



CHALMERS
UNIVERSITY OF TECHNOLOGY



Computing controllable sets for safe and stable automated vehicles

Combining Model Predictive Control and Active Learning for Safe and Efficient Longitudinal Vehicle Dynamics

Master's thesis in Mobility Engineering

Keivan Shojaei
Kiran Kumar Palle

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

MASTER'S THESIS 2026

Computing controllable sets for safe and stable automated vehicles

Combining Model Predictive Control and Active Learning for Safe
and Efficient Longitudinal Vehicle Dynamics

Keivan Shojaei
Kiran Kumar Palle



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Division of Systems and Control
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026

Computing controllable sets for safe and stable automated vehicles
Combining Model Predictive Control and Active Learning for Safe and Efficient
Longitudinal Vehicle Dynamics
Keivan Shojaei
Kiran Kumar Palle

© Keivan Shojaei, 2026.

© Kiran Kumar Palle, 2026.

Supervisor: Jordanka Kovaceva, Vehicle Safety
Examiner: Nikolce Murgovski, Electrical Engineering

Master's Thesis 2026
Department of Electrical Engineering
Division of Systems and Control
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Integration of Model Predictive Control with Active Learning in an autonomous vehicle system to illustrate safe and unsafe regions and obtain reachable set.

Typeset in L^AT_EX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2026

Computing controllable sets for safe and stable automated vehicles
Combining Model Predictive Control and Active Learning for Safe and Efficient
Longitudinal Vehicle Dynamics
Keivan Shojaei
Kiran Kumar Pale
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Adaptive Cruise Control (ACC) is an important driver-assistance function in automated and semi-automated vehicles. A well-designed ACC system can improve longitudinal safety, traffic flow, and driving comfort by controlling the ego vehicle in response to the motion of a preceding vehicle. Maintaining safety becomes particularly challenging when the lead vehicle behaves unexpectedly or when rapid braking is required.

This thesis presents a framework that combines Model Predictive Control (MPC) and Active Learning (AL) for longitudinal safety analysis. The MPC controller computes ego-vehicle acceleration commands while accounting for the system dynamics, input limitations, and safety constraints. A more detailed plant model is used to introduce effects such as actuator dynamics, aerodynamic drag, rolling resistance, road grade, and reaction delay. In addition to controlling the vehicle, MPC is used as a feasibility oracle to classify operating conditions as feasible or infeasible. AL then selects informative states, particularly near the transition between these regions, thereby reducing the number of MPC evaluations required to approximate the safety boundary.

Simulation results indicate that the proposed framework can identify safety-critical operating conditions and approximate the MPC-feasible region while concentrating computational effort near the feasibility boundary. The results also illustrate the influence of model mismatch and critical braking conditions on the estimated safety region.

Keywords: Adaptive Cruise Control, longitudinal vehicle dynamics, Model Predictive Control, Active Learning, safety boundary, feasibility, Support Vector Machine, collision avoidance.

Acknowledgements

I would like to express my sincere gratitude to our supervisor, Jordanka Kovaceva, and our examiner, Nikolce Murgovski, for their invaluable support and guidance during our master's thesis. I would also like to thank Viktor Göransson for his friendly and kind support provided throughout the research process. My deepest appreciation goes to my beloved wife, Tina, for her endless patience, understanding, and all the support during my postgraduate studies. A special thank you goes to my little princess, Melorin, who makes my life colorful. Moreover, I would like to thank my parents, brothers and all friends for their constant encouragement, support, and their important role in achieving this milestone. Thank you for being part of this remarkable journey.

Keivan Shojaei, Gothenburg, August 2026

I would like to sincerely thank our supervisor, Jordanka Kovaceva and our examiner, Nikolce Murgovski for all the support and guidance provided during our master's thesis. I would like to thank Viktor Göransson for the wonderful discussions. I would like to thank my wife, Preethi Parasurapu, for her constant support and encouragement throughout this journey. To my daughter Dhrithi, your cute smiles gave me joy during the long hours of research. I would also like to thank my mom and dad without whom I would not be where I am today.

Kiran Kumar Palle, Gothenburg, August 2026

List of Acronyms

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

2-D	2-Dimensional
ACC	Adaptive Cruise Control
ADAS	Advanced Driver-Assistance Systems
AL	Active Learning
AV	Automated Vehicle
CBF	Control Barrier Functions
CLF	Control Lyapunov Functions
DCBF	Differentiable Control Barrier Functions
GCBF	Generalized Control Barrier Functions
ITS	Intelligent Transport Systems
ML	Machine Learning
MPC	Model Predictive Control
QP	Quadratic Programming
RBF	Radial Basis Function
SVM	Support Vector Machine
V2X	Vehicle to Everything

Nomenclature

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

Indices

t	Index for continuous time
k	Index in discrete time

Sets

$\mathbb{R}$	Set of real numbers
$\mathbb{N}$	Set of natural numbers
$\mathbb{X}$	Set of system states of $\mathbf{x}_k$
$\mathbb{U}$	Set of states of control input $\mathbf{u}_k$
$\mathcal{X}$	Set of predicted states

Parameters

C_v	Coefficient of speed tracking in cost function
C_a	Coefficient of acceleration effort in cost function
C_s	Coefficient for slack penalty in cost function
C_N	Coefficient of terminal slack penalty in cost function
Δt	Time discretization step (time interval)
m	Mass of ego vehicle and lead vehicle (assumed same)
g	Universal Gravitational Constant
ρ	Fluid density (i.e., air)
C_d	Aero Drag coefficient

A_f	Reference Area
μ	Tyre Road Friction coefficient
C_{rr}	Rolling-resistance coefficient
a_{act}	Actual acceleration state
a_{rr}	Rolling resistance force
a_{grade}	Road slope force
a_{drag}	Aerodynamic force
θ	Road angle

Variables

D_k	Relative distance between the ego and lead vehicle
D_0	Initial gap
$V_{E,k}$	Velocity of the ego vehicle
$V_{L,k}$	Velocity of the lead vehicle
$S_{E,k}$	Position of the ego vehicle
$S_{L,k}$	Position of the lead vehicle
$a_{E,k}$	Acceleration of the ego vehicle
$a_{L,k}$	Acceleration of the lead vehicle
Δt	Sampling time
Δv_k	Relative speed at discrete time
Δv_0	Initial Relative Speed
x_0	State at the beginning of optimization
x_p	Plant state
associated with the actuator dy- namics	
K_1, K_2, K_3, K_4	Intermediate slope evaluations
t_r	actuator time constant
n_r	number of delayed samples

Contents

List of Acronyms	viii
Nomenclature	x
List of Figures	xvii
List of Tables	xix
1 Introduction	1
1.1 Background	1
1.2 Related Work	2
1.2.1 MPC in ACC and Automated Vehicle Control	2
1.2.2 Reachability and Safety-Set Analysis	3
1.2.3 Active Learning Approaches	4
1.2.4 Research Gap	4
1.3 Purpose	5
1.4 Aim	6
1.5 Scope	6
1.6 Limitations	7
1.7 Ethical and Sustainability Aspects	7
1.8 Disposition	8
2 Vehicle dynamic modelling	9
2.1 Longitudinal Vehicle Dynamics	9
2.2 Vehicle States Model	10
2.3 Discrete Time Dynamic Model	11
2.4 Adaptive Cruise Controller	14
3 Model Predictive Control	15
3.1 Prediction Model	18
3.2 Cost Function Design	18
3.3 Solver	20
3.4 Terminal Sets and Recursive Feasibility	21
3.5 Ego-Vehicle Plant Model	22
3.5.1 Reaction-Time Delay	22
3.5.2 Vehicle Resistance Forces	22
3.5.3 Road-Slip Limitation	23

3.5.4	Nonlinear Plant Model and RK4 Discretization	23
3.6	Integration with Active Learning	25
3.7	Safety Boundary Identification Using SINDy	26
3.7.1	Introduction	26
3.7.2	Longitudinal Safety Boundary	26
3.7.3	SINDy-Based Equation Identification	27
3.7.3.1	Candidate Function Library	27
3.7.3.2	Sparse Regression	28
4	Machine Learning	29
4.1	Reachability and Feasibility Analysis	30
4.1.1	Data-driven approach	31
4.2	MPC-based Oracle and Safety Evaluation	32
4.2.1	Computational Cost and Motivation for Learning	32
4.3	Slack Variable and Safety Margin	33
4.3.1	Interpretation as Safety Margin	33
4.3.2	Learning Trigger Based on Slack Activation	34
4.4	Active Learning Framework	35
4.4.1	Entropy-Based Uncertainty Sampling	35
4.4.2	Sample Selection Strategy	36
4.5	SVM Classifier	37
4.6	Controllable Set	38
4.7	Model Accuracy and Performance Evaluation	39
4.7.1	Safety Boundary Approximation	40
4.7.2	Reduction of Oracle Evaluations	40
5	Results	41
5.1	Curve fitting using Sparse Regression method SINDy	41
5.2	MPC Performance	42
5.3	Slack Variable Analysis and Learning Trigger	45
5.4	AL Result	49
5.4.1	Sample Selection	49
5.4.2	Generalization Capability	50
5.5	SVM Classification Results	51
5.6	Learned Safe Controllable Set	53
5.7	Controllable Sets For Safe and Stable Automated vehicles	55
6	Conclusion	59
6.1	Equation identification with Sparse Regression	59
6.2	MPC Feasible Set Approximation	59
6.3	MPC Performance and Sensitivity to Prediction Settings	61
6.4	Future Work	62
	Bibliography	63
A	All scenarios	I

B Slack and Trigger	V
C Initial values	VII

List of Figures

1.1	Conceptual interaction between MPC-based finite-horizon prediction and AL-based exploration of safety-critical operating conditions. . . .	5
2.1	Longitudinal dynamic model	9
2.2	Operating principle of Adaptive Cruise Control for speed regulation and safe vehicle following	14
3.1	Basic structure of MPC	16
5.1	Overview of the evaluated MPC-Active Learning framework.	41
5.2	MPC response in Scenario 1 with an initial ego-vehicle speed of 38 m/s, lead-vehicle speed of 23 m/s, and relative distance of 27 m.	43
5.3	MPC response in Scenario 2 with an initial ego-vehicle speed of 24 m/s, lead-vehicle speed of 7 m/s, and relative distance of 19 m.	43
5.4	MPC response in Scenario 3 with an initial ego-vehicle speed of 27 m/s, lead-vehicle speed of 14 m/s, and relative distance of 36 m.	44
5.5	MPC response in the worst-case scenario with an initial ego-vehicle speed of 40 m/s, a stationary lead vehicle, and an initial relative distance of 100 m.	44
5.6	Slack distribution of the model	45
5.7	A sample scenario to show feasibility and infeasibility, where yellow area shows feasibility and blue shows infeasibility	46
5.8	Learning trigger activation and Slack variable in the first scenario . .	47
5.9	Learning trigger activation and Slack variable in the second scenario .	47
5.10	Learning trigger activation and Slack variable in the third scenario . .	48
5.11	Worst-case scenario slack variable and learning trigger activation . . .	48
5.12	MPC controller performance and AL trajectory by initial and SVM sample selection	56
5.13	AL trajectory based on initial and SVM sample selection after introducing plant	57

List of Tables

5.1	Initial conditions for four MPC evaluation scenarios	43
-----	--	----

1

Introduction

Adaptive Cruise Control (ACC) requires the ego vehicle to maintain a safe longitudinal distance from a lead vehicle by responding to changes in the lead vehicle's motion. Identifying the initial conditions from which safe control remains possible can be computationally demanding, especially when a large state space must be evaluated. This thesis develops a framework to combine Model Predictive Control (MPC) and Active Learning (AL) to identify safe and unsafe operating regions for longitudinal automated driving. A discrete-time vehicle model is used within the MPC controller to predict the evolution of the ego and lead vehicles while satisfying acceleration and safety constraints. The MPC is then used as an oracle to classify initial operating conditions according to whether the safety constraints can be satisfied. The plant model subsequently incorporates additional effects to better represent realistic driving conditions. AL selects informative states, particularly near the boundary between feasible and infeasible regions, to reduce the number of required MPC evaluations. Simulation results indicate that the proposed MPC–AL framework can identify critical situations and identify the safety boundary while reducing unnecessary evaluations compared with uniform sampling.

1.1 Background

Human driving performance is influenced by several factors, including fatigue, distraction, drowsiness, impairment, and unexpected behavior from other road users. These factors can increase reaction time and reduce the driver's ability to respond effectively in safety-critical situations. Even a short delay may significantly reduce the available time and distance for collision avoidance.

Advanced driver-assistance systems are intended to reduce this dependence on human reaction by continuously monitoring the surrounding traffic and applying control actions when necessary. For longitudinal vehicle control, the controller must determine an appropriate acceleration or deceleration while respecting safety and physical constraints. This requires a model of the relative motion between the ego and lead vehicles and a method for predicting how the traffic situation may evolve.

MPC addresses this problem by repeatedly solving a finite-horizon optimization problem. At each sampling instant, the controller predicts the future system behavior using a discrete-time model, determines the control sequence that best satisfies

the control objectives and constraints, and applies the first control input before repeating the optimization at the next time step. This predictive structure makes MPC particularly suitable for safety-critical longitudinal control because future violations of distance or velocity constraints can be anticipated before they occur.

However, a challenge is that evaluating a large number of possible initial conditions using MPC can be computationally expensive. If the objective is to identify the boundary between safe and unsafe operating conditions over a large state space, a uniform grid may require many MPC evaluations, including evaluations far from the decision boundary that provide little additional information. This motivates the use of AL to select informative states more efficiently.

Machine-learning methods are increasingly used in intelligent transportation systems and automated driving to support prediction, classification, and data-efficient analysis. In safety relevant applications such as ACC, we could leverage learning methods for the controller to intelligently detect a dangerous condition, infer safe operating areas, and make overall control more reliable under uncertainty on driving scenarios.

Machine learning algorithms allow us to estimate the reachable and feasible regions in many nonlinear dynamical systems via simulation methods in a cost-effective and efficient way since it doesn't need to solve very complex nonlinear equations and allows our framework be applied on various vehicle models and driving scenes.

1.2 Related Work

Several research studies have investigated rear-end collision avoidance and safety assurance using different controller and learning-based approaches. Various studies have employed methods such as Control Barrier Functions (CBFs), Lyapunov-based methods, and MPC to formulate safety constraints, guarantee collision avoidance, and determine safe operating conditions. In addition, for learning-based approaches, some methods such as Reinforcement Learning (RL) and AL, have been investigated to improve safety assessment, identify critical operating conditions, and approximate safe regions of the state space. These approaches differ in their assumptions, safety guaranties, computational requirements, and methods for characterizing safe operating regions. Some of the relevant studies are summarized below.

1.2.1 MPC in ACC and Automated Vehicle Control

MPC has been widely investigated for trajectory tracking, motion control, and longitudinal control of automated vehicles. Saljanin et al. and Yao et al. demonstrate predictive-control approaches for automated vehicle motion, including combined longitudinal and lateral control and prediction models that account for variations in vehicle velocity [1, 2]. These studies illustrate the ability of MPC to handle multi-variable vehicle dynamics while incorporating constraints directly into the control

problem.

For longitudinal vehicle control, Gao et al. and Kempf et al. investigate MPC-based approaches for improving tracking, comfort, and robustness under uncertain or changing operating conditions [3, 4]. Their results support the use of predictive optimization for determining acceleration and braking commands while accounting for vehicle dynamics and practical limitations.

MPC has also been applied to more demanding safety-critical manoeuvres. Adaptive emergency-braking strategies and prediction-based collision-avoidance methods use future vehicle trajectories and time-varying safety constraints to achieve smoother and safer control actions [5, 6]. Related work has also considered coordinated braking and steering under varying friction conditions and combinations of MPC with learning-based methods for complex vehicle systems [7, 8]. Together, these studies show that MPC provides a flexible framework for constrained vehicle control, but they also highlight the computational burden associated with repeatedly solving optimization problems across many operating conditions.

1.2.2 Reachability and Safety-Set Analysis

Reachability and set-based safety analysis provide a complementary perspective to online control. Rather than focusing only on the control action at the current state, these methods analyze sets of states that can be reached, controlled, or maintained safely under specified system dynamics and constraints. Such analysis is useful for identifying boundaries beyond which safe control may no longer be possible.

Control Barrier Functions (CBFs) and Control Lyapunov Functions (CLFs) have been used to enforce safety and stability constraints in automotive and robotic systems. These approaches formulate safety conditions as constraints that can be incorporated into an optimization problem, often using quadratic programming [9, 10, 11]. Although barrier-function methods are not equivalent to reachability analysis, they address a closely related objective: preventing the system from entering unsafe regions while maintaining the desired control performance.

Chakrabarty et al. formulate reachable-set estimation for systems with unknown or complex dynamics as a classification problem [12]. Their method selects informative input-space samples and queries an oracle, such as a simulator or physical experiment, to determine the corresponding reachability labels. The study demonstrates that data-driven sampling can reduce the number of evaluations required to approximate a set boundary compared with exhaustive sampling. This idea is particularly relevant to the present thesis because an MPC simulation can similarly be treated as an oracle for determining whether a selected driving condition satisfies the defined safety constraints.

1.2.3 Active Learning Approaches

AL aims to improve a predictive model by selecting the samples that are expected to provide the most useful information rather than labeling all available samples uniformly. In safety-boundary estimation, the most informative states are generally located near regions where the classifier is uncertain or where the transition between safe and unsafe behavior occurs.

The active reachable-set estimation method proposed by Chakrabarty et al. demonstrates how informative sampling can be used to approximate safety-related boundaries while reducing dependence on expensive oracle evaluations [12]. This provides a direct motivation for applying AL to MPC-based safety evaluation: instead of solving the MPC problem for every point in a dense state-space grid, the learning algorithm can prioritize states that are expected to improve the current estimate of the boundary.

Other learning-based safety studies combine learned policies with control-theoretic safety mechanisms. For example, Generalized Control Barrier Functions and Differentiable Control Barrier Functions have been incorporated into learning-based control frameworks to improve constraint satisfaction during learning [13, 14]. These methods differ from the AL approach used in this thesis because they modify or constrain the learning controller itself, whereas the present work retains MPC as the model-based safety evaluator and uses AL primarily for efficient sampling and boundary identification.

1.2.4 Research Gap

The reviewed literature provides strong foundations for constrained vehicle control, control-theoretic safety analysis, and data-efficient boundary estimation. However, within the body of work considered in this thesis, these topics are most often treated separately. MPC studies primarily focus on online control performance, while reachability and learning studies focus on safety-set estimation or safe learning. Comparatively less attention is given to using an MPC controller directly as an oracle within an AL loop to efficiently identify the boundary between safe and unsafe longitudinal ACC operating conditions.

This thesis addresses that gap by combining a discrete-time longitudinal vehicle model, an MPC safety evaluation, and an AL-based sampling strategy. The MPC determines whether selected initial conditions can be controlled while satisfying the imposed constraints. The resulting labels are then used to train a classifier that approximates the safe-unsafe boundary. AL concentrates subsequent evaluations on informative regions of the state space, with the objective of reducing the number of MPC evaluations required to obtain an accurate representation of that boundary.

