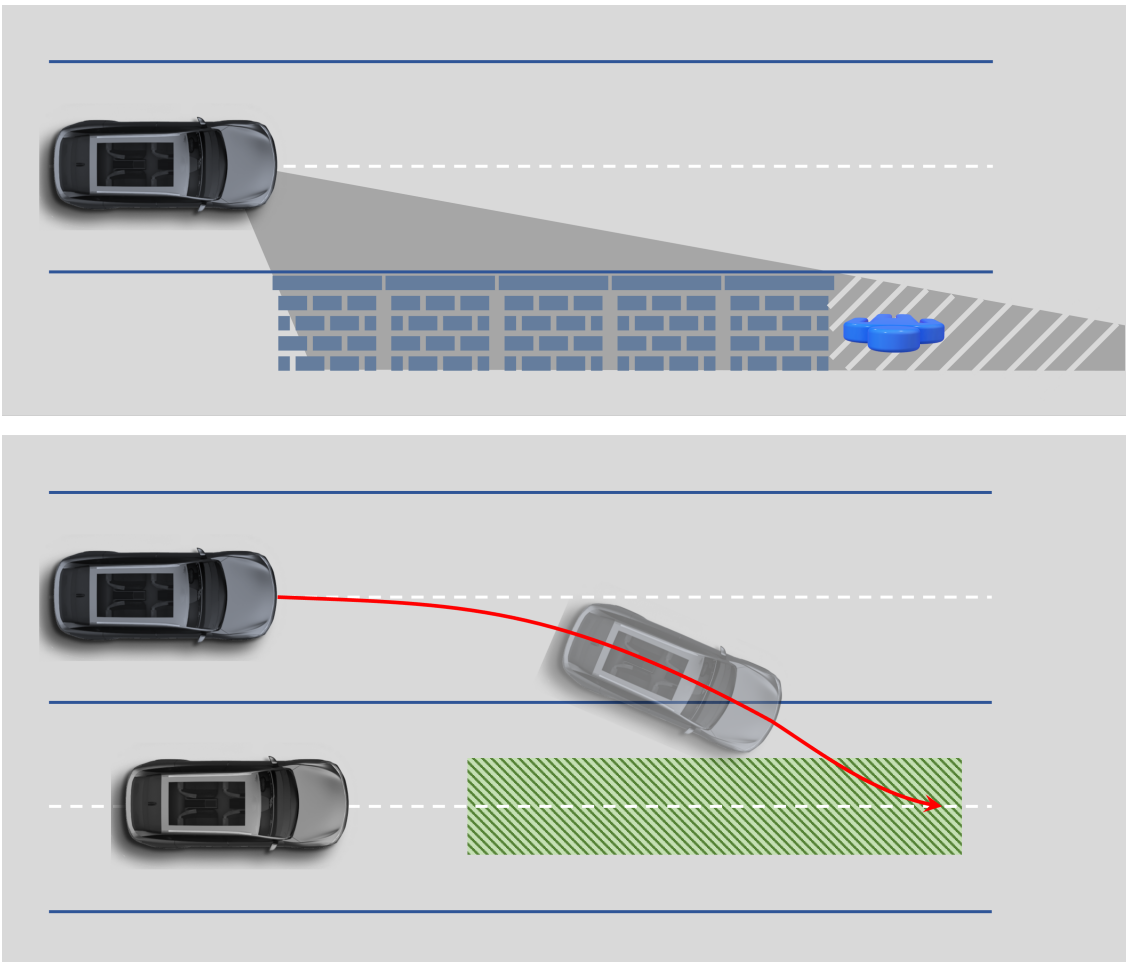
CHALMERS
UNIVERSITY OF TECHNOLOGY



# Safety Assessment for Precautionary Collision Avoidance

## A Collision Avoidance System Based on Safety-Set

Lizi Teng and Peilin Lyu

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024
www.chalmers.se

Master's thesis 2024

# Safety Assessment for Precautionary Collision Avoidance

A Collision Avoidance System Based on Safety-Set

Lizi Teng
Peilin Lyu


CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
*Division of Systems, Control, and Mechatronics*
Chalmers University of Technology
Gothenburg, Sweden 2024

