4	Ineffective	◯	◯	◯	◯	◯	Effective
ATT5	Drowsy	◯	◯	◯	◯	◯	Alerting
ATT6	Unpleasant	◯	◯	◯	◯	◯	Pleasant
ATT7	Annoying	◯	◯	◯	◯	◯	Not at all annoying
ATT8	Irritating	◯	◯	◯	◯	◯	Likeable
ATT9	Worthless	◯	◯	◯	◯	◯	Assisting

7.2 Hypothesis

To predict the user acceptance of predictive EDAS, following hypotheses are proposed in this work based on TAM [Dav85] and TPB [Ajz91].

H1: Perceived Ease of Use (PEoU) is a significant positive predictor of Perceived Usefulness (PU), implying that the individuals who believe that using predictive EDAS would be free of effort are more likely to believe that using predictive EDAS would enhance their driving performance.

H2: Perceived Usefulness (PU) has a positive significant effect on Attitude towards behavior (ATT), implying that the individuals who believe that using predictive EDAS would enhance their driving performance are more likely to possess positive feelings towards predictive EDAS behavior.

H3: Attitude toward behavior (ATT) has a positive significant effect on Behavioral Intention (BI), implying that the individuals who possess positive feelings towards predictive EDAS behavior are more likely to intend to use predictive EDAS in daily life.

H4: Perceived Ease of Use (PEoU) has a positive significant effect on Attitude toward behavior (ATT), implying that the individuals who believe that using predictive EDAS would be free of effort are more likely to possess positive feelings towards predictive EDAS behavior.

H5: Perceived Usefulness (PU) has a positive significant effect on Behavioral Intention (BI), implying that the individuals who believe that using predictive EDAS would enhance their driving performance are more likely to intend to use predictive EDAS in daily life.

H6: Subjective Norms (SN) have a positive significant effect on Behavioral Intention

(BI), implying that the individuals who believe that important people in their social network support using predictive EDAS are most likely to intend to use predictive EDAS in daily life.

H7: Perceived Behavioral Control (PBC) has a positive significant effect on Behavioral Intention (BI), implying that the individuals who perceive ease in using predictive EDAS in the presence of necessary resources are most likely to intend to use predictive EDAS in daily life.

7.3 Evaluation Results

In this study, structural equation modeling (SEM) [Mar98, UB12] was employed to analyze the relationships between latent constructs and assess the impact of various factors on the behavioral intention to use predictive EDAS. Unlike traditional regression techniques, SEM allows for the simultaneous analysis of multiple dependent variables, making it particularly suitable for subjective evaluation studies. The SEM approach consists of two key steps: (1) measurement model analysis, which evaluates the reliability and validity of the latent constructs, and (2) structural model analysis, which examines the hypothesized relationships between variables [Mar98]. All statistical analyses in this work were conducted using IBM SPSS [IBM21].

7.3.1 Measurement Model Analysis

The measurement model deals with the relationship between the latent constructs and its indicators [UB12]. To validate the goodness-of-fit of the measurement model, a confirmatory factor analysis must be conducted. In this analysis, the reliability and validity of the items in terms of indicator reliability, construct reliability, convergent validity and discriminant validity are tested [GP21]. Indicator reliability represents the proportion of the variance in an item that is explained by the latent construct [HHRS14]. It can be explained from the standardized factor loading λ. According to [HHRS14], the items with the factor loading above the desired threshold of 0.7 must be maintained. To estimate the internal consistency of the items, construct/composite reliability (CR) and Cronbach's alpha (Cα) are widely used methods in the literature [HHRS14]. CR is considered as an appropriate measure as compared to Cα, as the former assumes the varying factor loading in contrast to the latter [HHRS14]. The literature recommends the CR and the Cα values to be greater than 0.6 and 0.7 respectively. Convergent validity is defined as the extent to which an item correlates positively with other items of the same latent variable. To test a convergent validity, average variance extracted (AVE) is used. An AVE value of greater than 0.5 is a must for the latent variable to be able to explain more than 50% of the variance of its items [HHRS14, GP21]. Discriminant validity is defined as the extent to which a latent variable is unique and less correlated

with other latent variables. To validate the discriminant validity, a criteria suggested in [FL81] is used, which states that the square root of AVE of each latent variable must be larger than its correlation with other latent variables.

Table 7.5 presents the descriptive statistics (mean M and standard deviation SD) for 41 participants, along with the standardized factor loading λ, cronbach's alpha Cα, composite reliability CR, and average variance extracted AVE for each item. Furthermore, the bi-variate correlations between the latent constructs are summarized in the Table 7.4. The λ for the items PEOU2-PEOU3 and ATT1-ATT5 is found to be less than 0.7 (Table 7.5), however more evidence is necessary to delete the aforementioned items. It was noticed that the items of the construct ATT were highly correlating with the items of PU, which is not intended. Thus, omitting the items ATT1-ATT5 from the analysis resulted in satisfying the discriminant validity criteria for the inter-correlation construct ATT and other constructs as well (Table 7.4). To maintain consistency with the scale, the reverse scaled item PEoU3 was reversed. Furthermore, deleting the negative loaded item PEoU3 (Table 7.5) has improved the AVE of the construct PEoU. It can be observed from Table 7.5, that the CR and AVE are higher than the desired values of 0.6 and 0.5 respectively. The Cα for all the other constructs is found to be greater than 0.7 except for PEoU (0.596), which shows a moderate reliability as the value lies between 0.5-0.7 [HBMC04]. Deleting the item PEOU2 did not improve the Cα much, therefore this item was retained.

Table 7.4: Inter-construct correlation matrix of the TAM and TPB constructs.

Factors	PU	PEoU	ATT	PBC	SN	BI
PU	**0.803**†					
PEoU	0.450**	**0.641**†				
ATT	0.772**	.328*	**0.796**†			
PBC	.376*	.595**	0.106	**0.871**†		
SN	.492**	-0.041	.460**	-0.059	**0.928**†	
BI	.622**	.312*	.395*	.495**	.352*	**0.91**†

† square root of AVE.
** Correlation is significant at the 0.01 level (2-tailed).
* Correlation is significant at the 0.05 level (2-tailed).

7.3.2 Structural Model Analysis

To test the hypotheses H1-H7 discussed in Section 7.2 and explain the variance contributed by the latent constructs on the behavioral intention or the actual use of the system, a hierarchical regression analysis is performed in this study. Four models that use perceived usefulness (PU), attitude towards behavior (ATT) and behavioral intention (BI) as dependent variables are considered as illustrated in Table 7.6.

Table 7.5: Descriptive statistics along with standardized factor loading, cronbach's alpha, composite reliability and average variance extracted

Constructs	Mean	SD	λ	$C\alpha$	CR	AVE
Perceived Usefulness(PU)				0.799	0.879	0.646
PU1	4	0.847	0.837			
PU2	4.35	0.735	0.792			
PU3	3.625	1.21	0.747			
PU4	4.05	0.959	0.837			
Perceived Ease of Use (PEoU)				0.596	0.772	0.537
PEoU1	4.125	0.882	0.724			
PEoU2	3.75	0.839	0.585			
PEoU3†	3.65	1.026	-0.193*			
PEoU4	4.35	0.833	0.863			
Attitude of Behavior (ATT)				0.828	0.873	0.634
ATT1	4.25	0.808	0.464*			
ATT2	4.225	0.861	-0.034*			
ATT3	4.05	0.959	0.249*			
ATT4	4.2	0.648	0.687*			
ATT5	3.85	0.975	0.579*			
ATT6	3.8	0.822	0.817			
ATT7	3.575	0.957	0.820			
ATT8	3.725	1.085	0.802			
ATT9	4.375	0.627	0.743			
Perceived Behavioral Control (PBC)				0.837	0.904	0.759
PBC1	4.075	0.916	0.869			
PBC2	4.208	0.779	0.858			
PBC3	4.4	0.744	0.888			
Subjective Norms (SN)				0.917	0.949	0.862
SN1	3.675	1.095	0.92			
SN2	3.675	0.971	0.936			
SN3	3.525	0.933	0.93			
Behavioral Intention (BI)				0.792	0.907	0.829
BI1	3.525	1.03	0.911			
BI2	4.175	0.930	0.911			

SD = Standard deviation, λ = Factor loading, $C\alpha$ = Cronbach's Alpha, CR = Composite/Construct Reliability, AVE = Average Variance Extracted,

* omitted from the analysis

† reverse scaled item

Table 7.6: Hierarchical regression analysis on TAM and TPB models

Models	R^2	R^2_{Adj}	ΔR^2	B	$SE\ B$	β	95% CI
1. PEoU → PU							
Step 1: Dependent Variable - PU	0.161	0.139	0.161				
Independent Variable - PEoU				0.478	0.177	0.401*	0.217-1.031
2. PEoU + PU → ATT							
Step 1: Dependent Variable - ATT	0.056	0.031	0.056				
Independent Variable - PEoU				0.270	0.180	0.237	0.023-0.848
Step 2: Dependent Variable - ATT	0.602	0.580	0.546				
Independent Variable - PEoU				-0.099	0.129	-0.087	-0.361-0.162
Independent Variable - PU				0.773	0.109	0.807**	0.553-0.993
3. ATT + PU → BI							
Step 1: Dependent Variable - BI	0.156	0.133	0.156				
Independent Variable - ATT				0.488	0.184	0.395*	0.115-0.862
Step 2: Dependent Variable - BI	0.405	0.373	0.249				
Independent Variable - ATT				-0.262	0.247	-0.212	-0.762-0.238
Independent Variable - PU				0.931	0.236	0.785**	0.452-1.410
4. ATT + PBC + SN → BI							
Step 1: Dependent Variable - BI Independent Variable - ATT	same as model 3 step 1						
Step 2: Dependent Variable - BI	0.193	0.149	0.037				
Independent Variable - ATT				0.365	0.206	0.295	-0.052 - 0.782
Independent Variable - SN				0.208	0.160	0.216	-0.116 - 0.533
Step 3: Dependent Variable - BI	0.425	0.377	0.233				
Independent Variable - ATT				0.263	0.178	0.213	-0.098 - 0.624
Independent Variable - SN				0.273	0.138	0.283	-0.007 - 0.553
Independent Variable - PBC				0.617	0.162	0.489**	0.289 - 0.944

** Correlation is significant at the 0.001 level (2-tailed).
* Correlation is significant at the 0.05 level (2-tailed).

Performing the linear regression analysis on model 1 revealed that the construct perceived ease of use (PEoU) has a positive significant effect on PU with unstandardized coefficients $B = 0.478$, $SE\ B = 0.177$, and standardized coefficient $\beta = 0.401$. PEoU was able to explain 16.1% of variance in PU ($R^2_{Adj} = 0.139$, $\Delta R^2 = 0.161$). Even though the proportion of explained variance in PU is small, it is statistically significant with $p < 0.05$, thus validating hypothesis 1.