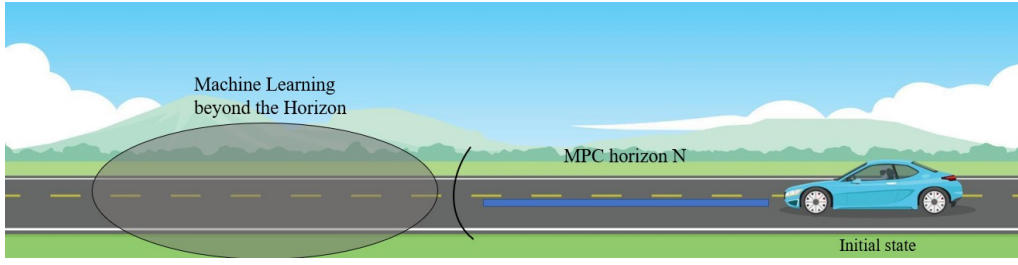


Figure 1.1: Conceptual interaction between MPC-based finite-horizon prediction and AL-based exploration of safety-critical operating conditions.

1.3 Purpose

The purpose of this thesis is to investigate how predictive control and data-efficient sampling can be combined to identify safety-critical operating conditions in longitudinal automated driving. The work considers the interaction between an ego vehicle and a lead vehicle, with particular emphasis on relative distance and relative velocity.

The MPC controller determines the acceleration or deceleration action required to maintain a safe separation and reduce undesirable relative motion. The safety requirement used in this study is defined by a minimum inter-vehicle distance of 5 m. The controller therefore aims to prevent the ego vehicle from violating this minimum separation. AL is utilized to reduce the computational cost of safety-boundary estimation by selecting informative states instead of the entire state space using the MPC-based oracle. The selection process focuses on states that are expected to provide more useful information for refining the boundary between feasible and infeasible regions. The MPC oracle evaluates the selected states and provides their safety labels, which are then used to update the Support Vector Machine (SVM) classifier.

Furthermore, the relative distance and relative velocity between the ego and lead vehicles are used to characterize the longitudinal safety state, while the slack variable provides additional information about the degree of safety-constraint violation. This information helps identify critical operating conditions, and supports the iterative refinement of the learned safety boundary.

In summary, the framework starts from the initial states of the ego and lead vehicles. The longitudinal discrete-time model is used by the MPC to determine an appropriate ego-vehicle acceleration based on the relative distance and velocity. The plant model then introduces additional effects, including aerodynamic drag, rolling resistance, reaction delay, and road-slip limitations. Finally, AL uses the MPC-based evaluations to refine the estimate of the feasibility boundary while reducing the number of required simulations.

1.4 Aim

The aim of this thesis is to develop and evaluate a reliable and computationally efficient approach for analyzing safety in automated longitudinal driving scenarios. The work is guided by the following research questions:

- How can an MPC controller be designed for longitudinal vehicle dynamics to achieve safe and stable following behavior?
- How can safety-critical operating regions be characterized in terms of the relative distance and relative velocity between the ego and lead vehicles?
- How can set-based safety information be obtained from repeated MPC evaluations and represented in a form suitable for classification?
- How can high-dimensional or nonlinear safety information be approximated using data-driven classifiers or low-complexity decision boundaries?
- How can AL be used to identify informative and safety-critical scenarios with fewer oracle evaluations than uniform sampling?
- Which quantitative metrics can be used to evaluate the safety, classification performance, and computational efficiency of the proposed MPC–AL framework?

1.5 Scope

The scope of this thesis is defined as follows:

- The system considers longitudinal interaction between an ego vehicle and a lead vehicle.
- The lead vehicle is allowed to perform different braking and acceleration manoeuvres in order to generate a range of driving scenarios.
- The primary state variables used for control and learning are relative distance and relative velocity. Acceleration and deceleration influence the state evolution through the system dynamics and control input.
- A slack variable is included in the MPC formulation to relax selected constraints when necessary. Its value is used as part of the feasibility assessment when labeling operating conditions for the AL framework.
- Slack values below 10^{-5} are treated as numerically negligible and therefore do not change the assigned feasibility label.

- Human driver reaction is not included in the control loop because the investigated system assumes automated longitudinal control.
- The analysis is limited to longitudinal vehicle motion. Lateral dynamics and road-curvature effects are outside the scope of the present study.

1.6 Limitations

The main limitations of the study are as follows:

- The ego and lead vehicles are represented using simplified longitudinal dynamics and predefined acceleration and deceleration capabilities.
- The proposed control and learning framework is evaluated only in simulation and has not been validated on a real vehicle.
- The road is assumed to be straight, and lateral vehicle dynamics are not considered.
- The classifier can only learn from the operating conditions evaluated by the oracle. Consequently, regions that are insufficiently sampled may remain poorly represented.
- Conclusions regarding computational efficiency depend on the selected state-space discretisation, MPC formulation, classifier, and AL strategy, and therefore require quantitative comparison with a suitable baseline.

1.7 Ethical and Sustainability Aspects

This thesis investigates safety in ACC systems by combining MPC with AL. From a sustainability perspective, improved longitudinal control can reduce unnecessary acceleration and braking and may therefore contribute to smoother traffic behaviour and lower energy consumption. These potential benefits are secondary to the primary objective of maintaining safe vehicle operation.

From an ethical perspective, the main objective is to reduce the risk of unsafe longitudinal interactions, particularly rear-end collisions caused by insufficient separation or delayed braking. Because automated control systems directly affect human safety, their behavior must be predictable, robust, and thoroughly validated across relevant operating conditions.

The use of a learning-based method introduces additional limitations because the classifier can only generalize from the data that have been evaluated and labeled. For this reason, safety is not delegated to the learning algorithm. MPC remains the

model-based mechanism used to evaluate the system constraints, while AL is used to improve the efficiency with which safety-critical operating conditions are explored.

All evaluations in this thesis are conducted in simulation. No conclusions are therefore made regarding direct deployment in production vehicles. Real-world implementation would require additional validation using higher-fidelity models, hardware-in-the-loop testing, and vehicle experiments before the approach could be considered for safety-critical use.

1.8 Disposition

The remainder of the thesis is organized as follows. Chapter 2 introduces the longitudinal vehicle model and the discrete-time formulation used throughout the study. Chapter 3 presents the MPC formulation, including the prediction model, objective function, constraints, optimization procedure, and ego-vehicle plant model. Chapter 4 introduces the AL framework used to identify informative operating conditions and approximate the boundary between safe and unsafe behavior. Chapter 5 presents the simulation results and evaluates the proposed approach using quantitative performance and efficiency metrics. Finally, Chapter 6 summarizes the main conclusions, discusses the limitations of the study, and outlines directions for future work.

2

Vehicle dynamic modelling

For the implementation of a well-functioning control algorithm, the vehicle dynamics must be suitably modeled to the characteristics of the ACC system. The selection of a suitable modelling concept directly impacts how well the predictions the controller can make, and the efficiency in which it is possible for this to be achieved. A suitable vehicle model is required for both MPC prediction and safety analysis. In this thesis, the model focuses on longitudinal motion and the interaction between an ego vehicle and a lead vehicle. The resulting continuous- and discrete-time representations define the states, control inputs, and disturbances used in the subsequent MPC and AL frameworks.

2.1 Longitudinal Vehicle Dynamics

Figure 2.1 illustrates the longitudinal interaction between the ego and lead vehicles and the variables used in the model. [5]



Figure 2.1: Longitudinal dynamic model

To achieve the best and reliable results from both the MPC and AL, a longitudinal vehicle model is required to describe the evolution of the system states. Relative velocities between two vehicles play a big role here but maybe that part gets overlooked at first.

The goal in longitudinal vehicle dynamics is to keep a safe distance from the lead vehicle. Avoiding collisions under different driving conditions matters too. The system behavior mainly depends on the relative distance and the velocities of both vehicles. Relative velocities between two vehicles play a big role here but maybe that part gets overlooked at first. The future motion of the lead vehicle is uncertain; for example its future speed, acceleration, maneuver, position and so on are uncertain, and these parameters should be considered when the ego vehicle adjust itself.

In this work the essential equation comes from 'Vehicle Dynamics and Control' book [15]. The longitudinal state vector is defined as:

$$x_k = \begin{bmatrix} D_k \\ V_{E,k} \\ V_{L,k} \end{bmatrix} \quad (2.1)$$

where

$$D_k \in \mathbb{R} \quad \text{represents the relative distance between the ego and lead vehicles,} \quad (2.2)$$

$$V_{E,k} \in \mathbb{R}_{\geq 0} \quad \text{represents the velocity of the ego vehicle,} \quad (2.3)$$

$$V_{L,k} \in \mathbb{R}_{\geq 0} \quad \text{represents the velocity of the lead vehicle.} \quad (2.4)$$

$$k \quad \text{represents discrete time step} \quad (2.5)$$

In addition to evaluating the ACC system under various driving scenarios, the thesis investigates emergency braking situation, worst-case scenario, when the ego vehicle cannot avoid the collision. The worst-case scenario occurs when the combination of relative distance and relative velocity are not in safe region. It means the initial relative distance is sufficiently small, and the relative velocity is quite high. Thus, in this scenario, a collision may become unavoidable because the ego vehicle cannot stop before reaching the lead vehicle. For the Worst-case scenario in this work, a stationary lead vehicle or a fixed obstacle in ahead of ego vehicle is assumed.

Because the lead vehicle is stationary, in worst-case scenario, the relative velocity becomes equal to the ego vehicle velocity. Therefore, the second state variable directly describes the speed of the ego vehicle during the braking scenario and we can neglect the third one, lead vehicles' velocity. The longitudinal motion of the ego vehicle is mainly affected by the applied acceleration or deceleration input generated by the controller. This dynamic model is later transformed into a discrete-time state-space representation for the implementation of the MPC controller and safety analysis framework.

$$x_k = \begin{bmatrix} D_k \\ V_{E,k} \end{bmatrix} \quad (2.6)$$

2.2 Vehicle States Model

The relative distance and relative velocity between the ego and lead vehicles are important variables for longitudinal collision avoidance. These parameters supply necessary information on the present state of traffic and assist the controller to estimate potential danger on upcoming time steps.

Definition of relative distance is the difference of longitudinal position between the lead vehicle and the ego vehicle, which can be expressed as:

$$D_k = S_{L,k} - S_{E,k} \quad (2.7)$$

Where $S_{L,k}$ and $S_{E,k}$ represent the longitudinal positions of the lead and ego vehicles, respectively.

The relative velocity is defined as the difference between the ego vehicle velocity and the lead vehicle velocity:

$$\Delta v_k = V_{E,k} - V_{L,k} \quad (2.8)$$

Where $V_{E,k}$ is the ego vehicle velocity and $V_{L,k}$ is the lead vehicle velocity.

The evolution of the relative distance depends directly on the relative velocity. Therefore, the distance dynamics can be written as

$$\dot{D}_k = -\Delta v_k \quad (2.9)$$

This relationship shows that the relative distance decreases when the ego vehicle moves or travels faster than the lead vehicle.

Similarly, the relative velocity dynamics depend on the accelerations of both vehicles, and can be expressed as

$$\dot{\Delta v}_k = a_{E,k} - a_{L,k} \quad (2.10)$$

Where $a_{E,k}$ and $a_{L,k}$ represent the accelerations of the ego and lead vehicles respectively.

As discussed in Section 2.1, a stationary lead vehicle is assumed to investigate worst-case emergency braking scenarios. Therefore:

$$V_{L,k} = 0 \quad (2.11)$$

For the stationary case, the lead-vehicle acceleration is also zero.

$$a_{L,k} = 0 \quad (2.12)$$

As a result, in worst-case scenario the dynamics of the system can be simplified while still representing the critical behavior required for collision avoidance and safety analysis.

2.3 Discrete Time Dynamic Model

The thesis introduces a discrete-time longitudinal vehicle model to study how an ego vehicle interacts with a lead vehicle under different conditions. The formulation tracks changes in relative distance, and relative velocity over time. An MPC

controller then evaluates safety during maneuvers. This assessment determines if the ego vehicle can react quickly enough to the lead vehicle's sudden movements to avoid an accident or not.

Ultimately, it serves as the prediction component in an MPC setup. From there, the same model is integrated into AL so that safety regions can be estimated and feasibility can be assessed and controllable set can be measured in the any scenarios for the thesis.

This model provides the basis for the MPC feasibility evaluations that are subsequently used by the AL framework.

The controller predicts the future behavior of the ego vehicle over a finite prediction horizon and computes the optimal control action that ego vehicle required to maintain safety while reducing the possibility of collision by keeping a safe distance between its vehicle and lead vehicle and reduce the relative speed between both vehicles.

Since MPC operates in discrete time, the longitudinal vehicle dynamics must first be transformed into a discrete-time state-space representation.

The general nonlinear discrete-time dynamic system can be represented as:

$$x_{k+1} = f(x_k, u_k) \quad (2.13)$$

subject to

$$x_k \in \mathbb{X}, \quad u_k \in \mathbb{U} \quad (2.14)$$

Where $x_k \in \mathbb{R}^{n_x}$ represents the system states and $u_k \in \mathbb{R}^{n_u}$ represents the control input at discrete time step $k \in \mathbb{N}$.

The general longitudinal vehicle model is written by the state variables and they are defined by the relative distance, ego vehicle velocity, and lead vehicle velocity.

The control input corresponds to the acceleration or deceleration of the ego vehicle. The discrete-time longitudinal equations of motion are formulated as:

$$D_{k+1} = D_k - V_{E,k}\Delta t + V_{L,k}\Delta t - \frac{1}{2}a_{E,k}\Delta t^2 + \frac{1}{2}a_{L,k}\Delta t^2 \quad (2.15)$$

$$V_{E,k+1} = V_{E,k} + a_{E,k}\Delta t \quad (2.16)$$

$$V_{L,k+1} = V_{L,k} + a_{L,k}\Delta t \quad (2.17)$$

Where Δt represents sampling time and other coefficients are mentioned in previous sections, Sec 2.1 and 2.2.

The relative distance dynamic describes how the range or spacing between the two vehicles changes over time according to their relative motion and accelerations. Similarly, the velocity equations describe the longitudinal motion of both vehicles over the discrete sampling interval.

In the stationary lead vehicle for investigating the worst-case emergency scenario we have:

$$V_{L,k} = 0, \quad a_{L,k} = 0 \quad (2.18)$$

which simplifies the system dynamics into [16]

$$D_{k+1} = D_k - V_{E,k}\Delta t - \frac{1}{2}a_{E,k}\Delta t^2 \quad (2.19)$$

$$V_{E,k+1} = V_{E,k} + a_{E,k}\Delta t \quad (2.20)$$

The simplified model is directly used in the MPC prediction framework developed in this thesis.

The corresponding state-space representation can therefore be written as:

$$x_{k+1} = Ax_k + Bu_k + Ew_k \quad (2.21)$$

For the three-state formulation, the state vector is [15]

$$x_k = \begin{bmatrix} D_k \\ V_{E,k} \\ V_{L,k} \end{bmatrix} \quad (2.22)$$

The initial conditions are:

$$A = \begin{bmatrix} 1 & -\Delta t & \Delta t \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} -\frac{1}{2}\Delta t^2 \\ \Delta t \\ 0 \end{bmatrix}, \quad E = \begin{bmatrix} \frac{1}{2}\Delta t^2 \\ 0 \\ \Delta t \end{bmatrix} \quad (2.23)$$

For the simplified two-state worst-case formulation, the model becomes: [5]:

$$x_k = \begin{bmatrix} D_k \\ V_{E,k} \end{bmatrix} \quad (2.24)$$

and

$$A = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} -\frac{1}{2}\Delta t^2 \\ \Delta t \end{bmatrix}, \quad E = \begin{bmatrix} \frac{1}{2}\Delta t^2 \\ -\Delta t \end{bmatrix} \quad (2.25)$$

The initial conditions of the system are defined as

$$D(0) = D_0 \quad (2.26)$$

$$V_E(0) = V_{E,0} \quad (2.27)$$

Where D_0 represents the initial relative distance between two vehicles and V_{e0} represents the initial relative velocity.

The worst-case analysis considers a high initial velocity of the ego-vehicle relative to the available stopping distance. Therefore,

$$V_{E,0} \gg 0 \quad (2.28)$$

The developed discrete-time model forms the foundation of the MPC prediction framework and enables the controller to estimate future vehicle behavior, evaluate safety constraints, and display appropriate braking actions under uncertain driving conditions.

2.4 Adaptive Cruise Controller

The main idea of an ACC system, which is classified as an automated driving feature in today's vehicle development world, is to relieve the driver by automatically observing road surroundings in the immediate vicinity of the driving scenario, allowing faster and more responsive adjustments to emerging conditions and to maintain appropriate car-following distance and speed, thereby potentially enhancing comfort, and increasing the smoothness of traffic and safety.[17]

The ACC system can operate in two primary modes. One when no vehicle/obstacle is detected ahead of the ego vehicle and the road ahead is clear and another when some object/vehicle is present ahead of the ego vehicle.

In the first case, the ACC controls vehicle speed such that it reaches the constant desired speed configured by the driver. In the second case, the ACC maintains a safe distance between its vehicle and the preceding vehicle, thereby avoiding collisions, and also the distances between vehicles are calculated by using velocity, relative velocity and relative distance etc.

As it can be seen in the following figure, Fig. 2.2, it shows its operating principle of the ACC and demonstrates how the ego vehicle adapt itself by traffic conditions ahead of the ego vehicle and other vehicle conditions [18].

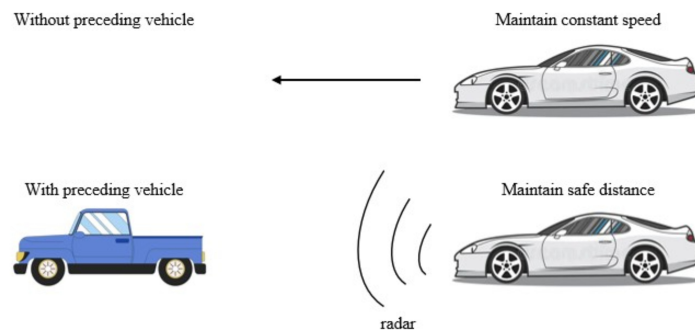


Figure 2.2: Operating principle of Adaptive Cruise Control for speed regulation and safe vehicle following

3

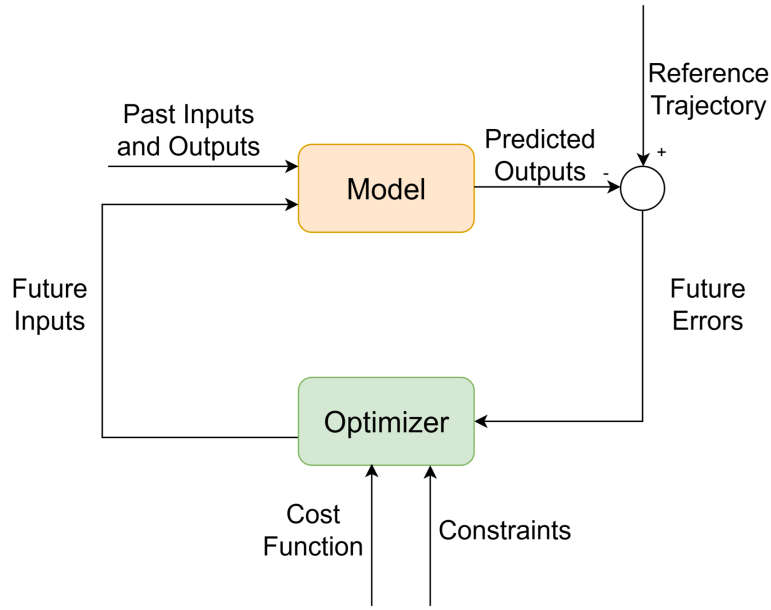
Model Predictive Control

Model Predictive Control (MPC) is selected in this thesis because it provides a practical way of combining prediction, optimization, and constraint handling in one control framework. Instead of reacting only to the current tracking error, MPC uses a model of the system to estimate how the states may evolve over a finite horizon. It then solves an optimization problem and chooses a control sequence that gives the best predicted behaviour while respecting the imposed constraints. Only the first control action is applied, after which the procedure is repeated using the latest state information. This repeated optimization is what gives MPC its receding-horizon character [19, 20, 21].