Safety Assessment for Precautionary Collision Avoidance
A Collision Avoidance System Based on Safety-Set
Lizi Teng, Peilin Lyu



Supervisor: Karl Rundstedt, Qualcomm
Examiner: Nikolce Murgovski, Department of Electrical Engineering

Master's Thesis 2024
Department of Electrical Engineering
Systems and Control
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000



Cover: Illustration of an Autonomous Vehicle's Response: Detections of a Hidden Pedestrian (Top) and a Target Vehicle (Bottom) to Avoid Collision

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2024

Safety Assessment for Precautionary Collision Avoidance
A Collision Avoidance System Based on Safety-Set
Lizi Teng and Peilin Lyu
Department of Electrical Engineering
Chalmers University of Technology

# Abstract

This thesis presents the implementation of a precautionary collision avoidance system, centered on obtaining a safety set and employing Model Predictive Control (MPC) based on the safety set. The safety set, which defines the boundaries within which the vehicle can safely operate, is derived through optimization problems or analytical calculations. By applying the approaches, the system proactively anticipates and mitigates potential collision risks, such as hidden pedestrians emerging from blind spots.

Two critical scenarios are examined: the hidden pedestrian scenario, where the vehicle must react to unforeseen obstacles, and the lane change scenario, which requires safe merging while considering nearby traffic. The integration of the precautionary safety-set with MPC allows the system to maintain a high success rate in collision avoidance while ensuring smooth and efficient vehicle operation. Experimental results confirm the system's reliability and demonstrate its ability to maintain a high success rate in safely navigating complex scenarios, thus highlighting its potential to significantly enhance the safety and performance of autonomous driving systems.

Keywords: Safety-set, CAS, MPC, N-step controlable set, nonlinear optimization, AEB

# Acknowledgements

First and foremost, we would like to express our deepest gratitude to my supervisor, Karl Rundstedt, for the continuous support, guidance, and encouragement throughout the process of completing this thesis. Your invaluable insights and expertise have greatly contributed to the development of our work.

We would also like to extend our heartfelt thanks to my examiner, Nicolce Murgovski, for your constructive feedback and valuable suggestions, which have been essential in shaping the final outcome of this project.

Additionally, We grateful to Qualcomm Gothenburg for providing us with the opportunity to collaborate and work on this project. The resources, knowledge, and support we received during our time at the company were crucial for the successful completion of this thesis.

Lastly, We want to thank our family and friends for their unwavering support, patience, and encouragement. Your belief in us kept us motivated and focused throughout this journey.

Peilin Lyu & Lizi Teng, Gothenburg, October 2024

# Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis. The following nomenclature is sorted in order of appearance:

- Vectors are bold lower case letters "$\mathbf{x}$"
- Sets are in blackboard-bold letters "$\mathbb{S}$"
- Time derivatives are marked by "$\dot{z}$"

## Sets

| | |
|---|---|
| $\mathbb{I}$ | Set of indices for inequality constraints |
| $\mathbb{E}$ | Set of indices for equality constraints |
| $\mathbb{R}$ | Set of real numbers |
| $\mathbb{X}$ | Set of states |
| $\mathbb{U}$ | Set of inputs |
| $\mathbb{X}_f$ | Set of final states |
| $\mathbb{F}$ | Feasible set of MPC |
| $\mathbb{I}_{mpc}$ | Set of indices for inequality constraints of MPC |
| $\mathbb{E}_{mpc}$ | Set of indices for equality constraints of MPC |
| $\mathbb{X}_N$ | Feasible set of initial states |
| $\mathbb{U}_N$ | Set of valid control sequences |
| $\mathbb{K}_N$ | N-step controllable set |
| $\mathbb{K}_1$, $\mathbb{K}_2$, $\mathbb{B}_m$, $\mathbb{O}_n$ | Compact convex sets |

## Parameters

| | |
|---|---|
| $lw$ | Wheelbase of the vehicle |
| $\mathbf{A}_p$ | State transition matrix of point mass model |
| $\mathbf{B}_p$ | Control input matrix of point mass model |

| | |
|---|---|
| $T_{claim}$ | The prediction horizon of claim area |
| $dt$ | Time interval |
| $r_p$ | Radius of a person |
| $\mathbf{P}$ | Control points of bézier curve |
| $\text{length}_E$ | The length of the ego vehicle |
| $G$ | Gravitational acceleration |
| $a_a$ | The maximum acceleration for the target vehicle |
| $a_b$ | The minimum acceleration for the target vehicle |
| $a$ | The minimum acceleration for the ego vehicle |
| $j$ | The minimum jerk for the ego vehicle |

$t_r$ Reaction time.