The hierarchical linear regression is performed on model 2, in which PEoU is inserted in to the model in step 1. Subsequently PU is added in to the model in the second step. In the first step, PEoU explained only 5.6% of the variance in ATT ($R^2_{Adj} = 0.031$, $\Delta R^2 = 0.056$). In step 2, PU increased the variance by a significant 54.6% ($p < 0.01$), increasing the R^2_{Adj} to 0.58 and ΔR^2 to 0.546. Therefore, PU is found to have a positive significant effect on ATT with unstandardized coefficients $B = 0.773$, $SE\ B = 0.109$ and standardized coefficient $\beta = 0.807$, thus supporting hypothesis 2. However, the addition of PU in turn reduced the effect of PEoU on ATT ($\beta = -0.087$) and it was not found to be significant ($p > 0.05$), thus hypothesis 4 is not supported. This shows that PU is a significant predictor of ATT above and beyond PEoU.

In the first step in model 3, ATT entered and was able to explain only 15.6% of the variance in BI ($R^2_{Adj} = 0.133$, $\Delta R^2 = 0.156$). Subsequently when PU entered the model in step 2, the variance increased to 40.5% ($R^2_{Adj} = 0.373$, $\Delta R^2 = 0.249$, $p < 0.001$). Thus, PU was found to be a significant predictor of BI ($p < 0.001$) with unstandardized coefficients $B = 0.931$, $SE\ B = 0.236$ and standardized coefficient $\beta = 0.785$, thus supporting hypothesis 5. The addition of PU in turn reduced the effect of ATT on BI ($\beta = -0.212$, $p > 0.05$), thus hypothesis 3 is not supported. This implies that PU alone is a significant predictor of BI above and beyond ATT. In the model 4, ATT entered the model in the step 1, subsequently subjective norms (SN) in the step 2 and perceived behavioral control (PBC) in the step 3. Adding the SN in step 2 in turn reduced the contribution of ATT ($\beta = 0.295$) rendering it insignificant. Although, after adding SN to the model the variance increased to 19.3% ($R^2_{Adj} = 0.149$, $\Delta R^2 = 0.037$, $p > 0.05$), the change in the R^2 was found to be very small (ΔR^2=0.037) and was not statistically significant. Thus, it can be concluded that SN is not a significant contributor in predicting the BI, thus hypothesis H6 is not supported. When PBC entered the model in step 3, the variance increased to 42.5% ($R^2_{Adj} = 0.377$, $\Delta R^2 = 0.233$) and it was significant ($p < 0.05$). This implies that PBC alone is a significant predictor of behavioral intention with $B = 0.617$, $SE\ B = 0.162$ and $\beta = 0.489$ above and beyond SN and ATT, thus supporting hypothesis 7.

7.4 Discussion of the Results

In the survey, three feedback questions were asked to the participants which addressed the key areas of predictive EDAS (Table 7.1). The feedback indicated that the participants have found suggestions from predictive EDAS to be very helpful in not only

minimizing the energy consumption in a battery electric vehicle but also additional features such as green-wave, displaying of traffic light remaining time and speed adaptation to a sharp curve were complimentary. The participants were also satisfied with the positioning and size of predictive EDAS visualization on the heads-up display. However, the feedback also indicated potential areas of improvements. Although the driving route was highlighted in the virtual driving environment, a few drivers missed the turns in the driving route while focusing on the suggestions from predictive EDAS. Drivers suggested in the future to display a navigation map on the HUD or provide auditory feedback for navigation. Another feedback from participants is that the suggestions from predictive EDAS to slow down before a stop sign or a sharp turn could appear a bit earlier, so that they had enough time to react. In the driving simulator experiments, the drivers visualized the virtual driving environment on a flat TV monitor with a restricted field of view. Therefore, in the urban environments this resulted in unsafe scenarios at sharp turns.

In future to provide an increased field of view to the driver, displaying the virtual environment using either a curved screen, triple-monitor or a virtual reality headset can be considered. Furthermore, a survey questionnaire based on TAM and TPB was formulated to understand the factors that influence the user acceptance of the proposed predictive EDAS. Considering TAM, the perceived usefulness was found to be the strongest predictor of behavioral intention towards using predictive EDAS. This further implies that persons are likely going to use predictive EDAS if they think it is useful in their daily commute. The driver's belief of usefulness of predictive EDAS is associated with several rewards such as increase in energy savings for the trip, less over speeding, spending less time at traffic light signals and continuous eco-driving assistance throughout the trip. These rewards or performance benefits will in turn contribute to the intention of using predictive EDAS, thereby increasing the acceptance amongst users. This study found that the perceived ease of use is a significant positive predictor of perceived usefulness. This implies that there is a high probability that the drivers will find predictive EDAS to be useful if they find it easy to use.

This is also logical in a way that if users find it easy to perform a behavior, here understanding suggestions from predictive EDAS, it will motivate them to use the technology more often. The analysis also showed that the perceived usefulness is a significant predictor of attitude towards behavior. Attitude, which relates to a person's positive and negative feelings towards performing a certain behavior (here, to use predictive EDAS) is explained by the person's belief of usefulness. The result implies that if a person believes that predictive EDAS is useful, then that person will likely develop a positive attitude towards using it. Moreover, the study points out that the attitude, which is a common construct in both TAM and TPB, did not exhibit a significant contribution in predicting the behavior intention. A possible explanation of the aforementioned discrepancies can be inferred from the participant feedback that predictive EDAS might have created ambiguity and distraction among few drivers. This claim can be further supported from the neutral responses obtained for items 7 and 8 of the Attitude construct and items

2,3 of the PEoU construct as illustrated in Table 7.5. For TPB, the subjective norm was found to have a small explainable variance on the behavioral intention, however, was found to be not statistically significant. This implies that the opinion of important people in one's social network have a very little effect on their intention to use predictive EDAS. Moreover, it was found from the study that the perceived behavioral control was the strongest predictor of behavioral intention in TPB. This implies that the individuals who perceive ease in using predictive EDAS in the presence of necessary resources are most likely to intend to use predictive EDAS in their daily life.

7.5 Summary

This chapter presents the subjective evaluation of the predictive eco-driving assistance system (EDAS) for BEVs. A survey questionnaire was administered to 41 participants to gather their opinions and feelings towards using predictive EDAS. A priori hypotheses was formulated based on the Technology Acceptance Model (TAM) and the Theory of Planned Behavior (TPB) model. The questionnaire included items related to perceived usefulness, perceived ease of use, attitude towards behavior, subjective norms, perceived behavioral control, and behavioral intention. Structural equation modeling was employed to analyze the correlations between these latent constructs and to identify the factors that influence the acceptance of predictive EDAS amongst users. The results reveal that perceived usefulness and perceived behavioral control are the strongest predictors of behavioral intention in using predictive EDAS.

8 Conclusions and Outlook

8.1 Conclusions

Energy efficiency in vehicles has become a key priority for the automotive industry, driven by the urgent challenges of environmental, economic, and regulatory pressures. Promoting energy-efficient driving behavior, commonly known as eco-driving, plays a pivotal role in reducing energy consumption in vehicles, making it a cost-effective strategy for achieving global sustainability goals. The primary limitation of current eco-driving systems available in the market is that they employ simple optimization methods and provide momentary suggestions to drivers that focus on minimizing the fuel or energy consumption within restricted driving scenarios.

The primary contribution of this dissertation is presented in Chapter 3, which introduces a novel architecture for a predictive eco-driving assistance system (EDAS). This system leverages data from on-board sensors, route topology information (road elevation and curvature), and traffic light infrastructure to provide continuous speed advisory feedback to drivers across various driving situations. The proposed EDAS is designed not only to promote a more energy-efficient driving style but also to enhance overall driving safety and comfort. A further significant contribution of this work is the introduction of a learning-based driver behavior modeling framework, which enhances the driver's ability to follow speed advisories effectively, thereby increasing the overall efficiency of the predictive EDAS.

Chapter 4 presented the problem formulations for the proposed Eco-MPCs, namely, the reference tracking MPC and car-following MPC, developed for battery electric vehicles (BEVs) to address multiple control objectives, including energy efficiency, driving safety, and comfort. These optimization problems were formulated in the time domain, with quadratic cost functions subject to linear inequality constraints. Evaluation results demonstrated that the reference tracking MPC yielded significant energy savings, particularly when following green-wave optimal speed (GWOS) trajectories to cross traffic light (TL) signals during green phases and when tracking offline-optimized speed trajectories generated via dynamic programming in freeway scenarios. The car-following MPC effectively planned the host vehicle's speed while tracking a leader vehicle and responding to TL signals. Additionally, the benefits of incorporating leader vehicle's future velocity information to enhance the control performance of the car-following MPC were demonstrated. Further evaluation in a realistic urban scenario confirmed the effective switching performance between Eco-MPCs based on the proposed switching logic. The results also

showed that the controllers require minimal computation time to solve the underlying quadratic optimization problems, thereby validating their suitability for real-time implementation. Furthermore, a more suitable MPC formulation for internal combustion engine (ICE) vehicles is presented in Chapter 4, where an Eco-HMPC (hybrid model predictive control) strategy is proposed for an urban city bus use case. The control problem is formulated as a mixed-integer quadratically constrained quadratic programming (MIQCQP) problem, enabling the simultaneous optimization of continuous engine power output and the discrete control of the engine's ON and OFF states. Simulation results demonstrate significant energy-savings due to incorporating a pulse-and-glide (PnG) driving strategy, minimizing unnecessary stops at signalized intersections with real-time utilization of SPaT data. Furthermore, the analysis indicates that fuel savings improve with an increase in the prediction horizon. This chapter also provides a detailed summary of the execution times associated with the proposed Eco-HMPC strategy for various prediction horizons, highlighting its potential for real-time implementation.