This feature is particularly useful in automated driving. A longitudinal controller is not expected to meet only one objective. It must maintain a enough distance from the vehicle in front, react to changes in the motion of the lead vehicle, avoid excessive braking or acceleration, and, when possible, maintain a desired cruising velocity. These requirements can compete with each other. For example, aggressive braking can improve safety margin but may reduce comfort, whereas a very smooth response may be too slow during an emergency. MPC offers a structured way to balance such objectives while enforcing safety-related limits directly in the optimization problem [18, 3, 1].

In this thesis work, MPC is used for a simplified longitudinal driving scenario. The controller is designed to determine a safe acceleration command for the ego vehicle and to assess whether a given initial state can be kept within the prescribed safety constraints.

Controllers within the MPC family are generally characterised by the following strategy. [21]



[21]

Figure 3.1: Basic structure of MPC

The closed-loop MPC simulation is later used together with AL to estimate the boundary between safe and unsafe initial conditions.

At every sampling instant, the controller follows the same basic sequence:

1. obtain the current state of the system,
2. predict the future state evolution over a finite horizon,
3. solve the constrained optimization problem,
4. apply the first element of the optimal control sequence,
5. repeat the calculation at the next sampling instant.

Because the optimization is solved again after every new state measurement, prediction errors and external disturbances can be taken into account in the following control step. The controller therefore operates in closed loop even though each individual optimization is based on a finite-horizon prediction.

A general discrete-time model used in MPC can be written as

$$x_{k+1} = Ax_k + Bu_k + Ew_k, \quad (3.1)$$

where x_k denotes the state vector, u_k is the manipulated input, and w_k represents an external input or disturbance. The matrices A , B , and E describe how these quantities affect the next state. In the longitudinal driving problem, the disturbance may represent the acceleration of the lead vehicle. Other effects, such as road grade, aerodynamic drag, or model uncertainty, may also be treated as disturbances depending on the model structure [19, 20].

The controller developed in this thesis uses a three-state vector. [15]. As shown in Equation (2.1), the state vector is

$$x_k = \begin{bmatrix} D_k \\ V_{E,k} \\ V_{L,k} \end{bmatrix} \quad (3.2)$$

where D_k is the relative distance between the ego and lead vehicles, $V_{E,k}$ is the ego vehicle velocity, $V_{L,k}$ is the lead vehicle velocity and

$$\Delta v_k = V_{E,k} - V_{L,k} \quad (3.3)$$

is the relative velocity as mentioned in Equation (2.8). With this sign convention, a positive value of Δv_k means that the ego vehicle is moving faster than the lead vehicle and the gap is therefore closing.

As mentioned in Equation (2.9) the corresponding continuous-time dynamics are

$$\dot{D}_k = -\Delta v_k, \quad (3.4)$$

And as mentioned in Equation (2.10),

$$\dot{\Delta v}_k = u_k - a_{L,k}, \quad (3.5)$$

where u_k is the ego-vehicle acceleration command and serves as the control input and a_L is the acceleration of the lead vehicle. These equations describe the relative motion directly. If the ego vehicle accelerates more strongly than the lead vehicle, the relative velocity increases. If the ego vehicle brakes more strongly, the relative velocity decreases.

For implementation, the continuous-time equations are discretized using the chosen sampling time Δt . The resulting discrete model is then used by the controller to predict the future gap and relative velocity over the prediction horizon [19, 20].

For a given initial state x_0 , the MPC determines whether there exists an admissible control sequence that satisfies the imposed vehicle and safety constraints over the prediction horizon. The set of initial conditions for which the MPC problem is feasible is therefore defined as

$$\boxed{\mathcal{F} = \{x_0 \in \mathcal{X} : \text{the MPC problem is feasible from } x_0\}.} \quad (3.6)$$

The main problem addressed in this thesis is to determine or approximate the set $\mathcal{F}$, thereby separating feasible (safe) initial conditions from infeasible (unsafe) initial conditions. MPC is used as the feasibility oracle, while Active Learning and classification are used to approximate the boundary of $\mathcal{F}$ with fewer MPC evaluations. The MPC result should not be interpreted only from the optimal control signal. It is also necessary to inspect the predicted distance, the actual plant response, constraint activity, and the effect of model mismatch.

MPC is used in this thesis as both a controller and a safety-evaluation tool. The three-state model predicts the future gap, ego and lead velocities, while the optimization chooses the ego-vehicle acceleration subject to the input and distance constraints. The resulting quadratic program follows the standard structure of constrained linear MPC [19, 20].

The plant model used in simulation is intentionally more detailed than the prediction model. It can include actuator dynamics, aerodynamic drag, rolling resistance, road-grade effects, and reaction delay. This difference introduces model mismatch. Rather than treating the mismatch as an implementation error, it is used to test whether the controller can still maintain safety when the actual vehicle does not behave exactly as predicted.

3.1 Prediction Model

Let the prediction horizon contain N future steps. By repeatedly applying the discrete state equation, the predicted states can be collected in the compact form

$$X = \Phi x_0 + \Gamma U + \Gamma_w W, \quad (3.7)$$

where X contains the predicted states, x_0 is the state at the beginning of the optimization, U is the sequence of future control inputs, and W contains the assumed disturbance sequence. The matrices Φ , Γ , and Γ_w are constructed from the discrete-time model matrices.

The horizon length affects both the behaviour of the controller and the computational effort. A longer horizon allows the controller to look further ahead and may improve its ability to anticipate a closing gap or a braking lead vehicle. At the same time, it increases the size of the optimization problem and may make the numerical solution more demanding. The horizon must therefore be chosen as a compromise between prediction quality and computational cost [21, 19, 20].

Although the optimization returns a complete sequence of future inputs, only the first input is sent to the plant. The remaining inputs are not applied directly because the system state will be measured again at the next sample. A new prediction and a new optimization are then carried out from the updated state.

3.2 Cost Function Design

The cost function was chosen to reflect the role assigned to the controller in this study. The minimum-distance requirement is treated primarily as a constraint, whereas the cost function is used to describe the preferred behaviour inside the feasible region. This distinction is important. A distance-tracking term would encourage the controller to follow a particular gap, but the objective here is not to force one fixed headway. The objective is to remain safe, approach the desired velocity when possible, and avoid unnecessarily large control actions.

The implemented objective is

$$J = \sum_{k=0}^{N-1} \left[C_v (v_{E,k} - v_{ref})^2 + C_a u_k^2 + C_s s_k^2 \right] + C_N s_N^2 \quad (3.8)$$

where $v_{E,k}$ is the predicted ego-vehicle velocity, v_{ref} is the desired velocity, u_k is the control input, and s_k is a slack variable associated with a softened safety constraint. The coefficients C_v , C_a , C_s , and C_N determine the relative importance of the corresponding terms.

The velocity term penalizes the difference between the predicted ego velocity and the reference velocity. Its role is to prevent the controller from braking more than necessary once the safety constraint can be satisfied. When the available gap is sufficiently large, this term encourages the vehicle to recover toward the requested cruising velocity.

The input term penalizes the square of the acceleration command. Without this term, the optimizer may choose very aggressive acceleration or braking whenever such an action reduces the other penalties. Including control effort usually leads to a less abrupt response and is common in vehicle MPC formulations [1, 2].

In the present implementation, this term should be understood as a regularization of the commanded acceleration rather than a complete passenger-comfort model. A separate jerk penalty would be required if changes in acceleration were to be penalized explicitly.

The slack penalty is included to avoid complete loss of feasibility in extreme situations. If the safety constraint cannot be satisfied exactly, the slack variable allows a limited violation so that the optimization problem still returns a solution. This does not mean that a violation is considered acceptable. The corresponding weight is chosen to be much larger than the ordinary performance weights, so the optimizer uses slack only when the available control authority is insufficient to satisfy the constraint without it [19, 20].

The terminal term

$$C_N s_N^2 \tag{3.9}$$

places an additional penalty on the slack at the end of the prediction horizon. A large terminal weight discourages solutions that postpone the safety problem until the last predicted steps. In practical terms, it encourages the controller to move the system back toward a state with a larger safety margin before the horizon ends.

A deliberate choice in the final formulation is that the distance gap is not tracked directly in the objective. The gap is instead handled through the safety constraint. This keeps the interpretation of the weights clearer: the cost describes preferred performance, while the constraint defines the minimum acceptable safety condition. Similar separation between hard safety requirements and softer performance objectives is common in constrained vehicle-control formulations [11, 6].

The selected cost function should therefore be read together with the constraints. On its own, it does not guarantee collision avoidance. Safety comes from the predicted

distance requirement, the available braking authority, the disturbance assumption, and the repeated closed-loop solution of the optimization problem. The cost function determines which feasible action is preferred once those conditions have been imposed.

3.3 Solver

The MPC optimization problem was solved as a quadratic programming problem. This formulation was suitable because the prediction model is linear, the cost function is quadratic, and the controller constraints can be written as linear inequalities. At each sampling instant, the solver calculates the future ego-vehicle acceleration sequence

$$U = \begin{bmatrix} u_{k|k} & u_{k+1|k} & \cdots & u_{k+N-1|k} \end{bmatrix}^T, \quad (3.10)$$

over the prediction horizon N . The predicted states are obtained from the condensed model

$$X = \Phi x_k + \Gamma U + \Gamma_w W, \quad (3.11)$$

where $x_k = [D_k \ V_{E,k} \ V_{L,k}]^T$ is the current state, W contains the assumed lead-vehicle acceleration, and X contains the predicted distance and relative-velocity states.

The MPC cost function was converted into the standard quadratic programming form

$$\min_U \quad \frac{1}{2} U^T H U + f^T U, \quad (3.12)$$

where the matrix H contains the quadratic penalties on state errors, acceleration magnitude, and acceleration changes. The vector f depends on the current state, the reference values, the previously applied control input, and the predicted lead-vehicle motion.

The optimisation was solved subject to the acceleration limits

$$u_{\min} \leq u_{k+i|k} \leq u_{\max}, \quad (3.13)$$

and the minimum-distance constraint

$$D_{k+i|k} \geq D_{\min}. \quad (3.14)$$

These constraints were rearranged into the matrix form required by the solver:

$$A_{\text{ineq}} U \leq b_{\text{ineq}}. \quad (3.15)$$

In cases where strict satisfaction of the distance constraint could make the optimisation infeasible, a non-negative slack variable was included. The slack variable was assigned a large penalty so that the controller only allowed a constraint violation when no feasible solution existed without it. Soft constraints of this type are commonly used in practical MPC implementations to improve numerical robustness

[19, 21].

The resulting optimisation problem was solved using the MATLAB `quadprog` function:

```
[U_opt,~,exitflag] = quadprog(H,f,Aineq,bineq,...
                               [],[],[],[],options);
```

Only the first control value of the optimal sequence was applied to the plant:

$$u_k = U_{\text{opt}}(1). \quad (3.16)$$

At the next sampling instant, the updated plant states were fed back to the MPC and the quadratic programming problem was solved again. In this way, the controller continuously updated the acceleration command based on the latest distance and relative-velocity measurements. The solver status was monitored using the returned `exitflag`, and a fallback braking command could be used if no valid solution was obtained.

3.4 Terminal Sets and Recursive Feasibility

An important consideration in finite-horizon Model Predictive Control is recursive feasibility. An MPC problem is recursively feasible if feasibility at the current sampling instant implies that the optimization problem remains feasible at the following sampling instant under the applied control law. One common approach for establishing this property is to introduce a terminal constraint of the form [19, 22]

$$x_N \in \mathcal{X}_f, \quad (3.17)$$

where $\mathcal{X}_f$ is a terminal invariant set. The set is chosen such that, once the predicted state reaches $\mathcal{X}_f$, an admissible local control law exists that keeps the subsequent states inside the constraints. Together with suitable terminal ingredients, this provides a standard mechanism for establishing recursive feasibility and closed-loop stability in MPC.

The MPC formulation used in this thesis does not explicitly impose a terminal invariant-set constraint. Instead, safety is enforced over the finite prediction horizon through the distance and input constraints, while a strongly penalized slack variable is introduced to avoid complete loss of feasibility in critical situations. An additional terminal penalty on the slack variable discourages the optimizer from postponing a constraint violation to the end of the prediction horizon. However, this terminal penalty should not be interpreted as a terminal set $\mathcal{X}_f$.

Consequently, recursive feasibility is not formally guaranteed by the present MPC formulation. Feasibility is evaluated numerically at each sampling instant, which is sufficient for the MPC-based feasibility classification used in this work, but it does not constitute a proof that a feasible state will remain feasible indefinitely.

A natural extension of the present framework would therefore be to compute a positively invariant terminal set $\mathcal{X}_f$ for the longitudinal vehicle model and impose $x_N \in \mathcal{X}_f$ as an additional MPC constraint. This would provide a stronger theoretical

connection between the finite-horizon MPC feasibility region and the controllable safe set considered in this thesis.

3.5 Ego-Vehicle Plant Model

The plant represents the actual longitudinal response of the ego vehicle after receiving the acceleration command from the MPC. Unlike the simplified linear model used by the controller, the Simulink plant includes actuator dynamics, aerodynamic drag, rolling resistance, road grade that can limit braking value. These effects introduce a realistic mismatch between the controller prediction and the actual vehicle response [16].

3.5.1 Reaction-Time Delay

The acceleration command calculated by the MPC is not applied to the actuator immediately. A discrete delay is used to represent the reaction time between generating the command and beginning the vehicle response. The delayed command is written as

$$u_{\text{del},k} = u_{k-n_r}, \quad (3.18)$$

where the number of delayed samples is

$$n_r = \text{round} \left(\frac{t_r}{\Delta t} \right). \quad (3.19)$$

Here, t_r is the reaction delay and Δt is the simulation sampling time. In Simulink, this behaviour is implemented using a discrete delay block placed between the MPC output and the actuator model.

3.5.2 Vehicle Resistance Forces

Aerodynamic drag is modelled as

$$F_{\text{drag}} = \frac{1}{2} \rho C_d A_f v_E^2, \quad (3.20)$$

which gives the corresponding acceleration

$$a_{\text{drag}} = -\frac{\rho C_d A_f v_E^2}{2m}. \quad (3.21)$$

Rolling resistance and road-grade acceleration are given by

$$a_{\text{rr}} = -g C_{\text{rr}}, \quad a_{\text{grade}} = -g \sin \theta, \quad (3.22)$$

where C_{rr} is the rolling-resistance coefficient and θ is the road angle. These expressions follow standard longitudinal vehicle-dynamics formulations [16].

The resulting ego-vehicle acceleration is calculated as

$$a_{E,k} = a_{\text{act},k} + a_{\text{drag},k} + a_{\text{rr}} + a_{\text{grade}}. \quad (3.23)$$

The following parameters are used in the plant model:

$m = 1500 \text{ kg}$	Vehicle mass
$C_{rr} = 0.01$	Rolling resistance coefficient
$\rho = 1.255 \text{ kg/m}^3$	Air density
$C_d = 0.3$	Aerodynamic drag coefficient
$A_f = 2.2 \text{ m}^2$	Frontal area
$\theta = 5^\circ$	Road grade angle
$t_{\text{reaction}} = 0.2 \text{ s}$	Reaction time

3.5.3 Road-Slip Limitation

The available braking force depends on the tyre–road friction coefficient μ . The maximum achievable braking acceleration is limited by

$$a_{\text{fric}} = \mu g. \quad (3.24)$$

Therefore, the effective lower acceleration limit used by the plant is

$$a_{\text{min,eff}} = \max(a_{\text{min}}, -\mu g). \quad (3.25)$$

This means that the plant may produce less braking than requested by the MPC when the road friction is low.

3.5.4 Nonlinear Plant Model and RK4 Discretization

The plant model used for simulation should be more detailed than linear prediction model used by the MPC. The simulation plant is more detailed than the linear MPC prediction model because it includes additional nonlinear and resistive effects. Consequently, the plant dynamics are nonlinear and cannot be discretised using the constant-acceleration formulation used for the linear prediction model. The continuous-time plant is therefore formulated as a nonlinear differential equation and discretised using a fourth-order Runge–Kutta (RK4) method.

The plant state is defined as [15]:

$$x_k = \begin{bmatrix} D \\ V_E \\ V_L \end{bmatrix} \quad (3.26)$$

where D is the relative distance, v_E and v_L are the ego- and lead-vehicle velocities, respectively, and a_{act} is the actual acceleration state associated with the actuator dynamics. The control input u represents the commanded ego-vehicle acceleration, while the lead-vehicle acceleration a_L is treated as an external disturbance. The continuous-time plant can be written in the general form:

$$\dot{x}_p = f_p(x_p, u, a_L) \quad (3.27)$$

Where

$$\dot{D} = v_L - v_E \quad (3.28)$$

$$\dot{v}_E = a_E \quad (3.29)$$

$$\dot{v}_L = a_L \quad (3.30)$$

$$\dot{a}_{act} = \frac{u - a_{act}}{t_r} \quad (3.31)$$

Where t_r denotes actuator time constant.

The final longitudinal acceleration of the ego vehicle combines the commanded acceleration with the plant dynamics and resistance effects.

$$a_{E_{final}} = a_{MPC} + a_{act} + a_{drag} + a_{rr} + a_{grade}. \quad (3.32)$$

Therefore, the nonlinear plant dynamics can be written explicitly as

$$\dot{x}_k = \begin{bmatrix} V_L - V_E \\ \dot{V}_E \\ \dot{V}_L \end{bmatrix} \quad (3.33)$$

Then

$$f_p(x, u, w) = \begin{bmatrix} v_L - v_E \\ a_E \\ w \end{bmatrix} \quad (3.34)$$

The acceleration depends nonlinearly on the ego-vehicle velocity in the aerodynamic drag parameter and the actuator state evolves during the sampling interval, the plant is discretised numerically rather than using the constant-acceleration update employed for the linear prediction model.

For a sampling time Δt , the fourth-order Runge–Kutta method is applied as follows. Given the plant state $x_{p,k}$ control input u_k , and lead-vehicle acceleration $a_{L,k}$, the four intermediate slopes are calculated as [23]:

$$K_1 = f_p(x_{p,k}, u_k, a_{L,k}) \quad (3.35)$$

$$K_2 = f_p(x_{p,k} + \frac{\Delta t}{2} K_1, u_k, a_{L,k}) \quad (3.36)$$

$$K_3 = f_p(x_{p,k} + \frac{\Delta t}{2} K_2, u_k, a_{L,k}) \quad (3.37)$$

$$K_4 = f_p(x_{p,k} + \Delta t K_3, u_k, a_{L,k}) \quad (3.38)$$

So the plant state for the next sampling time is obtained with [23]:

$$x_{p,k+1} = x_{p,k} + \frac{\Delta t}{6} (K_1 + 2K_2 + 2K_3 + K_4) \quad (3.39)$$

The approach is a numerical approximation of the nonlinear continuous time plants over a sample time. At every MPC sample instant k , the controller sends the acceleration command u_k , the nonlinear plant is simulated one sample ahead using the RK4 method, and the final plant states are used as the initial conditions for the subsequent MPC optimization. Note how the linear prediction model and the nonlinear simulation plant are kept separate; the MPC is performed based on the linear prediction model and then the nonlinear plant is simulated to verify the controller performance with model mismatch.

3.6 Integration with Active Learning

In this thesis, the MPC simulation is also used for labeling a set safe/unsafe for AL. For each selected initial condition, the closed-loop simulation determines whether the safety constraints can be maintained. The result of that simulation is then used to classify the initial condition as safe or unsafe. A similar simulation-based idea has been used for estimating reachable-set boundaries while reducing the number of required model evaluations [12].

The initial state considered by the learning algorithm is

$$x_0 = [D_0, V_{E,0}, V_{L,0}]^T, \quad (3.40)$$

where D_0 is the initial gap, $V_{E,0}$ is the initial ego velocity and $V_{L,0}$ is the initial lead vehicle velocity. For each sample, the MPC controller is simulated under the selected lead-vehicle scenario. The state is assigned a safety label according to

$$\text{Safe}(x_0) = \begin{cases} 1, & \text{if the safety constraints remain satisfied,} \\ 0, & \text{otherwise.} \end{cases} \quad (3.41)$$

The procedure starts with a limited set of initial samples spread over the state space. After these points have been simulated and labelled, a classifier is trained. The classifier is then used to identify regions where the predicted class is uncertain. New samples are selected close to this uncertainty region and evaluated through additional MPC simulations. In this way, the learning process gradually concentrates on the safe–unsafe boundary instead of spending most of the simulation budget on points that are obviously safe or obviously unsafe [12, 24].

This is useful because exhaustive gridding becomes expensive as the number of states or uncertain parameters increases. Active learning does not remove the need for simulation, but it can use the available simulations more effectively by prioritizing informative points. In the present two-dimensional case, this makes the evolution of the learned boundary easy to visualize. The same principle can later be extended to higher-dimensional settings, although interpretation becomes more difficult.

The combined MPC and AL framework therefore serves two purposes. MPC provides the closed-loop control action and checks whether the constraints can be sat-

ified, while active learning uses those results to approximate the controllable-set boundary with fewer simulation runs than a dense grid would require.

The resulting plant response is subsequently used in the AL-based safety evaluation. The final ego vehicle acceleration is combination of MPC evaluation and other parameters that are added in Plant.

$$a_{E_{final},k} = a_{MPC,k} + a_{act,k} + a_{drag,k} + a_{rr,k} + a_{grade,k}. \quad (3.42)$$

3.7 Safety Boundary Identification Using SINDy

3.7.1 Introduction

The safety boundary of the longitudinal vehicle model can be derived analytically for the simplified braking scenario considered in this work. However, for more complex vehicle models, obtaining such a closed-form expression may become difficult. Therefore, Sparse Identification of Nonlinear Dynamics (SINDy) was explored as a method for obtaining a compact mathematical representation of a safety boundary. SINDy was originally proposed as a data-driven method for discovering governing equations of nonlinear dynamical systems. The central idea is that many physical systems can be represented using only a small number of active terms selected from a larger library of possible functions. Sparse regression is then used to identify these terms and produce a compact and interpretable model [25]. This approach has subsequently been extended to control applications, including model predictive control [26].

In this work, the same sparse identification principle is applied to the longitudinal safety function. Since the analytical safety equation is already available for the simplified vehicle model, the purpose of this experiment is not to replace the analytical derivation. Instead, the known solution provides a reference case for evaluating whether sparse equation identification can recover the correct structure of the safety boundary. This serves as a first step towards applying similar methods to more complicated vehicle models where an analytical safety boundary may not be readily available.

3.7.2 Longitudinal Safety Boundary

As mentioned in Equation (2.1), the longitudinal state considered for the safety analysis is

$$x = \begin{bmatrix} D & v_E & v_L \end{bmatrix}^T, \quad (3.43)$$

where D represents the distance between the ego and lead vehicles and

$$\Delta v = v_E - v_L \quad (3.44)$$

is the relative closing velocity, as mentioned in equation (2.8). A positive value of Δv therefore indicates that the ego vehicle is approaching the lead vehicle.

For the simplified braking scenario, the ego vehicle applies a constant maximum braking acceleration a_E , while the lead vehicle is assumed to have acceleration a_L . No reaction-time delay is included in this analysis.

As mentioned in Equation (2.10), the relative acceleration is

$$a_{\text{rel}} = a_E - a_L. \quad (3.45)$$

When the ego vehicle has sufficient braking authority relative to the lead vehicle, the minimum distance required to eliminate the initial closing velocity can be expressed as

$$D_{\text{required}} = d_s + \frac{\Delta v^2}{2|a_E - a_L|}, \quad (3.46)$$

where d_s is the minimum distance that must remain between the two vehicles. This expression defines the analytical safety boundary. Initial conditions satisfying

$$D \geq D_{\text{required}} \quad (3.47)$$

are considered safe under the assumed braking conditions, whereas points below the boundary do not provide sufficient distance to satisfy the required safety margin.

A continuous safety function can therefore be written as

$$h(D, \Delta v) = D - D_{\text{required}}, \quad (3.48)$$

or equivalently,

$$h(D, \Delta v) = D - d_s - \frac{\Delta v^2}{2|a_E - a_L|}. \quad (3.49)$$

The sign of h provides a direct interpretation of the state:

$$h(D, \Delta v) \begin{cases} > 0, & \text{safe region,} \\ = 0, & \text{safety boundary,} \\ < 0, & \text{unsafe region.} \end{cases} \quad (3.50)$$

This continuous safety function was used as the target for the SINDy equation identification.

3.7.3 SINDy-Based Equation Identification

3.7.3.1 Candidate Function Library

A key step in SINDy is the construction of a library containing candidate functions that may describe the relationship of interest. SINDy then uses sparse regression to determine which of these candidate functions are required to represent the data [25].

A second-order polynomial library was selected in this work. For the states D and Δv , the candidate library is

$$\Theta(D, \Delta v) = [1 \quad D \quad \Delta v \quad D^2 \quad D\Delta v \quad \Delta v^2]. \quad (3.51)$$

The library intentionally contains more terms than are required by the analytical solution. The sparse regression algorithm is then responsible for identifying which terms are necessary and eliminating the remaining terms.

The general candidate representation can therefore be written as

$$h(D, \Delta v) = \xi_0 + \xi_1 D + \xi_2 \Delta v + \xi_3 D^2 + \xi_4 D \Delta v + \xi_5 \Delta v^2, \quad (3.52)$$

where the coefficients ξ_i are determined through sparse regression.

3.7.3.2 Sparse Regression

The implementation was carried out using the PySINDy package, which provides tools for constructing feature libraries and applying sparse optimization methods for SINDy-based model identification [27].

A polynomial feature library of degree two was combined with the Sequentially Thresholded Least Squares (STLSQ) optimizer. STLSQ is commonly used within the SINDy framework to obtain sparse coefficient matrices [25].

The optimizer first estimates coefficients for the candidate terms and then removes terms whose coefficients fall below a selected threshold. The regression is subsequently repeated using the remaining terms. In this way, unnecessary candidate functions can be removed and a compact equation is obtained.

For the present problem, the feature library contains terms such as D^2 , $D\Delta v$, and Δv , even though they are not required by the analytical safety function. A successful sparse identification should therefore eliminate these terms while retaining the terms that describe the underlying safety relationship.

4

Machine Learning

Active Learning (AL) is used in this thesis to reduce the number of expensive MPC evaluations required to approximate the MPC-feasible region. Rather than evaluating the entire state space uniformly, AL selects informative operating conditions, particularly those near the current classification boundary. [28, 29].

Our framework utilizes an MPC as a safety Oracle to determine if a sampled state is feasible or not with stopping distance, and system dynamics. The labels produced are used for training a classifier to approximate safe and unsafe operation areas of a vehicle [30].

Pool-based batch sampling approach is proposed here to sample new data by using uncertainty and information-based method to maximize sampling information gain and reduce overlap [31]. Moreover, Physics informed sampling with entropy based uncertainty estimations have been developed [12].

The main idea for this thesis is to apply the concept of AL to characterize critical driving scenarios and learn about the system safety boundary of the controller. This approach aims to specifically look for the parts of the space where it is hard for the controller to satisfy the safety constraints because this is exactly where the information for learning is the richest. Slack variables are added in the optimization problem to evaluate the extent to which the constraints are violated. Furthermore, a trigger is added to avoid insignificant slack values that come from numerical issues or minor disturbances and provide a more reliable representation of the controller's behavior in different states.

The learning is mainly induced by the relative distance between the two ego and lead vehicles, however it implicitly depends on various related variables, including the relative velocity, and relative deceleration capability of vehicles etc. Critical regions are extracted based on this process and a pool of new unlabeled samples are generated and examined in terms of controllable set and safety aspects. By utilizing an oracle-based framework (in our case MPC is acting as safety oracle), information on the stable and safe set of the controller can be collected.

The oracle can be represented as [12]:

$$\mathcal{O}(x_0) = \begin{cases} +1, & \text{if the MPC finds a feasible solution} \\ -1, & \text{otherwise} \end{cases}, \text{for } x_0 \in \mathbb{X} \quad (4.1)$$

Feasible states are assigned the label +1, whereas infeasible states are assigned the label -1. These labeled observations were utilized to train a classifier which closely resembles the system's safety boundary. The developed model is compared to the analytical stopping-distance boundary, and simulations are run to demonstrate the validity and evaluate the accuracy and reliability of the proposed approach.

4.1 Reachability and Feasibility Analysis

Reachability analysis is an important concept in control systems, and autonomous driving applications since it determines whether a vehicle is capable of evolving safely from an initial condition to a desired state while satisfying system constraints [5].

For a discrete-time dynamical system 2.3, the reachable set consists of the states that can be reached from a specified initial set under admissible control inputs. The dynamical system is represented as:

$$x_{t+1} = f(x_t, u_t) \quad (4.2)$$

Where $(x_t \in \mathbb{R}^{n_x})$ denotes the system state and $(u_t \in \mathbb{R}^{n_u})$ shows the control input at discrete-time by step (t). In this work, the state vector mainly consists of the relative distance, starts with relative initial distance, and the ego and lead vehicle velocity, while the control input corresponds to the acceleration or deceleration applied to the ego vehicle where lead vehicle's acceleration or deceleration is considered as a noise or disturbance. In addition, worst-case scenario is solved where the states is reduced to two states, relative distance and relative velocity, and the ego vehicle cannot stop before the lead vehicle and collision happens. In this thesis the worst-case scenarios is when the ego vehicle moving fast and the lead vehicle is parked, $a_l = 0$, to investigate the accident and display how our controller is worked.

Moreover, feasibility is strongly linked to the problem of reachability analysis. A state is feasible if the MPC controller can determine a sequence of inputs in order to satisfy all the constraints of the vehicle system without jeopardizing safe driving operation, whereas if it's not possible, the state is considered infeasible or unsafe.

In fact, in autonomous driving systems, feasibility becomes significantly important, particularly in critical operating situations, where the ego vehicle is moving in uncertain region in comparison to the lead vehicle. It means that relative velocity exceeds the velocity corresponding to the target distance at a certain time.

The reachable set $\mathcal{R}(T) \subseteq \mathbb{X}$ of a target set T is defined as the collection of all initial states from which there exists a sequence of admissible control inputs that drives the system to the target set while satisfying the system constraints over a finite prediction horizon and sampling time t . Mathematically, the reachable set can be expressed as [12]:

$$\mathcal{R}(T) = \{x_0 \in \mathbb{X} : \exists u_t \in \mathbb{U}, x_{t+1} = f(x_t, u_t) \in \mathbb{X}, x_T \in T\} \quad (4.3)$$

Where $\mathbb{X}$ denotes the state space, $\mathbb{U}$ represents the control input set, and T corresponds to the target safe region. However, the reachable set contains all states from which the system can safely evolve toward the target region under feasible control actions. In simple terms, it contains all states that are in a safe region with given sampling time and prediction horizon.

To evaluate feasibility, the MPC controller is implemented as a safety oracle and AL uses this evaluation for its investigation to predict the boundary between safe and unsafe region.

4.1.1 Data-driven approach

Analytical calculations of reachable and feasible regions for nonlinear or constrained system is computationally hard to be perform on real time. Data-driven approach such as simulating system with various initial conditions in order to estimate reachable and feasible regions of a given system offers computationally feasible solution. With the assistance of classification of states for feasibility of the MPC optimization problem via classification on sampled data, reachability analysis is converted into a classification problem. As we will discuss later, in this thesis, the MPC controller is used as an oracle to evaluate the feasibility of different operating conditions. For each sampled initial state (x_0), the oracle determines whether a feasible solution exists or not based on equation (4.1) and MPC defines how much acceleration or deceleration the system needs to be in a safe region.

A feasible label indicates that the ego vehicle can satisfy the imposed constraints and avoid a collision.

Evaluating safety oracle might cost a significant amount of computational time because the MPC optimization problem is solved across the prediction horizon for each query. As a solution, AL is employed. This approach is used, instead of sampling points in the state space randomly, choosing samples near the boundary, between the feasible and infeasible space. The reason behind this idea is that the border region contains the highest uncertainty and provides the most useful information in terms of reducing the classification error rate and improving it. The uncertainty of the classifier is calculated by using the estimated probabilities and different measures such as entropy-based to evaluate the uncertainty of the classifier's prediction [12]. High uncertainty states are selected, and labeled, and then used as labeled data. On the other hand, only a few sample evaluations are required from other regions that our system know it is safe or not and the system collect them to increase its prediction of all region. Hence, the proposed framework efficiently minimize the number of expensive oracle evaluations which usually costly to acquire with the proposed framework to obtain an accurate estimation of the safety boundary.

Therefore, the final objective of the learning framework is to approximate the controllable set of the system to estimate the transition boundary between safe and unsafe working conditions, and this learned safety boundary can be implemented in practice as for safety assessment, online safety monitoring, and decision support of the system in autonomous driving application.

4.2 MPC-based Oracle and Safety Evaluation

Here, MPC has not only been used as a feedback controller, but also as a decision tool for safety estimation. More specifically, the optimization of the MPC for varying initial conditions has been used to check if a given configuration of the system would allow to perform its task within the constraints. MPC has thus been used as an oracle to label states of the system as feasible or infeasible. This will allow for the construction of the data-driven learning formulation. Due to the longitudinal dynamics of vehicle and discretizations time and also horizon time, the MPC should solved the optimization constraint problem at each step time in Eq(3.8).

While our system dynamic is given in Eq(3.1) and we have to employ on that. Our vehicle constraints for this thesis are defined as:

$$u_{min} \leq u_k \leq u_{max} \quad (4.4)$$

$$D_k \geq d_{min} \quad (4.5)$$

Where D_k shows the internal relative distance between two vehicles, u_k is the ego vehicle acceleration (control input), controller that is designed to avoid collision, and a_l represents the lead vehicle acceleration, which is added as a disturbance in our control system. And again in the worst-case scenario, the lead vehicle is assumed to be stationary, so $a_l = 0$.

For each sampled state, the MPC oracle determines whether the safety constraints can be satisfied over the prediction horizon. States for which the required minimum distance is maintained are labelled feasible; states that require a significant constraint relaxation or lead to collision are labelled infeasible. In practice, in case of emergency, strict feasible may become restrictive so a slack variable is introduced to relax the constraint on the state.

$$D_k \geq d_{min} - s_k, \quad s_k \geq 0 \quad (4.6)$$

The slack variable penalizes constraint violations in the cost function and provides a measure of how severely safety constraints are violated when feasibility cannot be strictly maintained.

4.2.1 Computational Cost and Motivation for Learning

To analyze the Oracle, a Quadratic Program for the forecast horizon is solved for every sampled initial condition, which is computational intensive when numerous

samples are needed for accurate representation of the boundary region. In response to this, a data-driven learning is formulated where the Oracle is interrogated in a targeted manner. Rather than query the Oracle across all state values, AL ensures that sample locations, where boundary characterization would be most beneficial (especially near feasibility limit), are strategically targeted for evaluation. Thus, the Oracle implies a specification of the reachable set to within a constant level of accuracy. In addition, the Oracle boundary approximation is therefore an approximately feasible set, under MPC control policies.

4.3 Slack Variable and Safety Margin

In safety-critical optimal control problems, strict enforcement of state constraints can lead to infeasibility for some problems, especially in emergency driving situations when the vehicle operates at the boundaries of the state constraints. For the longitudinal vehicle control problem, this is commonly achieved when the internal vehicle distance approaches to the smallest allowed safe distance while relative speed is high. To address this issue a slack variable is introduced into MPC to relax the hard state constraint in a controlled manner and it is mentioned in equation (4.6) where s_k represents the slack variable at time step k . This approach penalizes infraction of the safety restriction where strict satisfaction of the restriction is impossible, and discourages unsafe behavior through the cost function. The following equation, Eq 4.7, shows the slack variable in MPC cost function:

$$J = J_{\text{MPC}} + \rho s_k^2 \quad (4.7)$$

Where $\rho \gg 1$ is a large weighting factor that penalises constraint violations heavily. In this work, a high penalty value is selected to ensure that slack activation occurs only in extreme or unavoidable situations.

4.3.1 Interpretation as Safety Margin

The slack variable can be interpreted as an indicator of the magnitude of safety-constraint violation. If $s_k = 0$, the system is in a region that is strictly safe and feasible. If $s_k > 0$, the system is in the region of conflict where the safety constraint cannot be respected and a compromise has to be made. Thus, the slack variable's magnitude quantifies the exact overshoot at the optimum, rather than acting as a simple binary indicator.

However, the slack variable refines this notion by distinguishing between:

- strictly feasible solutions ($s_k = 0$),
- weakly feasible or relaxed solutions ($s_k > 0$)

This differentiation is relevant to the construction of the learning data-set: high value states often have the slack feature being closed to the transition point from feasible or infeasible regions. Slack also helps design the AL policy. States are assigned a higher amount of slack when they are closer to the safety boundary, and therefore they are more informative. These states are preferentially sampled so that

the learning system more rapidly identifies the boundary region between the safe and the unsafe states

4.3.2 Learning Trigger Based on Slack Activation

For a safety-critical optimal control problem, it might be too restrictive to impose hard constraints on state variables. This can result in infeasible optimization problems, especially in an emergency driving scenario with the vehicle near boundary of state space. As for the problem of longitudinal vehicle control in this thesis, these are common when inter-vehicle distances are very close and the relative velocities are high. In this scenario, the optimizer cannot respect all of the safety constraints while having a feasible system.

In order to solve this problem, a slack variable has been incorporated into the MPC formulation to gently enforce safety hard constraints. This slack variable (s_k) enables the system to violate safety requirements temporarily if it cannot satisfy the required security. While making the optimization feasible, it imposes a penalty for violations of safety constraints into the objective function. The slack variable is shown in (4.7) and in this work the slack penalty is selected as:

$$\rho_{\text{slack}} = 10^6 \quad (4.8)$$

In addition, the slack variable gives us useful information about our distance from the safety boundary, however numerical optimizations very seldom produce a true zero output due to imperfect numerical stability. For this reason even though our safety constraints might seem satisfied numerically very tiny positive values of slack may be observed. Also a learning trigger is introduced to prevent learning mechanism from reacting to numerical noise.

$$s_k > s_{\text{trigger}} \quad (4.9)$$

Where (s_{trigger}) denotes the minimum slack value required to activate learning. In this work,

$$s_{\text{trigger}} = 10^{-5} \quad (4.10)$$

The threshold is large enough in order to be substantially greater than the numerical tolerance of the optimization solver, and allow detection of significant constraints violation. Any time the slack is beyond the trigger value the operating condition is labeled informative and saved for learning. This type of condition usually sits on the edge between the safe and unsafe region, and thus the separation edge is where there is greatest classification uncertainty and where further information would be most helpful.