Considering the performance benefits achieved by assuming accurate leader vehicle information in Chapter 4, the contribution towards predicting the leader vehicle's future velocities to improve the efficiency of car-following MPC was focused in Chapter 5. For the data preparation task, route-independent time series driving datasets from urban and highway networks were prepared using a novel approach with the microscopic traffic simulation tool SUMO. For the speed prediction task, recurrent neural networks (LSTM and GRU) and physics-based prediction models (CV and CA) were studied and compared in this dissertation. The results revealed that the LSTM model outperformed the GRU, CV and CA models, effectively capturing historical dependencies from several input features and performing long-term predictions of up to 10 s. This dissertation also identified critical environmental information that enhance prediction accuracy of the speed prediction models. Furthermore, incorporating additional input features, such as data from multiple leader vehicles in the driving route obtained through V2V communication and traffic light signal future phases gathered through V2I has further enhanced prediction accuracy. Chapter 5 also demonstrated that the LSTM prediction model consistently achieved better accuracy across various traffic scenarios (maneuvering a right turn, freeway scenario, approaching a stopped leader vehicle at a traffic light signal, navigating through a roundabout, and approaching a red traffic light signal). Furthermore, energy savings of up to 26% were realized for a host car equipped with the proposed LSTM-based car-following MPC while tracking predicted target vehicle velocities. Additionally, a further improvement of up to 2.5% in average energy savings was achieved compared to the constant speed (CS) model.

In Chapter 6, the effectiveness of the proposed predictive EDAS was evaluated through user studies involving 34 participants using a dynamic driving simulator. To provide drivers with optimal advisory speeds and alerts during unsafe driving situations, a continuous eco-driving feedback system (both visual and acoustic) was developed. Participants drove both with and without assistance from the predictive EDAS in a virtual driving environment that included urban and highway routes with a total length of

13.2 km. The simulation featured realistic driving scenarios, including surrounding traffic, traffic light signals, road signs (e.g., stop signs), and curved roads. The performance of the predictive EDAS was objectively assessed in this dissertation using key performance indicators (KPIs) derived from the driving data collected during the user studies. The results demonstrated that participants driving with the continuous feedback from the predictive EDAS achieved an average overall energy savings of 9.82% for the entire trip, an average savings of 4.6% in the urban segment, while the highest savings were observed in the highway segment, with an average reduction of 11% in energy consumption compared to driving without the assistance from predictive EDAS. Additionally, the findings highlighted the potential of predictive EDAS to improve energy efficiency across drivers with varying driving styles. Moreover, the use of predictive EDAS led to a nearly 50% reduction in overall speed limit violations throughout the trip, while stopping at red TL signals has been decreased on an average by 60%.

Additionally, to model driver behavior and identify driving patterns among 34 participants, data-driven unsupervised learning methods, namely, principal component analysis (PCA) and hierarchical cluster analysis (HCA) were explored in Chapter 6. This dissertation identified key driver-specific features using PCA, including speed variance, absolute mean speed error, speed error variance, and acceleration variance. Furthermore, the optimal number of driving style clusters was determined using HCA. In addition, a supervised deep-learning method named temporal convolutional network (TCN) was employed for driving style classification, achieving high classification accuracy. The effectiveness of the proposed learning-based approach was further validated through experiments involving six participants in a dynamic driving simulator. For the human driver speed-tracking error prediction task, this work compared the performance of deterministic gated recurrent unit (GRU) models and stochastic volatility (SV) models. The results demonstrated that the deterministic GRU model outperformed the stochastic SV model in terms of prediction accuracy. Moreover, the findings indicated that drivers using predictive EDAS with driver error compensation (cEDAS) exhibited improved advisory speed tracking and achieved greater energy savings in urban driving scenarios.

Finally, in Chapter 7, the user acceptance of the proposed predictive EDAS for battery electric vehicles (BEVs) was evaluated through a survey questionnarie based on the constructs of the state-of-the-art technology acceptance model (TAM) and the theory of planned behavior (TPB) model. A priori hypotheses was formulated to assess the significant impact of various constructs (i.e. perceived usefulness, perceived ease of use, attitude toward behavior, subjective norms, and perceived behavioral control) on users' behavioral intention to adopt predictive EDAS. The survey was administered to 41 participants who had previously tested the predictive EDAS in a driving simulator. A subjective evaluation was then conducted based on the collected responses. Subsequently, structural equation modeling (SEM) was applied in a two-step process: first, a measurement model analysis was performed to assess the reliability and validity of the latent constructs, followed by a structural model analysis to test the hypotheses using

hierarchical regression analysis. The results indicated an overall positive user attitude toward predictive EDAS. Additionally, perceived usefulness and perceived behavioral control emerged as the strongest predictors influencing users' behavioral intention to adopt the system.

8.2 Future Work

Looking ahead, this work provides several foundations for future research. These areas of investigation have the potential to enhance the positive outcomes achieved so far and should not be overlooked. The key directions include:

Further Development of the Driving Simulator: The continued development of the driving simulator setup used in this study is an important area for future work. To mitigate the motion-sickness experienced by some participants, future work should enhance the current washout-filter-based motion cueing algorithm used in the simulator by incorporating a more detailed model of the human vestibular system [BBM12]. Furthermore, participants feedback highlighted two key areas for improvement. Firstly, while participants appreciated the force-feedback on the steering wheel, they noted the absence of a recentering mechanism, which caused confusion about the center position while driving. This issue could be addressed by integrating a signal block into the Simucube True Drive and IPG Cockpit package software interface to determine and apply the necessary restoring torque to realign the steering wheel to its center position. Secondly, some participants encountered unsafe situations when navigating sharp turns in urban environments due to the restricted field of view on the flat TV monitor used for visualizing the virtual driving environment. To mitigate this issue, future enhancements to the driving simulator setup could include a curved screen, a triple-monitor arrangement, or a virtual reality headset to provide a more immersive and comprehensive visual experience.

Further Development of the Eco-Driving Feedback System: The proposed eco-driving feedback system (Section 3.1.3) provided drivers with continuous visual speed recommendations, helping them adopt a more energy-efficient driving style. However, subjective evaluations revealed that some participants found the continuous feedback from the predictive EDAS to be distracting, drawing their attention away from surrounding traffic. As a potential improvement, future research could explore the effectiveness of intermittent visual feedback as an alternative. An intermittent eco-driving feedback system ensures that information is provided to the driver only when it is genuinely needed, allowing them to prioritise the traffic environment and minimising unnecessary distractions. Kircher et al. [KFA14] examined the impact of intermittent versus continuous eco-driving feedback on drivers' glance behavior. Building on this, further investigation is needed to determine under which driving conditions intermittent feedback may be more appropriate than continuous feedback (and vice-versa) to minimize driver distraction while maintaining the benefits of predictive EDAS.

Evaluation of Predictive EDAS in Real-world Tests: Throughout this work, the predictive EDAS and its key components were validated extensively, either through simulation or using measurement data from various real-world driving scenarios. To further enhance the accuracy of leader-vehicle speed prediction, future research should explore additional input features that capture traffic rules at non-priority intersections and roundabout situations. Moreover, to fully demonstrate the real-world effectiveness of the proposed predictive EDAS, subsequent studies should focus on driver-in-the-loop experiments involving a larger group of participants as well as tests conducted in real vehicles. This will require, on one hand, the integration of the proposed Eco-MPCs with the vehicle's on-board localization (e.g., GNSS) and perception systems (e.g., radar, camera), as well as the human-machine interface (HMI) and electronic control units (ECUs). On the other hand, it will involve incorporating communication modules to enable real-time data exchange with infrastructure (such as traffic light controllers, road-side units, and cloud servers), as well as between vehicles (through prediction modules and on-board units). Establishing such a connected test platform would not only facilitate comprehensive validation of eco-driving assistance systems in real-world conditions but also open up new opportunities for the development and testing of future automated driving applications.

Appendix

The curated list of the literature on eco-driving studies is presented in Table A.1.

Table A.1: Summary of the curated literature on eco-driving studies.

Authors	Type	Objective of the study	Setup	Major aspects and findings	Energy savings
[BBD+09]	Eco-driving course	Study the long-term benefits of the eco-driving training from the data collected for several months	Field trials (P=10)	Immediate improvement in fuel consumption was observed, but some drivers tend to fall back to their original driving style	5.8%
[DBF+13]	Predictive eco-driving assistance system	Evaluating the efficacy of a predictive advisory system for heavy commercial vehicles	Driving simulator (P=40)	Instructing to drive fuel-efficiently and indication of advisory messages to drivers helped in energy consumption reductions. However, the energy consumption rose after immediate tests without the assistance	6.6 - 12.2%
[VBH+13]	Retrofit driver advisory system	Design and evaluation of the effectiveness of an real-time feedback system that encourages eco-driving in light commercial vehicles	Field trials (P=15)	Using the assistance system has lowered the rate of accelerations and early upshifting of the gears, thus resulting in reductions in the energy consumption	7.6%
[SSKK14]	Eco-driving support system	Evaluate an eco-driving support system in urban and rural environments while communicating with TL signals and traffic signs	Driving simulator (P=30)	Fuel-optimal gear shifting indication and green-wave speed suggestion enabled coasting behaviors and reduced stops at the TL signals	15.9-18.4%
[ZWRZ15]	Eco-driving support system	Investigate an eco-driving support system that provides both static and dynamic feedback to help drivers in reducing emissions and fuel consumption	Driving simulator (P=22)	Suggestions are provided to avoid rapid acceleration and deceleration events, limit engine RPM, avoid unstable speeds on freeways and idling for a longer time	3.43-5.45%