The trigger therefore acts as a filtering mechanism that suppresses small numerical variations while retaining safety-relevant constraint violations. By confining data samples close to safety boundary, AL process enhances the estimations quality and reduces drastically cost of MPC evaluations during training.

4.4 Active Learning Framework

The AL framework is introduced to reduce the computational effort required to identify the boundary between feasible and infeasible operating conditions. Evaluating the safety of every possible operating condition in using the MPC controller would be computationally expensive, since each evaluation needs to solve an optimization problem over the prediction horizon.

As the size of the state space increases, performing an exhaustive analysis quickly becomes impractical. Therefore, an AL framework should introduce to efficiently identify the boundary between safe and unsafe operating conditions while reduced the number of required MPC evaluations. Model-based safety assessment is combined with machine learning in this work and machine learning focuses on operating condition that it expects to have useful information about the boundary.

In this thesis, the most focus in the system is in the region where the behavior changes from feasible to infeasible. The learning begins by generating a set of initial labeled samples in the space consisting of the state formed by the relative distance and relative velocity between the ego and the lead vehicles. The MPC-based oracle is used to prove the labels required for training the learning model and for each selected data oracle determines the feasibility and infeasibility. After that, these labeled samples are used for training Support Vector Machine (SVM) classifier which learn the approximation of the boundary based on the safe and unsafe region of the state space. Next, the philosophy of AL based on SVM in this thesis is selecting states that are more useful and informative rather than select arbitrary new samples. Thus the AL strategy prioritize states for which the classifier is the most uncertain and also the most useful to predict safety boundary.

Besides the uncertainty sampling, the developed system is able to utilize knowledge of vehicle braking dynamics. Samples close to the stopping-distance-based physical boundary are highly informative for learning. Instead, the learning concentrates on areas which are not only uncertain, from classification sense, but which are also safety-relevant from a vehicle dynamics sense. Each sample will be selected and evaluated by the MPC based oracle and labeled as a database. Next, the classifier is retrained with the augmented labeled dataset, and therefore the estimated safe boundary gets closer to the ground truth boundary. The process then iterates until the desired accuracy is achieved while minimizing expensive oracle calls, effectively learning the boundary shapes of reachable and feasible regions. Eventually, AL produces an empirically determined safe reachable set that can highlight safe regions of operation, inform safety analysis, and estimate a computationally tractable safe boundary corresponding to the one resulting from the MPC.

4.4.1 Entropy-Based Uncertainty Sampling

Shannon-Entropy is employed for the first purpose of AL for recognizing safety boundary and label each state based on that. The Shannon Entropy defined the

states as [12]:

$$H(x) = -p(x) \log_2(p(x)) - (1 - p(x)) \log_2(1 - p(x)) \quad (4.11)$$

Where $p(x)$ shows the probability of the states. The probability $p(x)$ lies between 0 and 1. Predictions close to 0 or 1 indicate high classifier confidence, whereas $p(x) \approx 0.5$ corresponds to maximum uncertainty and therefore maximum Shannon entropy.

Upon querying the classifier they are added to the training set and are used to re-train the classifier. Through repeated interrogation of such samples the algorithm iteratively learns an increasingly good approximation of the safety boundary and in doing so avoid interrogating sample locations about which the learning algorithm already has knowledge. The approach saves costly evaluations by focusing computational resources on operating conditions for which uncertainty is high.

Entropy-based uncertainty sampling is therefore highly effective in reducing the number of sample queries required to obtain a sufficiently accurate approximation of the safe reachable set.

4.4.2 Sample Selection Strategy

The AL framework further integrates vehicle dynamics and the spatial location information of the already-labeled states to speed up learning. SVM is employed to increase data selection accuracy with collecting more informative states. Although SVM is implemented to select informative data and label them, it needs some initial data to begin and start evaluation data due to that. At the beginning point a pool of unlabeled data is considered, and it is generated with the state space condition. Initial sample selection gives a prior knowledge of the system and makes further sample selection easier for SVM. In fact, after random selection we need more accurate selection of data labeling to increase its efficiency, reduce computational analysis, and drop the cost of evaluation. So every step in the training process generates a set of unlabeled candidate states from the state space

Each candidate sample is then given a measure called AL score. This score reflects the priority a state holds to be labeled in the future. Classification uncertainty, proximity to the estimated safety boundary, and spatial diversity: classification uncertainty, nearness to the safety boundary, where uncertainty is high, exploration of unvisited area. In addition, a diversity criterion is introduced to reduce the overlapping samples. Therefore, when several candidate samples provide similar levels of information, preference is given to samples that are sufficiently separated from previously selected samples. This prevents the AL algorithm from repeatedly evaluating extremely similar states and improves the information obtained from each expensive MPC evaluation. The entropy, a measure that discussed in the subsection 4.4.1, is the first measure to evaluating the classifier uncertainty where it has high

potential to improve the learned decision boundary.

After selecting a wide concentration of essential data from critical area, boundary, the system needs to investigate some data a little far from the boundary to be certain about other points. When all needed data is collecting from the MPC evaluation, the system is free to investigate from other points to improve its prediction but it depends on iteration and computing cost and time.

Finally, the unlabeled samples are known as the most confident samples or they are samples that the system does not need to evaluate since they are very safe or unsafe either the system is investigated a similar data so to avoid wasting time and to reduce computational cost the system skip other ineffective data.

Through a mixture of uncertainty sampling, physical insight into the braking process and dataset touring mechanism the technique effectively focuses oracle evaluations around the point where a given operation region is no longer feasible and not yet safe to be entered. This process can reduce a large amount of information for investigation compared to full exhaustive exploration based on SVM evaluation number and iteration.

4.5 SVM Classifier

An SVM classifier is used because it is suitable for binary classification and can represent nonlinear decision boundaries through a kernel function. The purpose of the classifier is to predict a function that captures the correspondence between the system state and whether MPC of that system state has a solution. A data point in the dataset is characterized by its state vector-which stores the relative distance and relative velocity between the ego and obstacle and its label, which is determined by querying the oracle as either (+1) if the state is feasible, or (-1) if the state is not.

For a training data set containing N samples:

$$(x_i, y_i)_{i=1}^N, \quad y_i \in -1, +1, \quad (4.12)$$

An SVM tries to find the decision boundary that has largest separation margin between the two classes. The obtained classifier can then classify unseen operating conditions as belonging to the feasible or infeasible region. The threshold separating safe and unsafe states is often nonlinear in most of actual application like the emergency braking one presented in this paper. In that case a RBF kernel function is proposed to map the original states space into the new one higher dimensional with the aim to be linearly separable in the new one. We then use weighted training strategy so that to make features in unsafe zones to be well presented in the classifier. If a given condition has violated a certain constraint type by a value higher than a pre-defined threshold then corresponding sample will be up-weighted more, based on degree of violations revealed from slack variable provided in oracle evaluation.

```
mdl_svm = fitcsvm(X_labeled, Y, ...
    'KernelFunction','rbf', ...
    'KernelScale','auto', ...
    'BoxConstraint',7, ...
    'Standardize',true, ...
    'Weights',weights);
```

Hence, samples close to boundary of critical regions are contributed more for training. Once training has completed, posterior probability is obtained so as to transform SVM's output into class probabilities. AL algorithm uses these probabilities so as to quantify the classification certainty with the use of an entropic measure. Hence the SVM acts as a classifier and provides information on the uncertainty for selection of new sample. The final trained classifier depicts a representation of the approximation of the boundary of the feasibility region in state space. The feasible regions indicate conditions under which the MPC is able to regulate safe vehicle operation satisfying all specified constraints. An infeasible region denotes circumstances for which MPC is unable to guarantee that the vehicle will not stop before minimum distance or even collision will occur.

To sum up, by combining SVM classification with AL and MPC-based labeling, an accurate approximation of the safe reachable set can be obtained with significantly fewer oracle samples.

4.6 Controllable Set

The main objective of the proposed learning framework is to derive a reliable estimation of the safe reachable set of the emergency braking situation. By integrating MPC-based safety estimation, AL, and SVM classification, the system is expected to identify the boundary separation between safe and unsafe operating region within the considered state space.

In this work, states are defined by the relative distance and the ego and lead vehicle velocity. The aim of MPC-based oracle is to determine whether that ego vehicle can satisfy all safety constraints while avoiding collision. Afterward, the result will be labeled and be used to train the SVM classifier, which approximates the feasibility boundary over the entire operating region.

Next, the learned safe controllable set can also be interpreted as the set of all states from which the ego vehicle can successfully navigate to a safe state while fulfilling the constraints. The states within this region indicate safe operating states in which the controller can successfully avoid collision whereas the states outside the learned safe region pose safety risks due to the inability of the controller to manage them safely. Mathematically, the learned safe controllable set can be represented as

$$\hat{\mathcal{R}}_T = \{x \in \mathbb{X} : \hat{f}(x) = +1\}, \quad (4.13)$$

where $(\hat{f}(x))$ denotes the classification function that is obtained from the trained

SVM model. The classifier provides an estimation of whether a given state belongs to the feasible or infeasible region based on the information acquired during the AL process. The derived safety boundary almost coincides with the boundary of transition between collision-free and collision-reachable operating modes. States with a large relative distance and a small relative speed were nearly always classified as feasible states, while those states with a small relative distance and a high relative velocity were nearly always as infeasible states. Another significant feature of our approach is that the discovered boundary depends on only a small number of evaluations of the oracle. AL helps to put sampling to where the region is highly uncertain, around the real physical stopping-distance boundary. In this way the controllable set can be estimated with dramatically reduced costs yet a fairly high accuracy. The identified safe set can be effectively used in decision-making support as well as in safety analysis of autonomous driving vehicles. At a measured vehicle state the classifier outputs quickly an assessment whether it is safe to operate there without solving the MPC problem again. Thus, this approach permits the realization of efficient online safety analysis as it yields a computation-light substitute for the feasible set.

4.7 Model Accuracy and Performance Evaluation

The main motivation behind this proposed AL approach is to be able to correctly predict the controllable set of the vehicle while minimizing the number of expensive MPC oracle evaluations. Consequently, the performance of the learned model, final model, must be assessed from both a classification and a safety perspective.

In the traditional use case of ML only the prediction accuracy is evaluated. Safety-critical applications need to ensure not only to distinguish in the region of infeasible and feasible, but a correctly estimated classifier decision boundary (between feasible and infeasible region) is crucial since errors nearby the boundary may lead to incorrect safety evaluations and thus unsafe decisions. Therefore, evaluation should not only consider accuracy, but the safety estimation by a learned classifier is important.

As it is shown in Eq (4.1) the labeled data-set generated by the MPC oracle is divided into two classes, safe(+1) and unsafe(-1). Where feasibility label is obtained by solving the MPC optimization problem. Afterwards, the SVM classifier is trained with the selected active samples, and their predicted values are checked against their oracle labels. The overall classification accuracy is computed as:

$$\text{Accuracy} = \frac{N_{\text{inv}}}{N_{\text{total}}} \times 100 \quad (4.14)$$

where N_{inv} shows the number of investigation classified samples, correctly classified sampled, and N_{total} represents the total number of evaluated samples.

4.7.1 Safety Boundary Approximation

Since focusing of AL is to select samples around regions with high uncertainty, a large portion of the collected data is situated near the transition between feasible and infeasible region. As a result, the SVM can accurately approximate the safety boundary generated by the MPC oracle. The quality of the learned boundary can be visually evaluated by comparing:

- The analytical stopping-distance boundary
- The MPC-generated feasibility boundary
- The SVM classification boundary

This close correlation shows the classifier succeeds at identifying the intrinsic safety property of the system. For the considered emergency braking scenario, where worst-case scenario is considered, the learned boundary is expected to have a quadratic relationship between vehicle speed and required stopping distance, this quadratic relationship comes from equation of motion and stationary lead vehicle.

$$D_{\text{safe}} = d_{\min} + \frac{v_{\text{rel}}^2}{2|a_{\text{rel}}|} \quad (4.15)$$

Where $d_{\min}$ represents the minimum allowable safety distance that is defined by system, v_{rel} is the relative speed between ego and lead vehicle, and a_{rel} shows relative absolute acceleration between two vehicles. In our case, since in worst-case scenario the lead vehicle is stationary so v_{rel} and a_{rel} show ego vehicle speed and ego vehicle acceleration respectively.

The learned SVM boundary will be obtained from the actively selected samples closely follows this relationship and indicating that the classifier successfully captures the underlying safety characteristics of the system. Agreement between the analytical boundary, the MPC-generated boundary, and the learned SVM boundary demonstrates the effectiveness of proposed Active Learning approach.

4.7.2 Reduction of Oracle Evaluations

The obtained SVM boundary from the active set matches well the underlying relationship and shows that classifier grasps safety properties of the system. Close agreements between boundaries shows the efficiency of our proposed AL framework. Rather than interrogating each state in the search space the method instead selectively queries samples with a high level of uncertainty. This approach is likely to cluster useful samples at the feasibility frontier, where additional information most significantly benefits the classifier. This strategy dramatically reduces the number of required oracle to evaluate while maintaining a high estimation accuracy of the controllable set.

5

Results

This chapter presents the results obtained from the proposed MPC-based in AL framework for autonomous vehicle at different scenarios. The objective of this work is to identify the safe controllable set of the vehicle and accurately determine the boundary separating feasible and infeasible operating conditions. The method of this thesis is a combination of MPC with a slack-variable-based safety, AL framework and SVM classification. First, an MPC controller computes the feasibility of vehicle states and assigns safety labels to these states. The plant model is subsequently introduced to represent additional effects encountered in realistic driving conditions. AL strategy selects the training data, by employing Shannon-entropy and SVM sample selection, near the safety boundary that has the highest expected influence on the approximation of the safe controllable set based on MPC Oracle safety. The obtained results demonstrate how the proposed framework is able to estimate safety-critical regions accurately while significantly reducing the number of expensive MPC evaluations and also reduce its cost of computation.

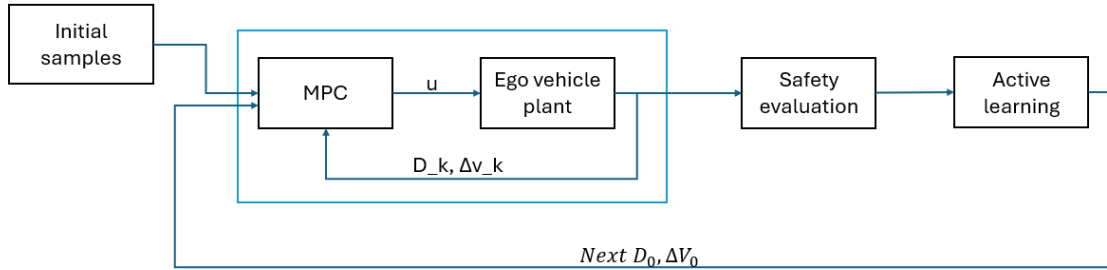


Figure 5.1: Overview of the evaluated MPC-Active Learning framework.

5.1 Curve fitting using Sparse Regression method SINDy

For the parameters considered below,

$$a_E = -8 \text{ m/s}^2, \quad a_L = -5 \text{ m/s}^2, \quad d_s = 5 \text{ m},$$

the relative braking capability becomes

$$|a_E - a_L| = 3 \text{ m/s}^2.$$

The analytical safety function is therefore

$$h(D, \Delta v) = D - 5 - \frac{\Delta v^2}{6}. \quad (5.1)$$

Using the polynomial candidate library, SINDy identified the following sparse equation:

$$\boxed{h(D, \Delta v) = -5.0000 + 1.0000D - 0.1667\Delta v^2} \quad (5.2)$$

while the coefficients associated with

$$\Delta v, \quad D^2, \quad D\Delta v$$

were eliminated by the sparse regression procedure.

Since

$$0.1667 \approx \frac{1}{6},$$

the identified equation can be written as

$$h(D, \Delta v) \approx D - 5 - \frac{\Delta v^2}{6}. \quad (5.3)$$

Setting the safety function equal to zero gives the identified boundary,

$$h(D, \Delta v) = 0, \quad (5.4)$$

and therefore

$$\boxed{D = 5 + \frac{\Delta v^2}{6}}. \quad (5.5)$$

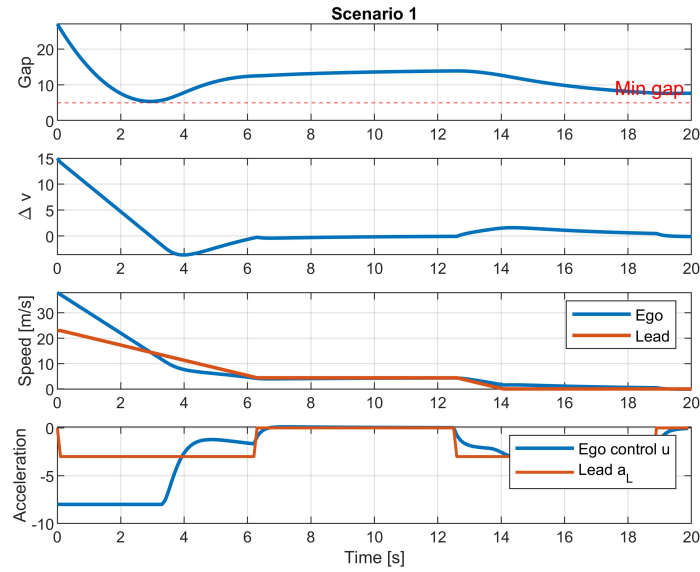
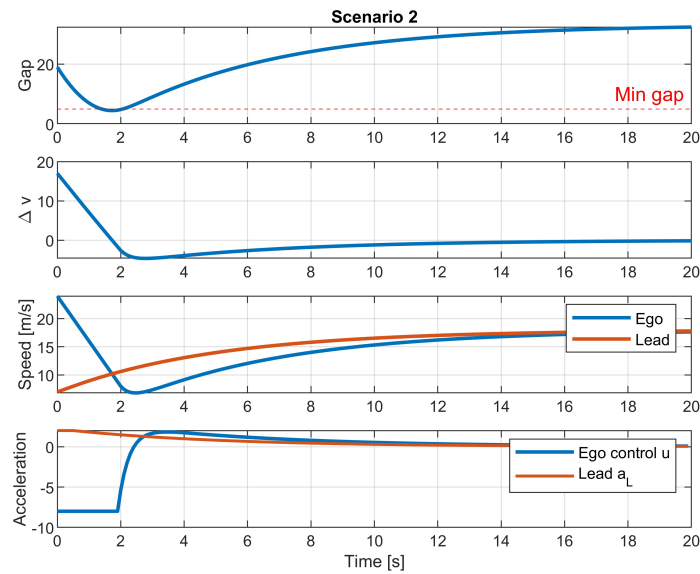
This is the same relationship obtained from the analytical stopping-distance calculation.

5.2 MPC Performance

The MPC response was evaluated using several longitudinal driving scenarios with different initial ego-vehicle speeds, lead-vehicle speeds, and inter-vehicle distances. Because the MPC serves as the feasibility oracle for the subsequent AL framework, its closed-loop performance is evaluated first. Several different scenarios are designed to evaluate MPC performances and they are shown in Appendix A while four representative cases illustrate here: three collision-free cases and one worst-case condition in which collision cannot be avoided under the imposed constraints. The corresponding initial conditions are listed in Table 5.1. The acceleration limits used in these simulations are defined in the Methodology chapter and are not repeated here.

Table 5.1: Initial conditions for four MPC evaluation scenarios

	Ego vehicle speed m/s	Lead vehicle speed m/s	Relative distance m
Scenario 1	38	23	27
Scenario 2	24	7	19
Scenario 3	27	14	36
Worst-case scenario	40	0	100

**Figure 5.2:** MPC response in Scenario 1 with an initial ego-vehicle speed of 38 m/s, lead-vehicle speed of 23 m/s, and relative distance of 27 m.**Figure 5.3:** MPC response in Scenario 2 with an initial ego-vehicle speed of 24 m/s, lead-vehicle speed of 7 m/s, and relative distance of 19 m.

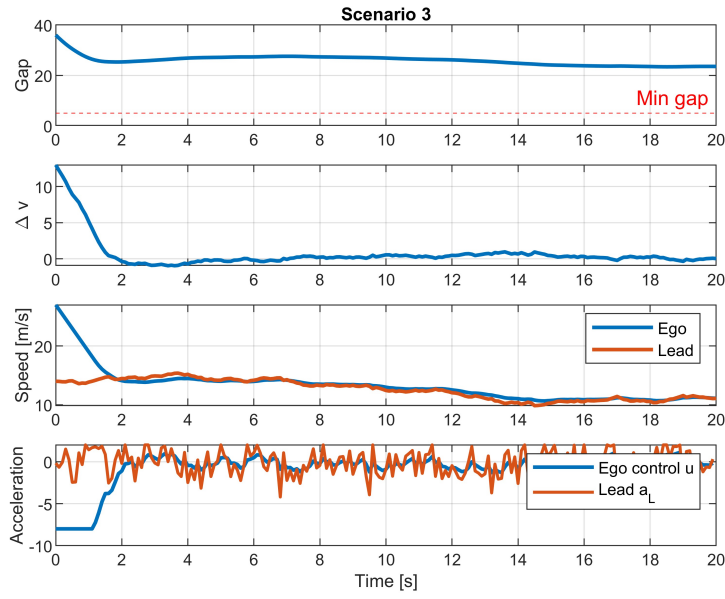


Figure 5.4: MPC response in Scenario 3 with an initial ego-vehicle speed of 27 m/s, lead-vehicle speed of 14 m/s, and relative distance of 36 m.

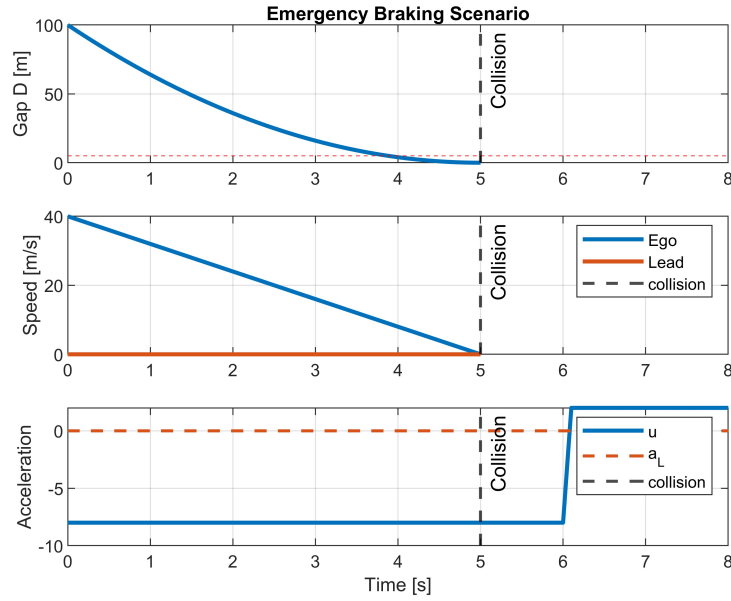


Figure 5.5: MPC response in the worst-case scenario with an initial ego-vehicle speed of 40 m/s, a stationary lead vehicle, and an initial relative distance of 100 m.

As shown in the figures, figures for evaluating MPC controller performance (all figures except the worst-case scenario), the MPC controller successfully avoids collisions while maintaining the ego vehicle within the safe region. Although in Fig. 5.2, 5.3, 5.4 collision did not occur, Figure 5.5 shows that a collision occurs in the worst-case scenario.

5.3 Slack Variable Analysis and Learning Trigger

The slack variable makes a great significance in the proposed AL approach since it indicates how severely constraints are violated, rather than just providing a feasible or infeasible decision on safety of current state. Even though the MPC-based oracle will eventually label the current state safe or unsafe, slack can identify the system's closeness to violation of safety constraint and highlight points that require close observation in operating environment.

Physically, the slack variable would represent the risk level. when it is zero or near zero, it means all safety constraints are fulfilled and the vehicle can keep all safety distance within the prediction horizon. A positive slack value shows that the controller can't fulfill all the requested safety constraints perfectly. A positive slack value indicates that the nominal safety constraint cannot be satisfied exactly under the available control authority.

Because of this, the slack variable is penalized in the MPC objective function with a very large multiplier. As such, the optimizer will try its best to keep constraint violations as small as possible and only introduces slack if a truly feasible solution cannot be found. This therefore makes a useful slack as a measurement of how close we are on the border between a safe and unsafe area of operation.

Figure 5.6 illustrates the distribution of slack values samples are selected based on the method mentioned earlier.

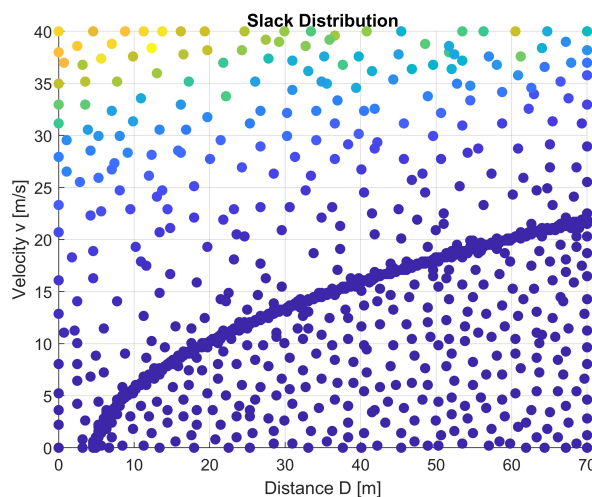


Figure 5.6: Slack distribution of the model

Fig. 5.6 is illustrated how samples are selected and then how slack is introduced to them in MPC cost function. Then the following picture, Fig. 5.7, shows how this slack is divided into feasible and infeasible for our system based on AL, Shannon entropy and SVM sample selection.

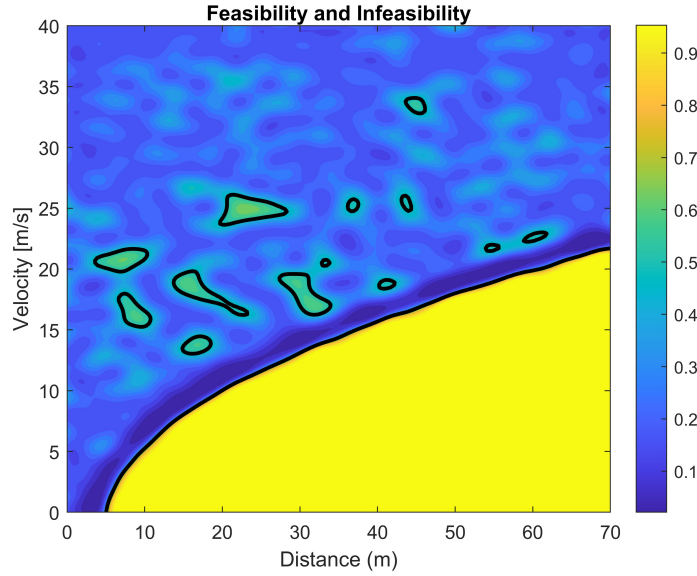


Figure 5.7: A sample scenario to show feasibility and infeasibility, where yellow area shows feasibility and blue shows infeasibility

As it is clear in Fig. 5.7 this slack defined the feasibility and infeasibility of our system. Feasible, yellow area in the picture, is an area where the system can guarantee safety while infeasible, blue area above the separation line in the picture, is an area where the system does not promise safety and a collision may occur.

Besides in optimization, slack has played other important roles too, in the suggested learning scheme, slack also acts as a trigger for AL strategy. Instead of performing learning query for each operating point and updating the learning process constantly, this AL approach triggers the learning at those potentially critical points.

During simulation, the accumulated slack value is monitored at all time step of our process. When the slack exceeds a predefined threshold, the corresponding state should be considered informative and be selected for further analysis. States with negligible slack values are generally located deep inside the feasible region where the system behavior is already well understand. However, states with positive slack values, quite close to threshold, typically are very close to the transition boundary separating feasible and infeasible operating conditions.

Thus, we can consider the learning trigger as an event driven based mechanism which would bring more computational load to handle the uncertainties. When the car is on the safe regions, away from unsafe regions, there would be no requirement of further learning; whereas, on traversing close to the border, there would be large slack and hence the learning starts in a region when it is hardest to identify the decision surface.

Furthermore, this method also possesses robustness to numerical noise and minor errors in the optimization. Indeed, slack values slightly greater than zero will usually

appear when implementing the constraints due to tolerances in the solver and do not reflect real safety constraints. To filter such negligible difference, a threshold is introduced which prevents such differences to be recorded as an event to trigger the learning.

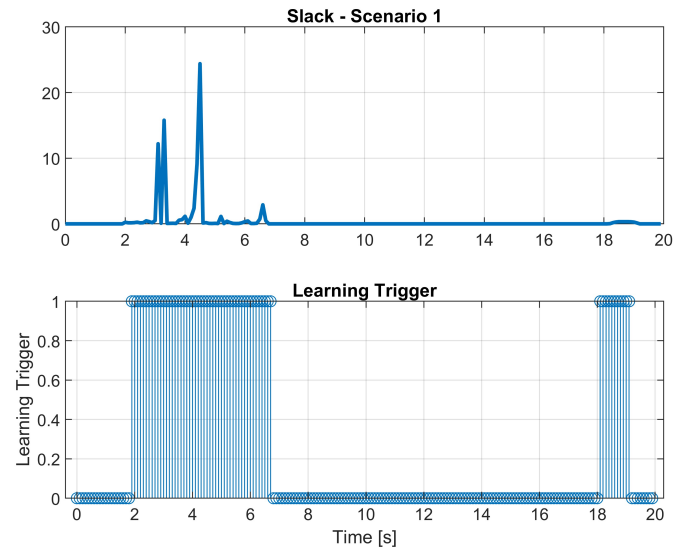


Figure 5.8: Learning trigger activation and Slack variable in the first scenario

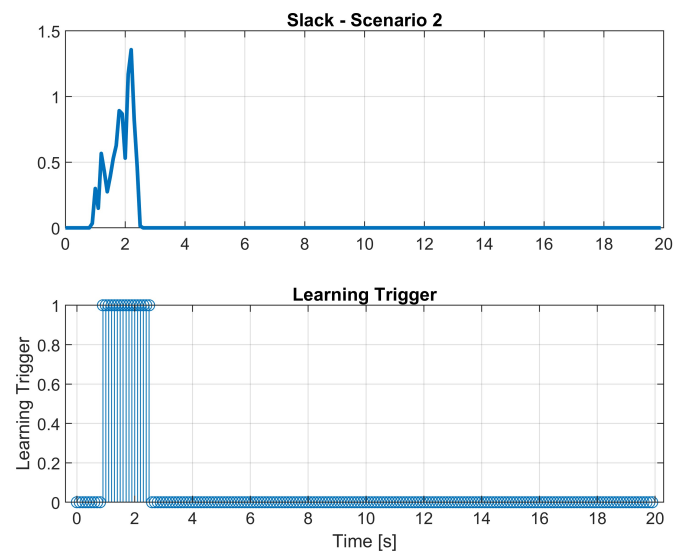


Figure 5.9: Learning trigger activation and Slack variable in the second scenario

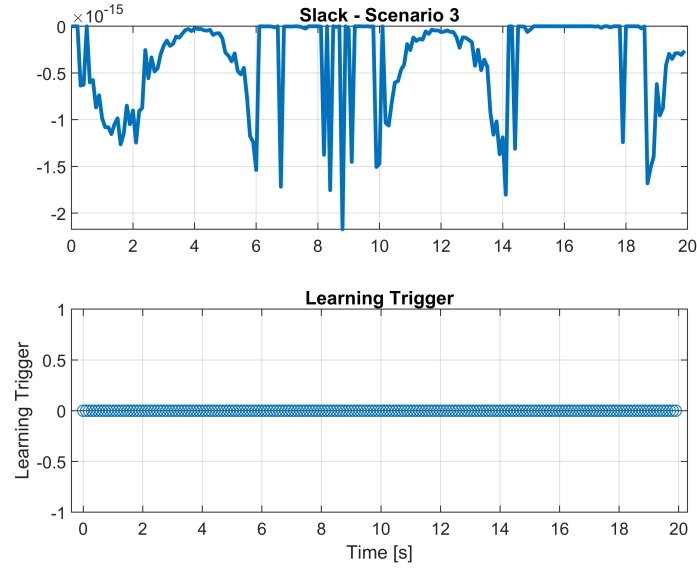


Figure 5.10: Learning trigger activation and Slack variable in the third scenario

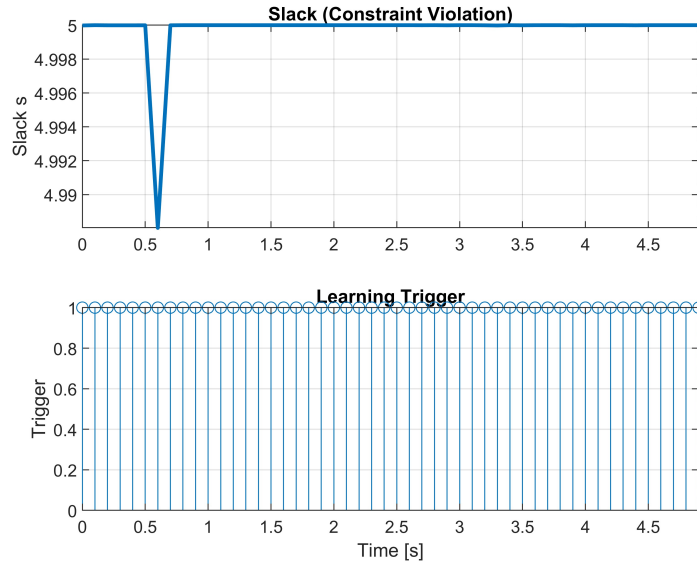


Figure 5.11: Worst-case scenario slack variable and learning trigger activation

These pictures are shown how slack varies in different situations for different scenarios. Increasing slack indicates that the system is approaching or violating the imposed safety constraint, causing the controller to apply stronger corrective action.

For example in Fig. 5.8 it is illustrated that the system is making alert in the beginning, almost from 2 to 7 second, and at the end of the process near 19 second, and when look back at Fig. 5.2, it is quite understandable why those areas are important to learn. However, in worst-case scenario, Fig. 5.11, the learning trigger shows it is essential to learn from beginning to end and it means that our system identifies

critical situation and predict an accident might be happened in the future and shows the vehicle is in an unsafe region. So the answer is in related to its picture, Fig. 5.5. Although the initial relative distance in the beginning is about 100 meters but it cannot satisfied the system by that initial relative distance.

Analyzing the distribution of slack values can also shed light on the overall shape of the safe controllable set. Operating conditions where the slack is small correspond to the safely controllable states; areas with large slack will mostly lie close to the collision boundary where evading are difficult. Slack value will then be exploited at a later step to increase the weights of safety-critical samples that are near the safety boundary.

All in all, the slack variable acts on two different functions of the introduced approach. On the one hand it eases the constraints in the unsafe zone for safer operation and in particularly unsafe scenarios maintaining feasibility of the optimization, and on the other hand it captures an estimation for the overall uncertainty in the state and consequently supports an efficient event triggered AL for efficient computation only where it truly matters for performance and safety.

The first purpose of the thesis is to recognize critical situation. For this aim MPC should be testes for different scenarios to know the situation and act based on that. MPC performance is shown in pictures above, 5.2 to 5.5. The scenario results indicate that the MPC responds appropriately to changes in the relative distance and velocity while attempting to maintain the safety constraints. The corresponding slack and trigger signals identify the periods in which the operating condition becomes safety-critical and additional AL evaluation is activated.

5.4 AL Result

5.4.1 Sample Selection

A second objective of AL in this thesis is to reduce the computational cost of estimating the feasible region. Instead of exploring all possible states of the operating domain and query our (expensive) MPC oracle, our approach focuses only on a subset of highly informative points and with their evaluation allows the classifier to reconstruct the boundary of the safe reachable set much faster.

The learning process begins with an initial set of labeled samples generated from all points of data. These samples provide a preliminary representation of the transition region between feasible and infeasible operating conditions and they are a starter point to recognize each point condition better. Each sampled state is evaluated using the MPC-based oracle and assigned a label according to its feasibility. Safe states are feasible, whereas states for which the controller cannot guarantee collision avoidance are labeled as infeasible.

5.4.2 Generalization Capability

Once the training is finished, the SVM provides a continuous approximation of the region feasible on the region of state space examined. Furthermore, as opposed to the MPC oracle which must resolve an optimization problem every time a new operating condition is encountered, a trained classifier may very easily determine whether a certain state is feasible or not.

Therefore, the learned model acts as a surrogate representation of the MPC oracle and shows rapid safety assessment for previously unseen states. This capability would be attractive for contemporary autonomous driving applications, where fast decision-making and low computational complexity are essential. The experimental results have confirmed that the introduced AL framework is able to obtain an estimated value of the safe controllable set using quite appropriate labeled samples.

The AL framework selectively explores the unlabeled space close to the boundary region, and uses the oracle evaluations on this particular area to reach an accurate classification. At the same time, the expense for the generation of labeled data has been greatly decreased. Finally, the learned SVM boundary follows the MPC that is generated safety boundary and reproduce the expected relationship between vehicle velocity and required stopping distance. Consequently, the proposed approach provides an efficient, and computationally inexpensive method of the safe controllable set that can be used for safety assessment, monitoring, and decision support in autonomous driving systems or systems in high level autonomous.

In this work the uncertainty is measured using Shannon entropy, Sec. 4.4.1, as the sampling metric, where the classifier outputs shows each states probability and states with probability near 0.5 have the highest uncertainty, so they are more informative in evaluation and identifying boundary between safe and unsafe region[12].

However, entropy alone is not sufficient to explore state-space efficiently. If sampling only focused on the entropy criteria, large regions would be selected at each iteration of sampled around the same center location-information overlap would be poor. A measure of diversity is therefore incorporated by considering the minimum Euclidean distance between the candidate sampled point and the labeled data space. It is the element that helps exploration in different parts of the operational space, by guiding selection of a new region in which an unvisited state can be evaluated.

Finally, vehicle dynamics are utilized in the sampling scheme, via the implementation of a physics-guided boundary metric. Given that the emergency braking problem has an approximately analytical relationship between stopping distance, states that are in close proximity to this boundary are more beneficial. The sampling scheme prioritizes samples that satisfy both uncertainty, variety and closeness to the boundary value.

AL score this value represents a ranking of candidates by combining two values mentioned in earlier points, such as the confidence estimate plus the distance to the

learned safety boundary to evaluate its “information value” to the classifier. In each training round, the candidate states obtaining highest AL scores are then chosen and evaluated via the MPC Oracle. Newly verified states are added to training set and the classifier is updated again by means of being re-trained

Combining these criteria in a ranking the combined score for AL represents both the predictive model’s belief and the potential for the state to be closer to boundary in terms of the decision threshold. In each iteration, the states that achieve the top scores are queried on the MPC oracle to obtain newly labeled data points, and this data is augmented into the existing training dataset to retrain the classifier. The safety threshold progressively converges towards its real location with repeated iterations.