[FKA15]	Eco-driving support system	Investigate various types of in-vehicle eco-driving feedback systems that advise on fuel-efficient driving	Driving simulator (P=24)	Drivers showed a positive attitude towards the eco-driving support systems, however fuel-efficiency was not investigated in their work	-
[SDN15]	Eco-driving training	Study whether the professional bus drivers trained with the eco-driving techniques on a simulator would be able to implement these learnings in daily life	Driving simulator (P=29)	Points out that the simulator-based eco-driving training has significant fuel-saving benefits in real-world driving as well	11.6-16.9%
[JHJ15]	Eco-driving feedback system	Evaluate both visual and haptic feedback systems for the uphill and downhill driving scenarios	Driving simulator (P=22)	Continuous real-time visual feedback to the drivers was proven to be more effective than the haptic-based system	-
[BCV+17]	Interactive guidance system	Measure the contribution of a real-time interactive guidance system to teach eco-driving using immersive simulation	Driving simulator (P=72)	Teaching eco-driving behavior through instructional videos and interactive guidance system had positive impact on fuel savings and CO2 emission reduction	7.42-12.38%
[ACNW17]	Eco-driving training	Evaluate the benefits of an eco-driving based advanced telematics platform in reducing the fuel consumption in freight transport	Field trials (P=25)	The real-time information and metrics on time lost, grade work, normalized braking and acceleration index for the trip helped the drivers to obtain energy savings	5.5%
[PBEM21]	Eco-driving training	Investigate the factors that can reinforce the benefits from an eco-driving theory course in the freight segment	Field trials (P=14)	To strengthen the benefits of an eco-driving course in long term, follow-up evaluations through performance reports and non-monetary rewards were found to be significant	7.5-9%
[MNR+21]	Eco-driving heuristics	Investigate on how modifying the driving style using heuristics could lead to a reduction in the energy consumption	Simulation	Modify the representative driving cycles into eco-driving speed profiles by limiting maximum travel speed, intensity of acceleration and braking at high speeds and encouraging coasting behavior	6%
[GICC21]	On-board driving assistance device	Monitor the fuel savings in waste-collection fleet by implementing a real-time eco-driving feedback to the drivers	Field trials (P=67)	Assisting drivers to minimize excessive idling, over-revving, harsh braking and over-speeding in real-time using visual green/red lights and auditory signals	7.45%

P = Number of participants.

Table A.2: BEV parameters

Symbol	Description	Value	Unit
$m + m_\mathrm{r}$	Total mass of the vehicle	1800	kg
A_v	Vehicle cross-section	2.27	m^2
c_d	Air drag coefficient	0.29	-
c_r	Rolling friction coefficient	0.011	-
ρ_a	Air density	1.202	$\mathrm{kg/m}^3$

Table A.3: City bus equipped with conventional ICE parameters

Symbol	Description	Value	Unit
$m + m_\mathrm{r}$	Total mass of the vehicle	19000	kg
A_v	Vehicle cross-section	7.65	m^2
c_d	Air drag coefficient	0.7	-
c_r	Rolling friction coefficient	0.012	-
ρ_a	Air density	1.202	$\mathrm{kg/m}^3$

Bibliography

[ACE22] ACEA. Report - vehicles in use, europe 2022, Last call on 10 December 2022. https://www.acea.auto/files/ACEA-report-vehicles-in-use-europe-2022.pdf.

[ACNW17] K. Ayyildiz, F. Cavallaro, S. Nocera, and R. Willenbrock. Reducing fuel consumption and carbon emissions through eco-drive training. *Transportation Research Part F: Traffic Psychology and Behaviour*, 46:96–110, 2017.

[Ade09] E. Adell. Acceptance of driver support systems. In *Proceedings of the European Conference on Human Centred Design for Intelligent Transport Systems*, pages 475–486, 2009.

[AEE15] A. F. Acosta, J. E. Espinosa, and J. Espinosa. Traci4matlab: Enabling the integration of the SUMO road traffic simulator and Matlab through a software re-engineering process. In *Modeling Mobility with Open Data*, pages 155–170. Springer, 2015.

[Ajz91] I. Ajzen. The theory of planned behavior. *Organizational Behavior and Human Decision Processes*, 50:179–211, 1991.

[AM14] M. S. Alam and A. McNabola. A critical review and assessment of eco-driving policy & technology: benefits & limitations. *Transport Policy*, 35:42–49, 2014.

[AM18] M. S. Alam and A. McNabola. Network-wide traffic and environmental impacts of acceleration and deceleration among eco-driving vehicles in different road configurations. *Transportation Planning and Technology*, 41:244–264, 2018.

[AV11] B. Asadi and A. Vahidi. Predictive cruise control: Utilizing upcoming traffic signal information for improving fuel economy and reducing trip time. *IEEE Transactions on Control Systems Technology*, 19(3):707–714, 2011.

[Bar10] J. N. Barkenbus. Eco-driving: an overlooked climate change initiative. *Energy Policy*, 38(2):762–769, 2010.

[BBD+09] B. Beusen, S. Broekx, T. Denys, C. Beckx, B. Degraeuwe, M. Gijsbers, K. Scheepers, L. Govaerts, R. Torfs, and L. I. Panis. Using on-board logging devices to study the longer-term impact of an eco-driving course. *Transportation Research Part D*, 14:514–520, 2009.

[BBM12] A. Beghi, M. Bruschetta, and F. Maran. A real time implementation of MPC based Motion Cueing strategy for driving simulators. In *Proceedings of the 51st IEEE Conference on Decision and Control (CDC)*, pages 6340–6345, 2012.

[BCV+17] S. Beloufa, F. Cauchard, J. Vedrenne, B. Vailleau, A. Kemeny, F. Mérienne, and J. M. Boucheix. Learning eco-driving behaviour in a driving simulator: Contribution of instructional videos and interactive guidance system. *Transportation Research Part F: Traffic Psychology and Behaviour*, 61:201–216, 2017.

[Ber12] D. P. Bertsekas. *Dynamic Programming and Optimal Control: Volume I*. Athena scientific, 2012.

[BG20] B. Boehmke and B. M. Greenwell. *Hands-On Machine Learning with R*. CRC Press: Chapman & Hall/CRC The R Series, 2020.

[BKK18] S. Bai, J. Z. Kolter, and V. Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv:1803.01271*, 2018.

[Bro18] J. Brownlee. *Deep Learning for Time Series Forecasting*. Machine Learning Mastery, 1st edition, 2018.

[BSL+18] P. Batta, M. Singh, Z. Li, Q. Ding, and L. Trajkovic. Evaluation of Support Vector Machine Kernels for Detecting Network Anomalies. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–4, 2018.

[CGCB14] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Proceedings of the NIPS Workshop on Deep Learning*, 2014.

[CGE+22] S. K. Chada, D. Görges, A. Ebert, R. Teutsch, and C. G. Min. Learning-based driver behavior modeling and delay compensation to improve the efficiency of an eco-driving assistance system. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 415–422, 2022.

[CGE+23] S. K. Chada, D. Görges, A. Ebert, R. Teutsch, and S. P. Subramanya. Evaluation of the driving performance and user acceptance of a predictive eco-driving assistance system for electric vehicles. *Transportation Research Part C: Emerging Technologies*, 153:104193, 2023.

[CGET21] S. K. Chada, D. Görges, A. Ebert, and R. Teutsch. A driver-in-the-loop co-simulation framework for testing predictive EDAS for commercial vehicles in urban environments. In *Proceedings of the 6th International Commercial Vehicle Technology Symposium*, pages 107–118, 2021.

[CGET23] S. K. Chada, D. Görges, A. Ebert, and R. Teutsch. Deep learning-based vehicle speed prediction for ecological adaptive cruise control in urban and highway scenarios. In *Proceedings of the 22nd IFAC World Congress*, pages 1107–1114, 2023.

[CKPW20] Z. Cui, R. Ke, Z. Pu, and Y. Wang. Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values. *Transportation Research Part C: Emerging Technologies*, 118:102674, 2020.

[CKR+22] S. K. Chada, M. Kunz, Y. Ranker, R. Teutsch, D. Görges, A. Ebert, and K. Mahjoub. Kompakter 6-dof-fahrsimulator für flexible forschungsanwendungen. In *Proceedings of the 7th Commercial Vehicle Technology Symposium*, pages 15–35, 2022.

[CL22] Y. Chen and M. Lazar. Driving mode advice for eco-driving assistance system with driver reaction delay compensation. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 69(1):134–138, 2022.

[CMV10] Z. Constantinescu, C. Marinoiu, and M. Vladoiu. Driving style analysis using data mining techniques. *International Journal of Computers, Communications and Control*, 5:654–663, 2010.

[CPG+20] S. K. Chada, A. Purbai, D. Görges, A. Ebert, and R. Teutsch. Ecological adaptive cruise control for urban environments using SPaT information. In *Proceedings of the 2020 IEEE Vehicle Power and Propulsion Conference (VPPC)*, pages 1–6, 2020.

[CTG+21] S. K. Chada, J. M. Thomas, D. Görges, A. Ebert, and R. Teutsch. Ecological adaptive cruise control for city buses based on hybrid model predictive control using PnG and traffic light information. In *Proceedings of the IEEE Vehicle Power and Propulsion Conference (VPPC)*, pages 1–7, 2021.

[Dav85] F. D. Davis. A technology acceptance model for empirically testing new end-user information systems: theory and results. *PhD Dissertation*, 1985.

[DBF+13] T. J. Daun, D. G. Braun, C. Frank, S. Haug, and M. Lienkamp. Evaluation of driving behavior and the efficacy of a predictive eco-driving assistance system for heavy commercial vehicles in a driving simulator experiment. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 2379–2386, 2013.

[DBR+] S. Durekovic, A. Bracht, B. Raichle, M. Rauch, J. Requejo, D. Toropov, and A. Varchmin. ADASIS v2 Protocol. `https://adasis.org/wp-content/uploads/sites/10/2024/08/200v2.0.1-D2.2-ADASIS-v2-Specification.0.pdf` [Last call on 16th January 2025].

[DBW89] F. D. Davis, R. P. Bagozzi, and P. R. Warshaw. User acceptance of computer technology: A comparison of two theoretical models. *Management Science*, 35:982–1003, 1989.

[Do20] T-N. Do. Automatic Learning Algorithms for Local Support Vector Machines. *SN Computer Science*, 1:2, 2020.