The result of the applied sampling scheme can be viewed in the selected sample distribution. The majority of queries lie, as it can be expected, within the transition region where the system switches between safe and unsafe modes. This is desirable since the boundary carries with itself the highest amount of information content of the reachable set in our scenario. Instead regions well within the safe/unsafe regions give little new information and need less samples.

This additional concept to classifier training is that we are utilizing slack-variable information in the classifier’s training stage as well. We place higher importance on samples that have larger slack-variables because it represents a more critical constraint violation condition and train it weighted manner, improving accurate prediction of high criticality conditions and boundary estimation for collision vicinity cases.

Fig. 5.6 shows how data is selected and accumulation of data is investigated in boundary shows how much important is that area and high uncertainty is occurred in that curve.

Overall, our results reveal the efficiency of the proposed AL framework to significantly decrease the number of oracle calls needed to approximate the safe controllable set. By integrating entropy based uncertainty sampling, diverse exploration based approaches, and the information from physics-guided boundary information, the framework smartly focuses computational resources in the informative areas of the state space and provide a good approximation to the safety boundary with drastically less computational overhead than usual sampling methods.

5.5 SVM Classification Results

As it mentioned before, Sec. 4.5, SVM is utilized for collecting informative data near safe boundary operating generated by the MPC-based oracle. Training is done on the collected labeled samples from the AL process, in which the SVM will generalize the mapping from the states to their safety status (unsafe/safe). This allows us to quickly estimate safe state space without solving the MPC on-line for any incoming states.

In this work, the classifier is trained with relative distance and the ego and lead vehicles velocities from the longitudinal vehicle model and each sample will be labeled based on AL evaluation.

For the classifier and AL evaluation a nonlinear RBF kernel is chosen as the limit between the region which feasibility and infeasibility are not linear, generally the two classes' limit is a nonlinear boundary. Even though, theoretical stopping-distance limit has a quadratic term, MPC's constraints, finite prediction horizon's influence and penalty associated with slack's variable force the feasible region boundaries a more nonlinear way. The RBF kernel is suitable for representing this nonlinear decision boundary.

To improve classification accuracy, weighted training is used. This involves down-playing small slack values and up-emphasizing slack values in the optimization procedure, due to the higher operating relevance or safety constraint violation magnitude for points having large slack values. The effect of down-emphasizing smaller slacks causes the classifier to place higher emphasis on accurate correct classification in proximity of the safety constraint.

Figs. 5.12 and 5.13 show the result of classification map, where we have the safe area, in green points or below yellow curve, the classification is +1 and in contrast in red points and above the yellow curve the classification is -1.

At the conclusion of training, the SVM has learned a decision function that discriminates between safe and unsafe state-space samples, and produces the classification map is depicted. feasible samples are treated as positive class examples while infeasible samples serve as negative class examples. The resulting decision boundary separates the two regions cleanly and accurately approximates the predicted stopping-distance relation as shown.

The difference between the analytical and learned boundaries arises because the plant model includes effects that are not present in the simplified stopping-distance calculation, including reaction delay, road grade, aerodynamic drag, and actuator dynamics.

The key information from the outputs is that the highest density of labeled samples occurs on or close to the boundary, and that defines how the samples are classified. This output is an example of the way the entropy-based method of AL can extract maximum benefit from the data; because it chooses for examination the most uncertain points (which also give the greatest new information to the learning process) we are learning an increasingly accurate boundary definition with each step.

The extension to SVM makes the results even more explainable. With the posterior probability estimation, not only does it identify whether or not the state is safe and unsafe, but it has also assigned a measure of confidence with it. The points which

lie at a great distance from the SVM decision boundary will usually yield the posteriors to close to either 0 and 1, and those near the decision boundary will output 0.5.

Alternatively, we can represent the learned decision boundary with a quadratic polynomial which would make it possible to interpret the classifier and compare the learned bound to the analytical stopping-distance equation. The picture shown in below, Fig. 5.13, shows a fairly accurate correspondence to the MPC labeled stops as well as the theoretical braking distance curve and thus verifies that the classifier has learned the relevant system dynamics of the car's safety behavior.

Overall, the SVM results indicate that the AL framework can approximate the feasibility boundary using a reduced set of MPC-labelled samples. If the SVM classification experiments indicated the effectiveness of the developed learning framework for effectively approximating the safe controllable set with a relatively small amount of oracle calls. The combination of MPC-induced labels, slack-weighted training and AL based sample selection successfully answer this aim and illustrates AL result is correct. In simple terms, it efficiently learn the safety boundary without sacrificing classification accuracy, which confirms their potential for being used as a fast learning surrogate models for feasibility analysis and safety evaluation.

5.6 Learned Safe Controllable Set

The main purpose of the framework is to develop an approximation for the safe controllable set of the emergency braking maneuver. The developed controllable set captures all possible safe states from which the emergency can be avoided with a reduced complexity than the optimization of a full MPC controller at each state.

To investigate controllable set area, the worst-case scenario is designed to collect boundary separation line and estimate the AL framework performance by investigating the combination of relative distance and relative velocity. Eventually, the safe controllable set can be expressed on a 2-D space which consists of distance and velocity.

Safe controllable set contains all those initial states for which the MPC controller can find a feasible sequence of control inputs such that collision is avoided subject to the system dynamics, acceleration constraints, and safety constraints. All states that lie outside the safe controllable set correspond to those situations where the MPC-based feasibility criterion is violated. Also, it should be mentioned we consider 5 meter for safety distance and it means the ego vehicle should stop in 5 meter gap with the lead vehicle.

First, a set of sampled states are assessed using the MPC-based oracle to determine whether each state is either feasible or infeasible (based on the outcome of the optimization). Following this, plant parameters are introduced to make our model more realistic. Now, the final ego vehicle acceleration is the combination of MPC and plant factors. Then, AL should investigate the model base on Entropy and sample

selection to identifies boundary between safe and unsafe region to determine the safe vehicle velocity related to instantaneous relative distance.

The shape of the learned boundary should be very similar to the real world stopping distance. It shows the relationship between distance and velocity(since braking distance needed to decelerate the car and get it to stop increases quadratically with velocity). This conforms to the theory of the vehicle motion and validates the ability of the proposed learning framework in characterizing the physics of emergency braking.

However it's important that the learnt boundary is obtained directly from the MPC evaluations rather than from an imposed analytical expression with the effects of prediction horizon, actuator limits, optimizing limits and control behavior all included in the learning set of the controllable region, as they affect the resulting boundary of stopping distance.

One major benefit of this approach for safety is the efficiency after classification training; for each safety assessment the process only needs a fast classification and no repeated solving of an optimization problem, providing real-time feasibility for online real-time decisions and applications.

The learned boundary and the analytically derived stopping boundary in 5.12 shows a good agreement across the operational range. As expected, small deviations exist close to extreme conditions, attributed to constraint interactions perceived by the MPC controller and to finite prediction horizon effects. Nevertheless, the overall consistency between the two boundaries validates both the MPC formulation and the AL approach. In addition, the gap between learned boundary and stopping boundary from equation of motion, Eq 4.15, is increased in next picture, Fig 5.13. The answer behind it is plant's parameters and important reason comes from considering reaction time.

This trained safe controllable set acts as a compact description of the vehicle's safety envelope. It identifies the maximum controllable and allowed values of distances and velocity for collision avoiding and acts as a handy tool to monitor safety during maneuvers, evaluate risks and design future automated driving.

In conclusion, the developed strategy demonstrates effective safe controllable set estimation under sample-scarce conditions using MPC with slack variable learning triggers, AL, and SVM classification. The learnt boundary is a reliable surrogate of safe/unsafe region boundaries and can be employed for verification of longitudinal vehicle control system safety.

5.7 Controllable Sets For Safe and Stable Automated vehicles

An important goal of the developed framework is to estimate both the safe controllable set and also to learn a good approximation of the boundary between the feasible and infeasible regions of the operating space. This is because the learned boundary replaces costly MPC evaluations in the real-time safety checking and thus its precision determines how trustworthy the entire framework is.

The results of the developed method are assessed by comparing the labels produced by the oracle, the MPC-based approach, with the labels generated by the learned SVM classifier. Since each label is a solution of a constrained optimization problem, that accounts for the vehicle dynamics, the braking constraints, the safety requirements and the prediction horizon, the oracle effectively provides the ground truth. Hence, the agreement between the predictions of the SVM model and the labels of the oracle represents an actual evaluation measure of the learned model.

As expected based on AL, results prove that the presented AL framework accurately characterizes the structure of the safe and unsafe regions of the explored state-space. The states which are far apart from the boundary between regions belonging to different classes are almost correctly classified; it reveals that the classifier is able to understand what distinguishes the operating conditions considered feasible to the infeasible ones. The highest uncertainties happen around the change region from safe to unsafe as is anticipated.

Additional support in high confidence classifier predictions are seen far from the decision boundary and uncertainty of the classifier is clustered along the boundary between safe and unsafe operation, again confirming the decision boundary appropriately captured the critical region of operation.

In summary, the present results indicate that the proposed AL and SVM approach can recover the safe boundary of MPC efficiently and achieve much less cost on the oracle evaluations than an exhaustive search approach. Hence, the learned model can serve as a efficient computational substitute to evaluate the safety or controllable sets of autonomous driving system.

The following picture, Fig. 5.12, shows initial data selection and SVM selection after MPC controller in worst-case scenario where it becomes very important for us to evaluate MPC performance for this thesis. Black line comes from the real equation of motion, Eq. 4.15, while yellow line comes from our MPC investigation. A difference is observed between the two curves and it comes from prediction horizon number, sampling time, and our MPC cost function coefficient. The results significantly illustrate that the proposed system performs effectively, while AL identifies the controllable region and the boundary separating the safe and unsafe regions using Shannon entropy and SVM classification.

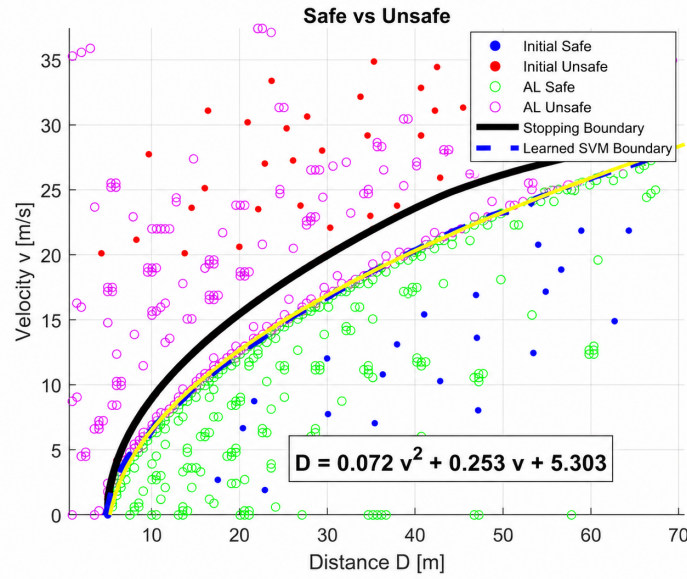


Figure 5.12: MPC controller performance and AL trajectory by initial and SVM sample selection

Our system estimated curve is equal to $d = 0.072v^2 + 0.253v + 5.303$. This is the best curve line, yellow curve, that our system could match to the separation of safe and unsafe area based on our sample selection and estimated blue dash line. Coefficient 5.303 comes from mismatches in the beginning and the fitted intercept differs slightly from the nominal safety distance because the learned boundary reflects the complete MPC and fitting procedure. Because blue-dash line would be able to recognize safe region from unsafe and it is obvious in the picture.

While the following picture, Fig 5.13, depicts the data selection and investigation after plant where other parameters is inserted and our model is very close to real situation of driving condition.

After adding other mentioned parameters in plant, Sec. 3.5, the gap between black and yellow lines became greater and it demonstrates the system has become more cautious, because the real stopping boundary comes from equation of motion, Eq. 4.15, where we do not have any realistic disturbances while when we considered the worst-case scenario in our model, and after adding all parameters and coefficients, the region between realistic stopping point and our model became a little more and the most important reason is reaction time. The reaction delay reduces the time available for braking and therefore increases the distance required to maintain the specified safety margin. Thus for example for 30 meter relative distance between two vehicles the system says the ego vehicle should have about 14m/s velocity instead of 20m/s.

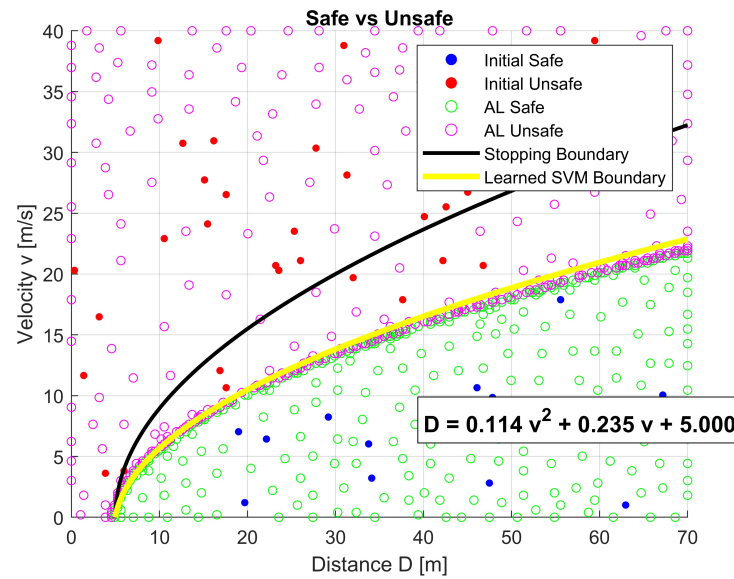


Figure 5.13: AL trajectory based on initial and SVM sample selection after introducing plant

To conclude, in the first place AL is employed to reduce the computational cost and time of MPC evaluation to identify the controllable safe boundary in order to maintain the vehicle in feasible region. This investigation is done by updating system states, including relative distance and relative velocity, as well as applied control input, a_e , at each step and evaluating individual state by MPC. Then AL utilized this information to identify boundary and prevent vehicle from exceeding velocity compare to relative distance. Figure 5.13 indicates that the combined MPC–AL framework identifies a consistent boundary between feasible and infeasible operating conditions.

Furthermore, figures in section 5.3 show the slack variable is added appropriately and the system can recognize the critical situation and decides when additional learning is necessary.

6

Conclusion

This chapter provides a summary of the key results obtained in the thesis and critically evaluates the suitability of the method for the safety assessment of autonomous vehicles. Simulation results indicated the effectiveness of the proposed methodology to delineate the border between infeasible and feasible operating points with a noticeable reduction in the computational cost through the selective evaluation of informative operating conditions. Finally, the chapter concludes by highlighting several limitations of the developed approach and suggesting some potential future work for improvements.

6.1 Equation identification with Sparse Regression

SINDy was investigated as a sparse equation identification method for representing the longitudinal safety boundary. A second-order polynomial library containing constant, linear, quadratic, and interaction terms was provided to the sparse regression algorithm.

For the simplified longitudinal braking case, SINDy retained only the constant term, the distance D , and the quadratic relative velocity term Δv^2 .

The result shows that sparse equation identification can produce a compact and interpretable representation of a known safety boundary while eliminating unnecessary candidate terms. In the present study, this serves primarily as a proof of concept because the training target is derived from the analytical safety equation itself. The more significant potential of the method lies in extending the same idea to higher-fidelity vehicle models, where safety boundaries may be obtained from simulation or experimental data but may not have a simple closed-form analytical expression.

6.2 MPC Feasible Set Approximation

The AL framework provides an efficient approach for evaluating the safety boundary while limiting the number of computationally expensive MPC evaluations. Instead of labeling a large number of states uniformly across the entire state space, the algorithm attempts to identify and evaluate the states that provide the most useful information for refining the current estimate of the boundary. This method can

substantially reduce the number of required labeled samples and, consequently, the computational time associated with repeated MPC evaluations.

The final objective of this approach depends strongly on the quality of the selected samples. Increasing the number of training samples can improve the representation of the feasible and infeasible area, particularly when the initial dataset is insufficient to describe the boundary accurately. However, increasing the dataset also increases the computational cost of classifier training and prediction. More importantly, adding samples does not necessarily result in further improvement in classification accuracy. Once the classifier has already learned the regions that are clearly feasible or infeasible, extra samples provide relatively little information.

The most informative samples are near the transition between feasible and infeasible states, where classifier uncertainty is highest. In contrast, states far inside either region obviously have high confidence and provide limited additional information. Therefore, AL concentrates computational effort on the boundary rather than maximizing the total number of samples.

The SVM, kernel function, and Shannon entropy work together in this process. The SVM distinguishes feasible and infeasible states, while the kernel represents the nonlinear safety boundary. Shannon entropy quantifies prediction uncertainty and guides the selection of the most informative states for further MPC evaluation. The resulting learned boundary shows a smooth curvature and closely follows the reference boundary obtained from the vehicle dynamics and stopping-distance relationship.

A Python implementation was also developed for comparison, particularly for cases in which the initial states are unknown. This provides an additional consistency check for the vehicle model, MPC formulation, optimization, sample selection, SVM, and entropy-based uncertainty estimation

However, the performance of AL depends on the initial samples, classifier configuration, uncertainty measure, sampling strategy, MPC-based oracle, and for this thesis Plant . Increasing the number of samples beyond the informative boundary region can lead to diminishing returns while increasing computational cost. Overall, MPC provides the feasibility evaluation, SVM learns the nonlinear boundary, the kernel represents the shape of boundary, and Shannon entropy identifies informative samples, allowing the framework to approximate the safety boundary efficiently.

The main contribution of this thesis is the development of the data-driven framework for approximating the safety boundary for an autonomous emergency braking system by integrating MPC and AL. The main contribution is the integration of MPC-based feasibility evaluation with AL-based sample selection to approximate the safety boundary with fewer expensive oracle evaluations.

6.3 MPC Performance and Sensitivity to Prediction Settings

The performance of the MPC controller depends on several key components, such as prediction horizon, sampling time, system matrices (equation 2.23), constraints, and cost function (equation 3.8). In this thesis, a sampling time of 0.1 s and a prediction horizon of $N=50$ were used. The sampling time determines how frequently the controller updates its state estimate and control action. A larger sampling time results in less frequent updates, and therefore provides less frequent information about the future evolution of the vehicle, which may reduce the controller's ability to respond instantly to rapid changing situations.

The prediction horizon determines how far into the future the MPC controller predicts the vehicle behavior. Increasing the prediction horizon allows the controller to consider a longer portion of the future trajectory and can enable earlier responses to approaching safety constraints. In contrast, a shorter horizon provides less information about future vehicle, and may cause the controller to react later to developing critical situations. However, increasing the horizon also increases the computational burden of the optimization problem. Therefore, the horizon represents a trade-off between prediction capability and computational cost.

The different scenarios considered in thesis are presented in Appendix A. The controller avoids collisions in the collision-free test scenarios, while the deliberately constructed worst-case scenario is correctly identified as infeasible. These results demonstrate the role of MPC as the model-based safety oracle used to evaluate the initial states in the AL framework.

Finally, the accuracy of the MPC controller is particularly important because it serves as the oracle for the AL framework. The MPC determines whether each evaluated initial state is feasible or infeasible, and these labels are subsequently used to train the SVM classifier. Consequently, errors or inappropriate assumptions in the MPC model or optimization procedure can propagate into the learning process and alter the estimated safety boundary. The reliability of the learned boundary therefore depends not only on the performance of the SVM and AL strategy, but also on the accuracy and consistency of the MPC-based feasibility evaluation.