[EEG13] P. Elbert, S. Ebbesen, and L. Guzzella. Implementation of dynamic programming for n-dimensional optimal control problems with final state constraints. *IEEE Transactions on Control Systems Technology*, 21(3):924–931, 2013.

[EU 23] EU Regulation. Regulation (EU) 2019/631 of the european parliament and of the council of 17 april 2019 setting CO2 emission performance standards for new passenger cars and for new light commercial vehicles, and repealing regulations (EC) no 443/2009 and (EU) no 510/2011, Last call on 1 April 2023. https://climate.ec.europa.eu/eu-action/transport-emissions/road-transport-reducing-co2-emissions-vehicles/co2-emission-performance-standards-cars-and-vans.

[FA75] M. Fishbein and I. Ajzen. *Belief, Attitude, Intention, and Behavior: An Introduction to Theory and Research.* Reading, MA: Addison-Wesley, 1975.

[FKA15] C. Fors, K. Kircher, and C. Ahlström. Interface design of eco-driving support systems - truck drivers' preferences and behavioural compliance. *Transportation Research Part C: Emerging Technologies*, 58:706–720, 2015.

[FL81] C. Fornell and D. F. Larcker. Evaluating structural equation models with unobservable variables and measurement error. *Journal of Marketing Research*, 18:39–50, 1981.

[For] Ford. EcoMode: Helping you drive more efficiently. `https://www.ford.ie/shop/explore/technology/performance-and-efficiency/ecomode` [Last call on 20th December 2024].

[GA99] L. Guzzella and A. Amstutz. CAE tools for quasi-static modeling and optimization of hybrid powertrains. *IEEE Transactions on Vehicular Technology*, 48:1762–1769, 1999.

[GAL+19] T. D. Gaikwad, Z. D. Asher, K. Liu, M. Huang, and I. Kolmanovsky. Vehicle velocity prediction and energy management strategy part 2: Integration of machine learning vehicle velocity prediction with optimal energy management to improve fuel economy. *SAE Technical Paper*, 2019-01-1212, 2019.

[GICC21] J. F. Gonzalez, A. G. Izquierdo, F. Commans, and M. Carlos. Fuel-efficient driving in the context of urban waste-collection : A Spanish case study. *Journal of Cleaner Production*, 289:125831, 2021.

[GP21] T. Günthner and H. Proff. On the way to autonomous driving: How age influences the acceptance of driver assistance systems. *Transportation Research Part F: Traffic Psychology and Behaviour*, 81:586–607, 2021.

[GWM+21] X. Gong, J. Wang, B. Ma, L. Lu, Y. Hu, and H. Chen. Real-time integrated power and thermal management of connected HEVs based on hierarchical model predictive control. *IEEE/ASME Transactions on Mechatronics*, 26:1271–1282, 2021.

[HAK+22] Q. Hu, M. R. Amini, I. Kolmanovsky, J. Sun, A. Wiese, and J. B. Seeds. Multihorizon Model Predictive Control: An application to integrated power and thermal management of connected hybrid electric vehicles. *IEEE Transactions on Control Systems Technology*, 30:1052–1064, 2022.

[HBBD15] M. Henzler, A. Boller, M. Buchholz, and K. Dietmeyer. Acceptance of driver support systems. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pages 2007–2012, 2015.

[HBMC04] P. R. Hinton, C. Brownlow, I. McMurray, and B. Cozens. *SPSS Explained*. Routledge, 1st edition, 2004.

[HCTK12] D. Hrovat, S. Di Cairano, H.E. Tseng, and I.V. Kolmanovsky. The development of Model Predictive Control in automotive industry: A survey. In *Proceedings of the IEEE International Conference on Control Applications*, pages 295–302, 2012.

[HHRS14] J. F. Hair, G. T. Hult, C. M. Ringle, and M. Sarstedt. *A Primer on Partial Least Squares Structural Equation Modeling (PLS-SEM)*. Thousand Oaks, CA: Sage, 1st edition, 2014.

[HIÅN09] E. Hellström, M. Ivarsson, J. Åslund, and L. Nielsen. Look-ahead control for heavy trucks to minimize trip time and fuel consumption. *Control Engineering Practice*, 7(2):245–254, 2009.

[HLW+24] J. Hong, X. Luo, H. Wu, X. Na, H. Chu, B. Gao, and H. Chen. Energy-Saving Driving Assistance System Integrated With Predictive Cruise Control for Electric Vehicles. *IEEE Transactions on Intelligent Vehicles*, 9:4518–4528, 2024.

[HNZ+18] 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*, 93:596–609, 2018.

[Hon] Honda Tech Info. Driver information interface: Honda Civic. `https://techinfo.honda.com/rjanisis/pubs/om/ah/at202222iom/enu/details/131229047-291341.html` [Last call on 21 December 2023].

[HVS19] J. Han, A. Vahidi, and A. Sciarretta. Fundamentals of energy efficient

driving for combustion engine and electric vehicles: An optimal control perspective. *Automatica*, pages 558–572, 2019.

[IBM21] IBM Corp. IBM SPSS Statistics for Windows. Version 28.0., Released 2021. `https://www.ibm.com/spss`.

[Igb93] M. Igbaria. User acceptance of microcomputer technology: An empirical test. *Omega, The International Journal of Management Science*, 21:73–90, 1993.

[JCG20] Y. Jia, C. Cai, and D. Görges. An LSTM-based speed predictor based on traffic simulation data for improving the performance of energy-optimal adaptive cruise control. In *Proceedings of the 23rd IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–7, 2020.

[JHJ15] S. L. Jamson, D. L. Hibberd, and A. H. Jamson. Drivers ability to learn eco-driving skills; effects on fuel efficient and safe driving behaviour. *Transportation Research Part C: Emerging Technologies*, 58:657–668, 2015.

[JJG20] Y. Jia, R. Jibrin, and D. Görges. Energy-optimal adaptive cruise control for electric vehicles based on linear and nonlinear model predictive control. *IEEE Transactions on Vehicular Technology*, 69(12):14173–14187, 2020.

[JJIG19] Y. Jia, R. Jibrin, Y. Itoh, and D. Görges. Energy-optimal adaptive cruise control for electric vehicles in both time and space domain based on model predictive control. In *Proceedings of the 9th IFAC International Symposium on Advances in Automotive Control*, pages 13–20, 2019.

[KFA14] K. Kircher, C. Fors, and C. Ahlström. Continuous versus intermittent presentation of visual eco-driving advice. *Transportation Research Part F: Traffic Psychology and Behaviour*, 24:27–38, 2014.

[KMMK13] M. A. S. Kamal, M. Mukai, J. Murata, and T. Kawabe. Model Predictive Control of Vehicles on Urban Roads for Improved Fuel Economy. *IEEE Transactions on Control Systems Technology*, 21:831–841, 2013.

[KPKP21] M. Krznar, P. Piljek, D. Kotarski, and D. Pavkovic. Modeling, Control System Design and Preliminary Experimental Verification of a Hybrid Power Unit Suitable for Multirotor UAVs. *Energies*, 14:2669, 2021.

[LAG+19] K. Liu, Z. Asher, X. Gong, M. Huang, and I. Kolmanovsky. Vehicle velocity prediction and energy management strategy part 1: Deterministic and stochastic vehicle velocity prediction using machine learning. *SAE Technical Paper*, 2019-01-1051, 2019.

[LBBw+18] P. A. Lopez, M. Behrisch, L. Bieker-walz, J. Erdmann, Y. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wiessner. Microscopic traffic simulation using SUMO. In *Proceedings of the 21st IEEE In-*

ternational Conference on Intelligent Transportation Systems, pages 2575–2582, 2018.

[LDX+19] N. Lyu, C. Deng, L. Xie, C. Wu, and Z. Duan. A field operational test in China: Exploring the effect of an advanced driver assistance system on driving performance and braking behavior. *Transportation Research Part F: Traffic Psychology and Behaviour*, 65:730–747, 2019.

[LGL14] X. Lin, D. Görges, and S. Liu. Eco-driving assistance system for electric vehicles based on speed profile optimization. In *Proceedings of the IEEE Conference on Control Applications (CCA)*, pages 629–634, 2014.

[LGW17] X. Lin, D. Görges, and A. Weißmann. Simplified energy-efficient adaptive cruise control based on model predictive control. In *Proceedings of the 20th IFAC World Congress*, volume 50, pages 4794–4799, 2017.

[LKSM20] F. Leach, G. Kalghatgi, R. Stone, and P. Miles. The scope for improving the efficiency and environmental impact of internal combustion engines. *Transportation Engineering*, 1:100005, 2020.

[LLRW09] S. Li, K. Li, R. Rajamani, and J. Wang. Multi-objective coordinated control for advanced adaptive cruise control system. In *Proceedings of the 48th IEEE Conference on Decision and Control (CDC)*, pages 3539–3544, 2009.

[LSBL14] S. Lefèvre, C. Sun, R. Bajcsy, and C. Laugier. Comparison of parametric and non-parametric approaches for vehicle speed prediction. In *Proceedings of the American Control Conference*, pages 3494–3499, 2014.

[LVL14] S. Lefèvre, D. Vasquez, and C. Laugier. A survey on motion prediction and risk assessment for intelligent vehicles. *ROBOMECH Journal*, 1:1, 2014.

[LVR+20] X. Li, A. Vaezipour, A. Rakotonirainy, S. Demmel, and O. Oviedo-Trespalacios. Exploring drivers' mental workload and visual demand while using an in-vehicle HMI for eco-safe driving. *Accident Analysis & Prevention*, 146:105756, 2020.

[Mar98] G. M. Maruyama. *Basics of structural equation modeling*. Thousand Oaks, CA: Sage, 1st edition, 1998.

[MGC+16] D. Maamria, K. Gillet, G. Colin, Y. Chamaillard, and C. Nouillant. On the use of dynamic programming in eco-driving cycle computation for electric vehicles. In *Proceedings of the IEEE Conference on Control Applications (CCA)*, pages 1288–1293, 2016.

[MGP88] M. Morari, C. E. Garcia, and D. M. Prett. Model predictive control: Theory and practice. *IFAC Proceedings Volumes*, 21:1–12, 1988.