In addition, AL is used to identify safety-critical operating conditions and decides when additional learning is necessary. For this purpose, slack variable is added to MPC cost function using a large weighing factor, ρ , so that constraint violations occur only when they cannot be avoided under the given condition. Moreover, we introduced the learning trigger to recognize critical situations from noise. The figures, Fig 5.8 to 5.11, in result shows that our system can identify the critical situation and uses this information for future updates as well as tries the best to keep the vehicle to safe region and avoid collision or mitigate the result of collision.

6.4 Future Work

The proposed framework demonstrates promising simulation results in estimating the safety boundaries of the approaching vehicle for autonomous emergency braking systems, several areas for improvement and extension can be investigated.

One natural extension is the inclusion of more realistic traffic scenarios. For example, we can collect information from data collected from sensors in real traffic conditions that are implemented in different vehicles and gather their data from traffic safety to simulate real traffic world. this method will give us real estimation of lead vehicle driver's behavior.

Future work could also investigate cooperative safety through vehicle-to-vehicle, vehicle-to-infrastructure, vehicle-to-pedestrian, and other V2X communication mechanisms. The system has 2 dimensions of motion and it would be able to change lane in order to avoid collision instead of aggressively braking.

Next enhancement option could be to investigate different machine learning approaches to build the presented learning framework. With respect to classification the applied SVM approach gave accurate performance in the scope of the tested classification problem. However other algorithms like Gaussian process models, neural networks, ensembles of learners, or deeper architectures could possibly scale better and achieve superior performance for problems in higher dimensions. Comparative assessments are necessary to understand which learning algorithms would excel for boundary estimation tasks.

Another possible option might be, the learned safety boundary that was developed in this thesis can be used in real-time decision-making: Instead of repeatedly resolving optimization problems that take time and computational effort to run, the learned classifier could be utilized as a safety monitoring component, used at the end of each loop before the final driving commands are generated. This could lead to a significant decrease in processing cost and provide real-time safety guarantees.

Finally, cooperative active safety system and exchanging information between vehicle to vehicle, vehicle to infrastructure, Vehicle to pedestrian, and vehicle to network or all we can say V2X would be the final update of increasing safety and avoid collision between vehicles and other obstacles.

Bibliography

- [1] Miralem Saljanin, Sven Müller, Jochen Kiebler, Jens Neubeck, and Andreas Wagner. A model predictive control approach for highly automated vehicles in urban environments. *Automotive and Engine Technology*, 7(1):105–113, 2022.
- [2] Qiangqiang Yao and Ying Tian. A model predictive controller with longitudinal speed compensation for autonomous vehicle path tracking. *Applied sciences*, 9(22):4739, 2019.
- [3] Xingbang Gao, Jiaojiao Li, Ruiyuan Liu, Shuai Zhang, and Pengcheng Ma. Research on vehicle longitudinal control method based on model predictive control. *Front. Comput. Intell. Syst*, 1(3):42–47, 2022.
- [4] André Kempf, Markus Herrmann-Wicklmayr, and Steffen Müller. Robust model predictive longitudinal position tracking control for an autonomous vehicle based on multiple models. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 850–857. IEEE, 2020.
- [5] Liang Zhang, Bin Ma, and Penghui Li. Adaptive mpc-based aeb control strategy with dynamic weight and sampling time adjustment. *IET Intelligent Transport Systems*, 19(1):e70112, 2025.
- [6] Manel Ammour, Rodolfo Orjuela, and Michel Basset. Collision avoidance for autonomous vehicle using mpc and time varying sigmoid safety constraints. *IFAC-PapersOnLine*, 54(10):39–44, 2021.
- [7] Ektor Karyotakis. *Braking distance minimization on roads with varying friction*. PhD thesis, Department of Electrical Engineering, Chalmers University of Technology, 2025.
- [8] LUKAS WISELL and SANDER VAN DAM. Safe and efficient vehicle motion control of articulated heavy vehicles.
- [9] Aaron D Ames, Xiangru Xu, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2016.
- [10] Aaron D Ames, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs with application to adaptive cruise control. In *53rd IEEE conference on decision and control*, pages 6271–6278. IEEE, 2014.
- [11] Jun Zeng, Bike Zhang, and Koushil Sreenath. Safety-critical model predictive control with discrete-time control barrier function. In *2021 American control conference (ACC)*, pages 3882–3889. IEEE, 2021.
- [12] Ankush Chakrabarty, Claus Danielson, Stefano Di Cairano, and Arvind Raghunathan. Active learning for estimating reachable sets for systems with unknown dynamics. *IEEE Transactions on Cybernetics*, 52(4):2531–2542, 2020.

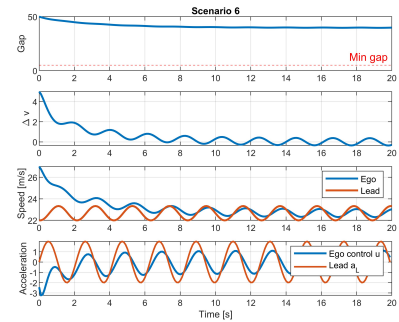
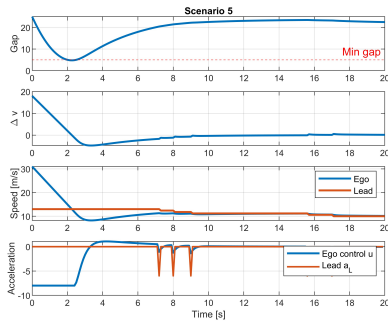
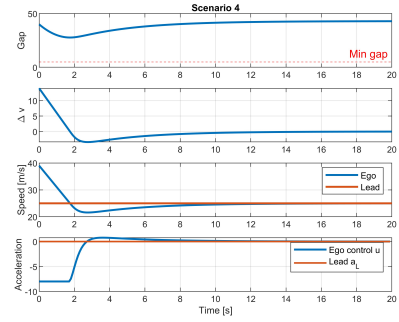
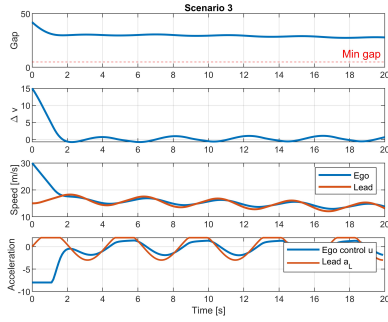
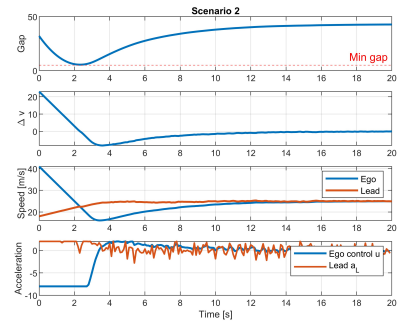
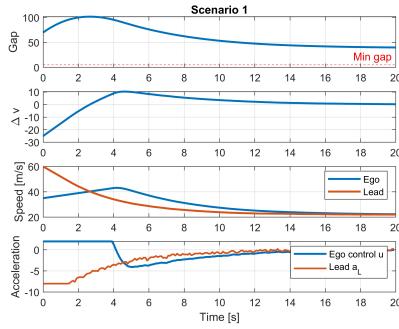
- [13] Wei Xiao, Tsun-Hsuan Wang, Makram Chahine, Alexander Amini, Ramin Hasani, and Daniela Rus. Differentiable control barrier functions for vision-based end-to-end autonomous driving. *arXiv preprint arXiv:2203.02401*, 2022.
- [14] Haitong Ma, Jianyu Chen, Shengbo Eben, Ziyu Lin, Yang Guan, Yangang Ren, and Sifa Zheng. Model-based constrained reinforcement learning using generalized control barrier function. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4552–4559. IEEE, 2021.
- [15] Rajesh Rajamani. *Vehicle dynamics and control*. Springer, 2006.
- [16] Rajesh Rajamani. *Vehicle Dynamics and Control*. Springer, New York, 2 edition, 2012.
- [17] Liangyao Yu and Ruyue Wang. Researches on adaptive cruise control system: A state of the art review. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 236(2-3):143–161, 2022.
- [18] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE access*, 8:58443–58469, 2020.
- [19] James B. Rawlings, David Q. Mayne, and Moritz Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, Madison, WI, 2 edition, 2017.
- [20] Francesco Borrelli, Alberto Bemporad, and Manfred Morari. *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, Cambridge, 2017.
- [21] Eduardo F. Camacho and Carlos Bordons. *Model Predictive Control*. Springer, London, 2 edition, 2007.
- [22] David Q. Mayne, James B. Rawlings, Christopher V. Rao, and Pierre O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.
- [23] Ernst Hairer, Syvert P. Nørsett, and Gerhard Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*, volume 8 of *Springer Series in Computational Mathematics*. Springer, 2 edition, 1993.
- [24] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.
- [25] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.
- [26] Eurika Kaiser, J. Nathan Kutz, and Steven L. Brunton. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2219):20180335, 2018.
- [27] Alan A. Kaptanoglu, Brian M. de Silva, Urban Fasel, Kadierdan Kaheman, Andy J. Goldschmidt, Jared L. Callahan, Charles B. Delahunt, Zachary G. Nicolaou, Kathleen Champion, Jean-Christophe Loiseau, J. Nathan Kutz, and Steven L. Brunton. PySINDy: A comprehensive python package for robust sparse system identification. *Journal of Open Source Software*, 7(69):3994, 2022.
- [28] Ankush Chakrabarty, Vu Dinh, Martin J Corless, Ann E Rundell, Stanisław H Żak, and Gregery T Buzzard. Support vector machine informed explicit non-

- linear model predictive control using low-discrepancy sequences. *IEEE Transactions on Automatic Control*, 62(1):135–148, 2016.
- [29] Weiming Xiang, Hoang-Dung Tran, and Taylor T Johnson. Output reachable set estimation and verification for multilayer neural networks. *IEEE transactions on neural networks and learning systems*, 29(11):5777–5783, 2018.
- [30] Jing Zhang, Xindong Wu, and Victor S Shengs. Active learning with imbalanced multiple noisy labeling. *IEEE transactions on cybernetics*, 45(5):1095–1107, 2014.
- [31] Andreas Krause and Carlos Guestrin. Nonmyopic active learning of gaussian processes: an exploration-exploitation approach. In *Proceedings of the 24th international conference on Machine learning*, pages 449–456, 2007.

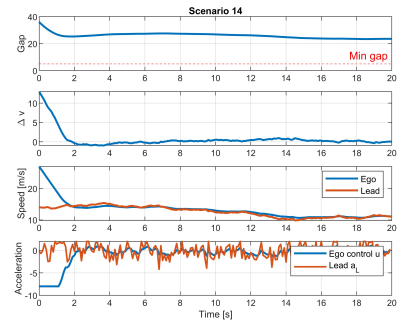
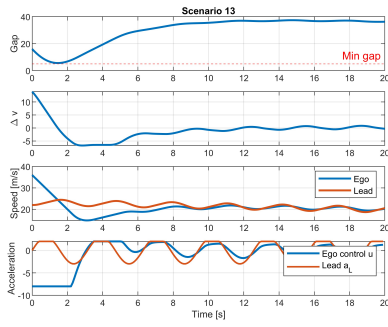
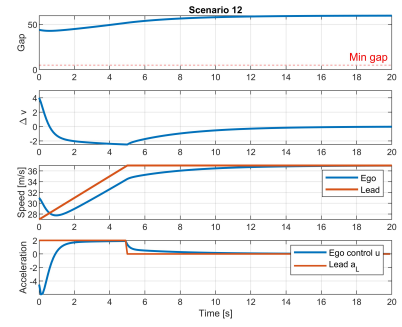
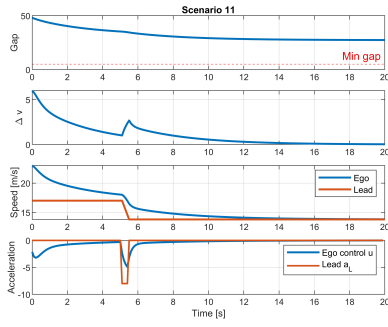
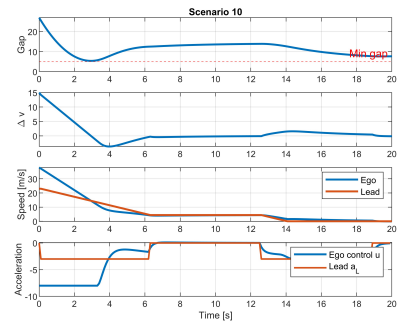
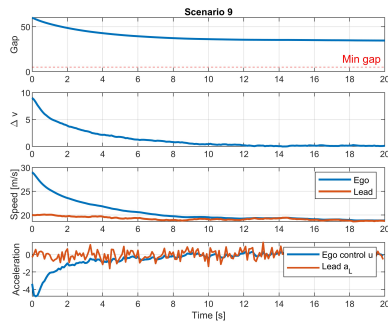
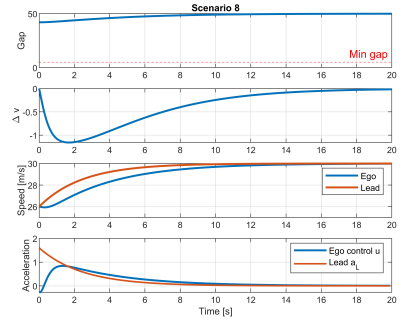
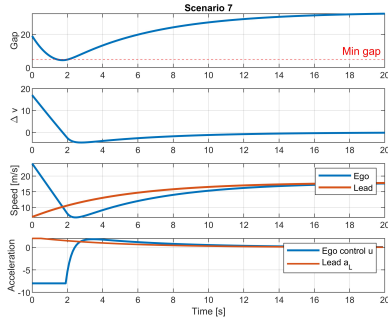
A

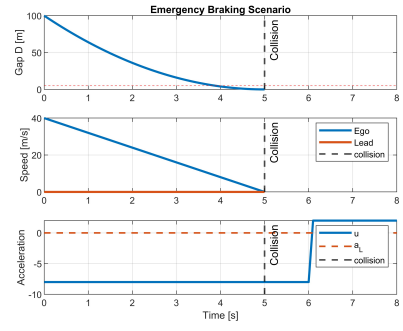
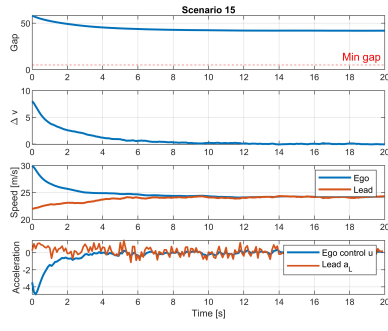
All scenarios

The following scenarios are considered and investigated to evaluate the performance, accuracy, and response of the MPC controller in the proposed autonomous emergency braking system.



A. All scenarios

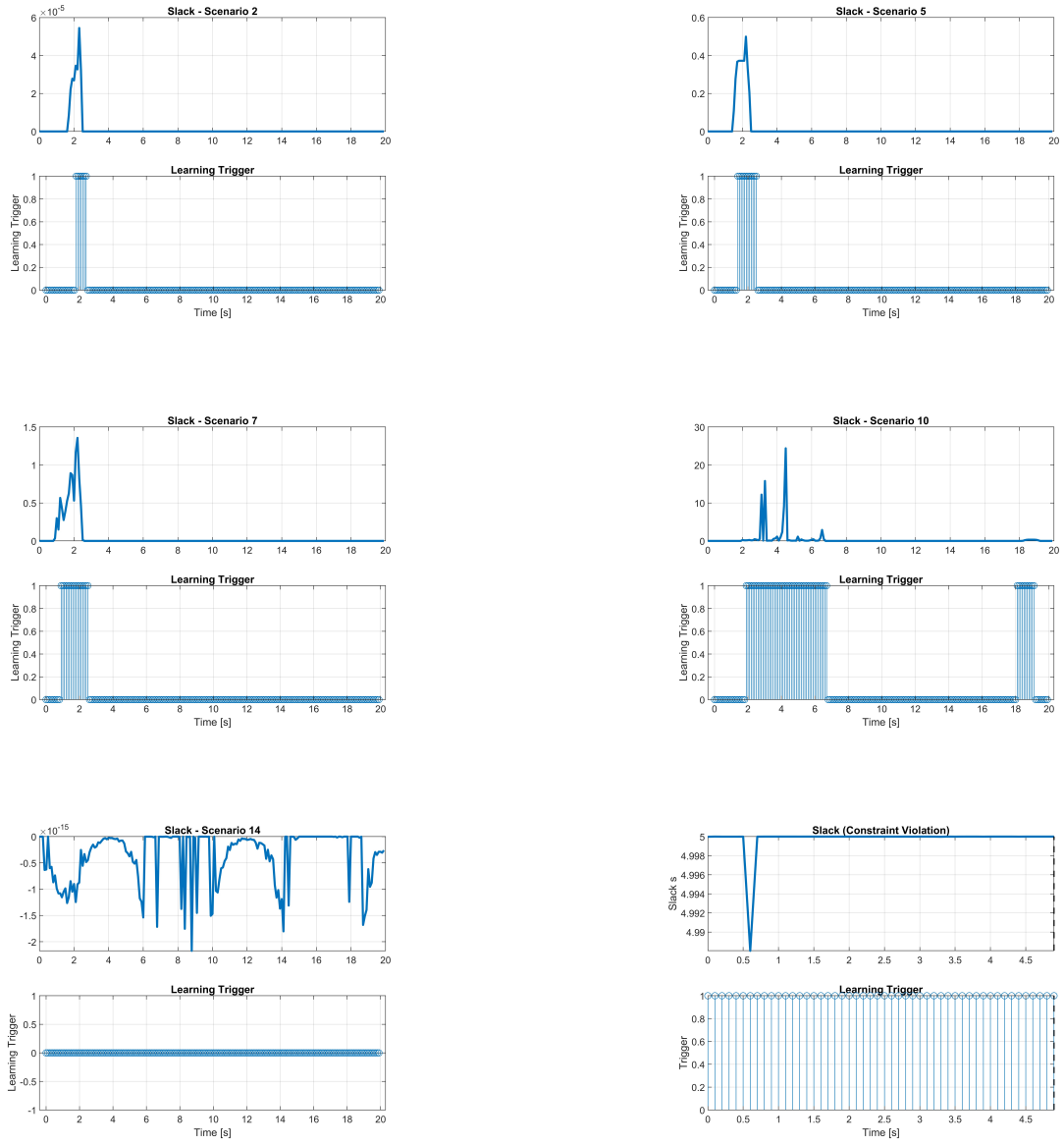




B

Slack and Trigger

Five representative scenarios and the worst-case scenario are presented to illustrate the variation in the slack variable and learning trigger.



C

Initial values

The following table lists the initial ego-vehicle speed, lead-vehicle speed, and relative distance for each scenario.

	Ego vehicle speed m/s	Lead vehicle speed m/s	Relative distance m
Scenario 1	35	60	70
Scenario 2	41	18	32
Scenario 3	30	15	42
Scenario 4	39	25	40
Scenario 5	31	13	25
Scenario 6	27	22	50
Scenario 7	24	7	19
Scenario 8	26	26	42
Scenario 9	29	20	60
Scenario 10	38	23	27
Scenario 11	23	17	48
Scenario 12	31	27	44
Scenario 13	36	22	16
Scenario 14	27	14	36
Scenario 15	30	22	58
Worst-case scenario	40	0	100

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



CHALMERS
UNIVERSITY OF TECHNOLOGY