[MLZC18] C. Miao, H. Liu, G. G. Zhu, and H. Chen. Connectivity-based optimization of vehicle route and speed for improved fuel economy. *Transportation*

Research Part C: Emerging Technologies, 91:353–368, 2018.

[MNR+21] M. Miotti, Z. A. Needell, S. Ramakrishnan, J. Heywood, and J. E. Trancik. Quantifying the impact of driving style changes on light-duty vehicle fuel consumption. *Transportation Research Part D*, 98:102918, 2021.

[Mot] MotionSystems. Software: full range of software for building amazing simulators, cutting edge cinemas and reliable environment for equipment testing. `https://motionsystems.eu/software/` [Last call on 11th January 2025].

[MSGW17] R. C. McIlroy, N. A. Stanton, L. Godwin, and A. P. Wood. Encouraging eco-driving with visual, auditory, and vibrotactile stimuli. *IEEE Transactions on Human-Machine Systems*, 47:661–672, 2017.

[MT16] C. Miyajima and K. Takeda. Driver-behavior modeling using on-road driving data. *IEEE Signal Processing Magazine*, 33:14–21, 2016.

[MWS+15] D. Moser, H. Waschl, R. Schmied, H. Efendic, and L. Del Re. Short term prediction of a vehicle's velocity trajectory using ITS. *SAE International Journal of Passenger Cars - Electronic and Electrical Systems*, 8(2):364–370, 2015.

[NFC+15] I. Neumann, T. Franke, P. Cocron, F. Bühler, and J. F. Krems. Eco-driving strategies in battery electric vehicle use - how do drivers adapt over time? *IET Intelligent Transport Systems*, 9(7):746–753, 2015.

[NHSS10] D. V. Ngo, T. Hofman, M. Steinbuch, and A. F. A. Serrarens. An optimal control-based algorithm for hybrid electric vehicle using preview route information. In *Proceedings of the American Control Conference*, pages 5818–5823, 2010.

[Nisa] Nissan Leaf. Nissan LEAF benutzerhandbücher: Modus ECO. `https://www.nissan.de/kunden/benutzerhandb%C3%BCcher/benutzerhandb%C3%BCcher/iom/leaf/0ze1/e0/2023/modus-eco-2.shtml` [Last call on 21th December 2024].

[Nisb] Nissan Motor Corporation. Nissan unveils world's first ECO pedal. `https://global.nissannews.com/en/videos/video-73532195001` [Last call on 20th December 2024].

[NPB+14] J. Neubauer, A. Pesaran, C. Bae, R. Elder, and B. Cunningham. Updating united states advanced battery consortium and department of energy battery technology targets for battery electric vehicles. *Journal of Power Sources*, 271:614–621, 2014.

[NSLF19] K. Nikzadfar, N. Bakhshinezhad S. A. Sadeghi, H. T. Ledari, and A. Fathi. An Optimal Gear Shifting Strategy for Minimizing Fuel Consumption Based on Engine Optimum Operation Line. *SAE Technical Paper*, 2019-01-5055,

2019.

[OBL+19] T. O'Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, L. Invernizzi, et al. Keras Bayesian Optimization Tuner. `https://github.com/keras-team/keras-tuner`, 2019.

[OOF17] E. Ozatay, U. Ozguner, and D. Filev. Velocity profile optimization of on road vehicles: Pontryagin's maximum principle based approach. *Control Engineering Practice*, 61:244–254, 2017.

[Ope17] OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org . `https://www.openstreetmap.org`, 2017.

[PBEM21] D. R. Pinchasik, I. Beate, B. Eirill, and C. S. Mjosund. Can active follow-ups and carrots make eco-driving stick? Findings from a controlled experiment among truck drivers in Norway. *Energy Research & Social Science*, 75:102007, 2021.

[PFN12] J. C. G. Palencia, T. Furubayashi, and T. Nakata. Energy use and CO2 emissions reduction potential in passenger car fleet using zero emission vehicles and lightweight materials. *Energy*, 48:548–565, 2012.

[PWD18] G. P. Padilla, S. Weiland, and M. C. F. Donkers. A Global Optimal Solution to the Eco-Driving Problem. *IEEE Control Systems Letters*, 2:599–604, 2018.

[QWW+18] X. Qi, P. Wang, G. Wu, K. Boriboonsomsin, and M. J. Barth. Connected cooperative ecodriving system considering human driver error. *IEEE Transactions on Intelligent Transportation Systems*, 19(8):2721–2733, 2018.

[RAG+10] L. Del Re, F. Allgöwer, L. Glielmo, C. Guardiola, I. Kolmanovsky, M. Morari, and M. Thoma. *Automotive Model Predictive Control: Models, Methods and Applications*. Springer London, 1st edition, 2010.

[RAG+21] A. Rabinowitz, F. M. Araghi, T. Gaikwad, Z. D. Asher, and T. H. Bradley. Development and evaluation of velocity predictive optimal energy management strategies in intelligent and connected hybrid electric vehicles. *Energies*, 14:5713, 2021.

[RBC+10] G. Ripaccioli, D. Bernardini, S. D. Cairano, A. Bemporad, and I.V. Kolmanovsky. A stochastic model predictive control approach for series hybrid electric vehicle power management. In *Proceedings of the American Control Conference (ACC)*, pages 5844–5849, 2010.

[REKR08] C. Ress, A. Etemad, D. Kuck, and J. Requejo. Electronic Horizon - Providing Digital Map Data for ADAS Applications. In *Proceedings of the 2nd International Workshop on Intelligent Vehicle Control Systems*, pages 40–49, 2008.

[Ren21] Renub Research. Global automotive market, growth and forecast, impact of coronavirus, industry trends, by region, opportunity company analysis. *Report*, pages 1–250, 2021.

[RLHS17] M. M. Rahman, M. F. Lesch, W. J. Horrey, and L. Strawderman. Assessing the utility of TAM, TPB, and UTAUT for advanced driver assistance systems. *Accident Analysis and Prevention*, 108:361–373, 2017.

[RSM20] E. S. Rahman, M. Subhan, and H. Maulana. Convergence analysis of euler-maruyama method on ornstein-uhlenbeck equation. *Journal of Physics: Conference Series*, 1554:012004, 2020.

[SABA21] M. Schwenzer, M. Ay, T. Bergs, and D. Abel. Review on model predictive control: an engineering perspective. *The International Journal of Advanced Manufacturing Technology*, 117:1327–1349, 2021.

[SCG22] G. Sundaram, S. K. Chada, and D. Görges. A novel approach to classify and replicate human drivers using model predictive control. In *Proceedings of the 7th Commercial Vehicle Technology Symposium*, pages 36–51, 2022.

[SDN15] M. J. M. Sullman, L. Dorn, and P. Niemi. Eco-driving training of professional bus drivers - does it work? *Transportation Research Part C: Emerging Technologies*, 58:749–759, 2015.

[SHMS15] C. Sun, X. Hu, S. J. Moura, and F. Sun. Velocity predictors for predictive energy management in hybrid electric vehicles. *IEEE Transactions on Control Systems Technology*, 23(3):1197–1204, 2015.

[SKH22] G. S. Sankar, M. Kim, and K. Han. Data-driven leading vehicle speed forecast and its application to ecological predictive cruise control. *IEEE Transactions on Vehicular Technology*, 71(11):11504–11514, 2022.

[SKSH19] J. Shin, S. Kim, M. Sunwoo, and M. Han. Ego-vehicle speed prediction using fuzzy markov chain with speed constraints. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 2106–2112, 2019.

[SKT+20] L. Shao, A. E. H. Karci, D. Tavernini, A. Sorniotti, and M. Cheng. Design approaches and control strategies for energy-efficient electric machines for electric vehicles - a review. *IEEE Access*, 8:116900–116913, 2020.

[SNO15] A. Scirretta, G. D. Nunzio, and L. L. Ojeda. Optimal Ecodriving Control: Energy-Efficient Driving of Road Vehicles as an Optimal Control Problem. *IEEE Control Systems*, 35:71–90, 2015.

[SPP15] J. Son, M. Park, and B. B. Park. The effect of age, gender and roadway environment on the acceptance and effectiveness of advanced driver assistance systems. *Transportation Research Part F: Traffic Psychology and Behaviour.*, 31:12–24, 2015.

[SRDV13] B. Saerens, H. A. Rakha, M. Diehl, and E. Van den Bulck. A methodology for assessing eco-cruise control for passenger vehicles. *Transportation Research Part D: Transport and Environment*, 19:20–27, 2013.

[SSKK14] M. Staubach, N. Schebitz, F. Köster, and D. Kuck. Evaluation of an eco-driving support system. *Transportation Research Part F: Traffic Psychology and Behaviour*, 27:11–21, 2014.

[SWD15] R. Schmied, H. Waschl, and L. Del Re. A simplified fuel efficient predictive cruise control approach. *SAE Technical Paper*, 2015-01-0296, 2015.

[SZ21] Y. Shi and K. Zhang. Advanced model predictive control framework for autonomous intelligent mechatronic systems: A tutorial overview and perspectives. *Annual Reviews in Control*, 52:170–196, 2021.

[Tho12] S. C. E. Thomas. How green are electric vehicles? *International Journal of Hydrogen Energy*, 37:6053–6062, 2012.

[TNS18] L. Thibault, G. D. Nunzio, and A. Sciarretta. A Unified Approach for Electric Vehicles Range Maximization via Eco-Routing, Eco-Driving, and Energy Consumption Prediction. *IEEE Transactions on Intelligent Vehicles*, 3:463–475, 2018.

[Toy] Toyota. Toyota to introduce Eco drive indicator: New feature aims to encourage environmentally considerate driving. `https://global.toyota/en/detail/272098` [Last call on 20th December 2024].

[UB12] J. B. Ullman and P. M. Bentler. Structural equation modeling. In I. B. Weiner, editor, *Handbook of Psychology*, pages 661–690. John Wiley & Sons, 2012.

[VBH+13] C. Vagg, C. J. Brace, D. Hari, S. Akehurst, J. Poxon, and L. Ash. Development and field trial of a driver assistance system to encourage eco-driving in light commercial vehicle fleets. *IEEE Transactions on Intelligent Transportation Systems*, 14:796–805, 2013.

[VHD97] J. D. Van Der Laan, A. Heino, and D. De Waard. A simple procedure for the assessment of acceptance of advanced transport telematics. *Transportation Research Part C: Emerging Technologies*, 5(1):1–10, 1997.

[VLF+14] S. Vazquez, J. I. Leon, L. G. Franquelo, J. Rodriguez, H. A. Young, A. Marquez, and P. Zanchetta. Model Predictive Control: A Review of Its Applications in Power Electronics. *IEEE Industrial Electronics Magazine*, 8:16–31, 2014.

[VMDD03] V. Venkatesh, M. G. Morris, G. B. Davis, and F. D. Davis. The theory of planned behavior. *MIS Quarterly*, 27:425–478, 2003.

[Vola] Volkswagen. Die ID. Modelle im Winter: Neun Tipps

fuer mehr Reichweite. https://www.volkswagen.ch/de/e-mobilitaet-und-e-autos/laden-und-reichweite/die-id-modelle-im-winter-neun-tipps-fuer-mehr-reichweite.html [Last call on 21th December 2024].

[Volb] Volvo. Here's how your Volvo saves fuel in Eco drive mode. https://techinfo.honda.com/rjanisis/pubs/om/ah/at202222iom/enu/details/131229047-291341.html [Last call on 20th December 2024].

[VPD+20] G.-D. Voinea, C. C. Postelnicu, M. Duguleana, G.-L. Mogan, and R. Socianu. Driving performance and technology acceptance evaluation in real traffic of a smartphone-based driver assistance system. *International Journal of Environmental Research and Public Health*, 17(19):7098, 2020.

[VTX12] V. Venkatesh, J. Y. L. Thong, and X. Xu. Consumer acceptance and use of information technology: Extending the unified theory of acceptance and use of technology. *MIS Quarterly*, 36:157–178, 2012.

[WAG+19] Z. Wang, P. Amar, E. Garmon, S. Tanugula, Y. Hsu, A. Vu, F. Caballero, P. Hao, G. Wu, K. Boriboonsomsin, M. J. Barth, and A. Kailas. Early Findings from Field Trials of Heavy-Duty Truck Connected Eco-Driving System. In *Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 3037–3042, 2019.

[WGL18] A. Weißmann, D. Görges, and X. Lin. Energy-optimal adaptive cruise control combining model predictive control and dynamic programming. *Control Engineering Practice*, 72:125–137, 2018.

[WHK+23] M. Wegener, F. Herrmann, L. Koch, R. Savelsberg, and J. Andert. Longitudinal vehicle motion prediction in urban settings with traffic light interaction. *IEEE Transactions on Intelligent Vehicles*, 8(1):204–215, 2023.

[WLW+20] Z. Wang, X. Liao, C. Wang, D. Oswald, G. Wu, K. Boriboonsomsin, M. J. Barth, K. Han, B. Kim, and P. Tiwari. Driver behavior modeling using game engine and real vehicle: A learning-based approach. *IEEE Transactions on Intelligent Vehicles*, 5(4):738–749, 2020.

[WYBD25] F. Wu, H. Ye, T. Bektas, and M. Dong. New and tractable formulations for the eco-driving and the eco-routing-and-driving problems. *European Journal of Operational Research*, 321:445–461, 2025.

[ZWRZ15] X. Zhao, Y. Wu, J. Rong, and Y. Zhang. Development of a driving simulator based eco-driving support system. *Transportation Research Part C: Emerging Technologies*, 58:631–641, 2015.

Zusammenfassung

Die Verkehrsbranche steht angesichts des weltweit zunehmenden Fahrzeugbestands vor erheblichen Herausforderungen. Insbesondere Verkehrsüberlastung und Umweltverschmutzung infolge steigender Treibhausgasemissionen stellen zentrale Problembereiche dar. Zur Bewältigung der drängenden Problematik des Klimawandels haben Gesetzgeber strenge Regulierungen zur Reduktion der CO_2-Emissionen im Straßenverkehr eingeführt. Der Übergang zu nachhaltigen Mobilitätslösungen, insbesondere batterieelektrischen Fahrzeugen (BEVs) sowie dem öffentlichen Personennahverkehr, spielt hierbei eine entscheidende Rolle. Obwohl BEVs aufgrund ihrer lokal emissionsfreien Fahrweise als umweltfreundliche Alternative gelten, ist deren Betrieb insbesondere in der Übergangsphase, in der der benötigte Strom teilweise noch aus fossilen Quellen stammt, möglichst energieeffizient zu gestalten. Gleichzeitig ist auch im Bereich des öffentlichen Verkehrs, etwa bei Linienbussen mit modernen Verbrennungsmotoren (ICEs), trotz Einhaltung strenger Emissionsvorgaben eine weitere Senkung des Kraftstoffverbrauchs notwendig, sowohl im Hinblick auf ökologische Nachhaltigkeit als auch zur Reduzierung der Betriebskosten ganzer Fahrzeugflotten. Vor diesem Hintergrund hat sich das sogenannte Eco-Driving als eine vielversprechende und kosteneffiziente Strategie etabliert, um das Fahrverhalten energieeffizient zu gestalten. Diese Fahrweise hat in den letzten Jahren sowohl in der wissenschaftlichen Forschung als auch in der Automobilindustrie zunehmende Aufmerksamkeit erlangt. Zwar haben zahlreiche Automobilhersteller bereits Funktionen zur Förderung effizienter Fahrtechniken implementiert, doch bestehen in bestehenden Eco-Driving-Systemen deutliche Einschränkungen. Diese beruhen meist auf vereinfachten Optimierungsansätzen und geben dem Fahrer lediglich situativ begrenzte Hinweise zur Reduktion von Energie- bzw. Kraftstoffverbrauch.

Im Rahmen dieser Dissertation wird ein neuartiges, prädiktives Eco-Fahrerassistenzsystem (engl.: *Eco-Driving Assistance System* (EDAS)) entwickelt und vorgestellt. Dieses System nutzt Information von On-Board-Sensorik, topografische Streckeninformationen (z. B. Steigung und Straßenkrümmung) sowie Signaldaten der Lichtsignalanlage, und generiert daraus kontinuierliche Geschwindigkeitsvorgaben. Ziel ist es, den Fahrstil unter verschiedensten Fahrsituationen systematisch in Richtung Energieeffizienz zu optimieren und gleichzeitig Fahrkomfort und Verkehrssicherheit zu erhöhen. Damit leistet diese Arbeit einen bedeutenden Beitrag zur Weiterentwicklung Eco-Fahrerassistenzsysteme, indem sie bestehende Forschungslücken systematisch identifiziert und gezielt adressiert.

Kapitel 2: In diesem Kapitel wurde das kontinuierliche, zeitabhängige Optimalsteuerungsproblem (OCP) unter Nebenbedingungen eingeführt und verschiedene numerische

Lösungsverfahren dafür dargestellt. Es wurden insbesondere die Einschränkungen des Pontryagin'schen Minimumprinzips im Hinblick auf die Behandlung von Ungleichheitsnebenbedingungen und dessen eingeschränkte Anwendbarkeit auf zeitdiskrete Systeme erläutert. Darüber hinaus wurde die allgemeine Formulierung der dynamischen Programmierung (engl.: *Dynamic Programming* (DP)) analysiert. Diese Methode zeichnet sich zwar durch ihre Robustheit bei der Berücksichtigung von Nebenbedingungen und der Gewährleistung globaler Optimalität aus, leidet jedoch erheblich unter dem sogenannten Fluch der Dimensionalität, was zu einer ineffizienten numerischen Lösung führt. Im weiteren Verlauf wurde das Konzept der modellprädiktiven Regelung (engl.: *Model Predictive Control* (MPC)) behandelt, wobei insbesondere deren Vorteile hinsichtlich Echtzeitoptimierung, rechentechnischer Effizienz sowie Flexibilität gegenüber Störungen herausgestellt wurden. Ergänzend wurden maschinelle Lernverfahren betrachtet, sowohl unüberwachtes Lernen (engl.: *Unsupervised Learning*) zur Dimensionsreduktion und Clusteranalyse als auch überwachtes Lernen (engl.: *Supervised Learning*) zur Klassifikation und Regression.

Kapitel 3: Dieses Kapitel beschreibt die Architektur des prädiktiven EDAS und dessen zentrale Komponenten, darunter die Vorschau- Information, die Zustandswechsel-Logik, das Eco-Driving Feedback-System sowie das Driver-in-the-Loop-Setup. Das Hauptziel des vorgeschlagenen prädiktiven EDAS besteht darin, den Fahrer durch die Bereitstellung optimaler Geschwindigkeitsvorgaben in unterschied- lichen Fahrsituationen dabei zu unterstützen, ihren Fahrstil energieeffizienter zu gestalten und somit den Gesamtenergieverbrauch des Fahrzeugs zu senken. Darüber hinaus wurde eine auf maschinellem Lernen basierende Architektur zur Modellierung des Fahrverhaltens vorgestellt, die zur Clusterung bzw. Klassifikation von Fahrdaten integriert. Ergänzt wird das System durch eine Methode zur Kompensation von Geschwindigkei- tsverfolgungsfehlern auf Fahrerseite, um die Effizienz des prädiktiven EDAS weiter zu erhöhen.

Kapitel 4: In diesem Kapitel wurden Problemformulierungen für Eco-Driving modellprädiktive Regler (Eco-MPCs) sowohl für batterieelektrische Fahrzeuge (BEVs) als auch für Fahrzeuge mit Verbrennungsmotor (ICE) vorgestellt, die als zentrale Komponenten des prädiktiven EDAS fungieren. Zunächst wurden die längsdynamischen Systemgleichungen und ein entsprechendes Vorhersagemodell entwickelt, welche die mathematische Grundlage für die Formulierung der MPC-Strategien bilden. Daraufhin wurde ein BEV-Antriebsstrangmodell eingeführt sowie eine angenäherte Leistungsaufnahmekennlinie erstellt. Die Problemstellung für einen Reference tracking MPC in typischen urbanen Umgebungen mit Lichtsignalanlagen wurde anschließend detailliert beschrieben. Zudem wurde die Ableitung des MPC-Optimierungsproblems in der Form eines quadratischen Programms (QP) mit linearen Ungleichheitsnebenbedingungen erläutert. Darüber hinaus wurde ein Car-Following MPC formuliert, der sowohl das Fahrverhalten eines vorausfahrenden Fahrzeugs als auch die Interaktion mit Lichtsignalanlagen berücksichtigt, um energieoptimale Geschwindigkeitstrajektorien für das Host-Fahrzeug abzuleiten. Die simulationsbasierte Validierung dieser Regler zeigte nicht nur signifikante Energieeinsparpotenziale für BEVs, sondern auch Verbesserungen hinsichtlich Fahrkomfort und -sicher-

heit. Darüber hinaus wurde die nahtlose Umschaltung zwischen unterschiedlichen Regelstrategien demonstriert. Zur Erweiterung der Anwendbarkeit des prädiktiven EDAS auf alternative Antriebskonzepte wurde ein Eco-HMPC-Ansatz (engl.: *Hybrid Model Predictive Control*) für einen Stadtbus mit konventionellem Verbrennungsmotor entwickelt. Dieser wurde als ein gemischt-ganzzahlig quadratisch beschränktes quadratisches Programm (engl.: *Mixed-integer Quadratically Constrained Quadratic Programming* (MIQCQP)) formuliert, das sowohl die kontinuierliche Leistungsabgabe als auch den diskreten Motorzustand (Ein/Aus) optimiert. Die Implementierung des Eco-HMPC ermöglichte zusätzliche Energieeinsparungen infolge der Umsetzung einer sogenannten *Pulse-and-Glide*-Strategie (PnG), bei der das Fahrzeug wiederholt gleitet, während der Verbrennungsmotor abgeschaltet und der Antriebsstrang entkoppelt wird. Abschließend wurde durch eine Analyse der Ausführungszeiten über verschiedene Vorhersagehorizonte hinweg die Echtzeitfähigkeit der vorgeschlagenen Eco-MPCs bestätigt.

Kapitel 5: Zur Effizienzsteigerung des im vorhergehenden Kapitel vorgestellten Car-Following-MPCs liegt der Fokus dieses Kapitels auf der genauen Vorhersage der zukünftigen Geschwindigkeiten des vorausfahrenden Fahrzeugs. Hierfür wurde ein skalierbarer, allgemein einsetzbarer Ansatz zur zeitreihenbasierten Datenerhebung mittels des mikroskopischen Verkehrssimulationswerkzeugs SUMO entwickelt, der sowohl urbane als auch außerstädtische Umgebungen berücksichtigt. Die Eingangsdaten wurden in zwei Merkmalsgruppen untergliedert. Zur Prädiktion der Geschwindigkeit des vorausfahrenden Fahrzeugs kamen tiefer rekurrenter neuronaler Netzwerke, konkret LSTM (engl.: *Long-Short Term Memory*) und GRU (engl.: *Gated Recurrent Unit*), zum Einsatz und wurde mit physikalischen Modellen (konstante Geschwindigkeit, konstante Beschleunigung) verglichen. Dabei zeigte das LSTM-Modell die höchste Vorhersagegenauigkeit. Zusätzlich konnte durch die Einbindung kritischer Umgebungsinformationen mittels V2V- und V2I-Kommunikation, beispielsweise durch Berücksichtigung mehrerer vorausfahrender Fahrzeuge und zukünftiger SPaT-Informationen (engl.: *Signal Phase and Timing*), die Prädiktionsgüte signifikant verbessert werden. Die anschließende Evaluierung des Car-Following-MPC auf Basis der vorhergesagten Geschwindigkeiten des vorausfahrenden Fahrzeugs zeigte ein wachsendes Energieeinsparpotenzial mit zunehmender Länge des Vorhersagehorizonts. Durch die Verwendung des LSTM-basierten Car-Following-MPC im Vergleich zum konstanten Geschwindigkeitsmodell ein zusätzlicher durchschnittlicher Energiegewinn von bis zu 2,5% erzielt werden.

Kapitel 6: In diesem Kapitel wurde die objektive Evaluierung des prädiktiven EDAS im Rahmen von Driver-in-the-Loop-Versuchen dargestellt. Zur Analyse der Systemleistung wurden experimentelle Studien mit Probandinnen und Probanden auf einem dynamischen Fahrsimulator durchgeführt. In Evaluation I absolvierten 34 Teilnehmende Fahrten mit und ohne Unterstützung des prädiktiven EDAS in einer virtuellen Umgebung, die sowohl urbane als auch autobahnähnlichen Streckenabschnitte umfasste. Die Simulation bildete realitätsnahe Verkehrsbedingungen ab, einschließlich deterministischer Lichtsignalanlagen und Stoppschilder. Das prädiktive EDAS lieferte kontinuierliche Geschwindigkeitsempfehlungen, um den Fahrer bei der Einhaltung energieopti-

maler Fahrweisen zu unterstützen, etwa durch dynamisch optimierte Geschwindigkeitsempfehlungen auf Autobahnen, grünphasenoptimierte Ampeldurchfahrten, energieeffizientes Folgen eines vorausfahrenden Fahrzeugs sowie sichere Geschwindigkeitshaltung in Kurven. Die Auswertung der Ergebnisse zeigte, dass mithilfe des EDAS Energieeinsparungen von bis zu 10% erzielt erzielt werden konnten. Zudem nahmen Geschwindigkeitsüberschreitungen sowie unnötige Stopps signifikant ab. Des Weiteren konnte ein stabiler und nahtloser Übergang zwischen den eingesetzten MPC-Regelstrategien nachgewiesen werden. In Evaluation II wurde das historische Fahrverhalten der Teilnehmenden analysiert, um deren Fahrstil auf Basis des Geschwindigkeitstrackings zu klassifizieren. Hierzu wurde ein lernbasierter Ansatz zur Modellierung des Fahrverhaltens entwickelt. Dabei kamen unüberwachte Lernverfahren wie die Hauptkomponentenanalyse (engl.: *Principal Component Analysis* (PCA)) und hierarchisches Clustering (engl.: *Hierarchical Clustering Analysis* (HCA)) zur Segmentierung der Fahrdaten zum Einsatz. Für die anschließende Klassifikation wurde ein überwachter Deep-Learning-Verfahren, das *Temporal Convolutional Network* (TCN), verwendet. Zur Vorhersage individueller Fahrfehler beim Geschwindigkeitstracking wurden stochastische Volatilitätsmodelle (engl.: *Stochastic Volatility* (SV)) und deterministische GRU-Modelle miteinander verglichen, wobei die GRU-Modelle eine höhere Vorhersagegenauigkeit aufwiesen. Probanden, die das EDAS mit integrierter Fahrfehlerkompensation nutzten, zeigten eine verbesserte Einhaltung der empfohlenen Sollgeschwindigkeit und erzielten zusätzliche Energieeinsparungen, wodurch die Gesamteffizienz des Systems weiter gesteigert werden konnte.

Kapitel 7: Dieses Kapitel widmet sich der subjektiven Bewertung des prädiktiven EDAS für batterieelektrische Fahrzeuge. Zu diesem Zweck wurde ein strukturierter Fragebogen an insgesamt 41 Probandinnen und Probanden ausgegeben, um deren Einstellungen, subjektive Wahrnehmungen sowie Nutzungsintentionen im Kontext des prädiktiven EDAS systematisch zu erfassen. Die zugrunde liegende Hypothesenbildung erfolgte basierend auf dem Technologieakzeptanzmodell (engl.: *Technology Acceptance Model* (TAM)) sowie der Theorie des geplanten Verhaltens (engl.: *Theory of Planned Behavior* (TPB)). Der Fragebogen umfasste Items zu den Konstrukten der wahrgenommenen Nützlichkeit (engl.: *Perceived Usefulness*), der wahrgenommenen Benutzerfreundlichkeit (engl.: *Perceived Ease of Use*), die Intention zur Nutzung (engl.: *Behavioral Intention*), der subjektiven Normen (engl.: *Subjective Norms*), der wahrgenommenen Verhaltenskontrolle (engl.: *Perceived Behavioral Control*) sowie zur persönlichen Einstellung gegenüber der Nutzung des Systems (engl.: *Attitude Toward the Behavior*). Zur Analyse der Zusammenhänge zwischen diesen latenten Konstrukten wurde die Methode der Strukturgleichungsmodellierung angewandt. Die Ergebnisse belegen, dass insbesondere die wahrgenommene Nützlichkeit sowie die wahrgenommene Verhaltenskontrolle signifikante Prädiktoren für die Verhaltensintention darstellen und somit entscheidende Einflussfaktoren für die Nutzerakzeptanz des prädiktiven EDAS sind.

Kapitel 8: Dieses abschließende Kapitel fasst die zentralen Erkenntnisse der vorliegenden Arbeit zusammen und gibt einen Ausblick auf potenzielle zukünftige Forschungsrichtungen.

Curriculum Vitae

Education

10/2016 – 01/2019	**Rheinland-Pfälzische Technische Universität (RPTU) Kaiserslautern - Landau**	
	Subject:	Commercial Vehicle Technology
	Master Thesis:	Analysis of the Downsizing Potential of 48V Hybrid Powertrains considering Component and Driveability Requirements
	Degree:	Master of Science (M. Sc.)
09/2011 – 04/2015	**SRM University, Chennai, India**	
	Subject:	Automobile Engineering
	Degree:	Bachelor of Technology (B. Tech.)
09/2009 – 06/2011	**Narayana IIT Academy, India**	
	Subject:	Maths, Physics and Chemistry
	Degree:	Board of Intermediate Education

Professional Experience

02/2024 – till date	**RPTU Kaiserslautern-Landau**
01/2019 – 09/2023	Wissenschaftlicher Mitarbeiter (Research Associate) Institute for Electromobility Department of Electrical and Computer Engineering
10/2023 – 12/2023	**Porsche Engineering Services GmbH** Development Engineer
10/2017 – 10/2018	**Mercedes Benz, Stuttgart** Internship & Master Thesis
10/2015 – 03/2016	**Mahindra Truck and Bus Division, India** Graduate Engineer Trainee
10/2015 – 03/2015	**Hyundai Motor India Pvt. Ltd.** Bachelor Thesis