# Variables

| | |
|---|---|
| $\mathbf{z}$ | Vector of decision variables |
| $\mathbf{x}$ | State vector |
| $\mathbf{u}$ | Control input vector |
| $\mathbf{x}^*$ | Optimal state vector |
| $\mathbf{u}^*$ | Optimal control input vector |
| $\mathbf{x}_0$ | Initial state vector |
| $\mathbf{x}_b$ | State vector of kinematics bicycle model |
| $\mathbf{u}_b$ | Control input vector of kinematics bicycle model |
| $\mathbf{x}_p$ | State vector of point mass model |
| $\mathbf{u}_p$ | Control input vector of point mass model |
| $x_b$ | Position of center of rear axle on x axis of kinematics bicycle model |
| $y_b$ | Position of center of rear axle on y axis of kinematics bicycle model |
| $v_b$ | Longitudinal velocity of kinematics bicycle model |
| $a_b$ | Longitudinal acceleration of kinematics bicycle model |
| $\gamma_b$ | Yaw angle of kinematics bicycle model |
| $\delta_b$ | Steering angle of kinematics bicycle model |
| $j_b$ | Jerk of kinematics bicycle model |
| $p_x$ | Longitudinal position of point mass model |
| $p_y$ | Lateral position of point mass model |
| $v_x$ | Longitudinal velocity of point mass model |

| | |
|---|---|
| $v_y$ | Lateral velocity of point mass model |
| $a_x$ | Longitudinal acceleration of point mass model |
| $a_y$ | Lateral acceleration of point mass model |
| $j_x$ | Longitudinal Jerk of point mass model |
| $j_y$ | Lateral Jerk of point mass model |
| $\mathbf{q}$, $\mathbf{p}$ | points of convex sets |
| $V_{P_1}$, $V_{P_2}$ | Matrices of vertices of polytopes |
| $y$ | Vector of observed values of polynomial regression |
| $X$ | Matrix of input features of polynomial regression |
| $\beta$ | The parameters to be estimated of polynomial regression |
| $\varepsilon$ | The error vector of polynomial regression |
| $a_{\text{req}}$ | The required acceleration in the AEB algorithm |
| $x_P$ | The longitudinal positions of pedestrian in the AEB algorithm |
| $x_E$ | The longitudinal positions of ego vehicle in the AEB algorithm |
| $v_E$ | The longitudinal velocity of ego vehicle in the AEB algorithm |
| $x^{\text{c}}$ | Longitudinal position of the point on bézier curve |
| $y^{\text{c}}$ | Lateral position of the point on bézier curve |
| $\kappa$ | The curvature of bézier curve |
| $v_{init}^{c}$ | The initial velocity when entering the bézier curve |
| $p_x^p$ | The longitudinal position of pedestrian |
| $p_y^p$ | The lateral position of pedestrian |
| $v_x^p$ | The longitudinal velocity of pedestrian |
| $v_y^p$ | The lateral velocity of pedestrian |
| $v^e$ | The velocity of ego vehicle |
| $\alpha_{pedes}$ | Coefficient of pedestrian velocity |
| $D_b$ | Braking distance of target vehicle |
| $D_a$ | Accelerating distance of target vehicle |
| $v_0^t$ | Initial velocity of target vehicle |
| $\mathbf{x_s^e}$ | State vector of ego vehicle for calculating safety-set |
| $\mathbf{u_s^e}$ | Control input vector of ego vehicle for calculating safety-set |
| $\mathbf{x}_{\text{ctrl}}^{\mathbf{e}}$ | State vector of ego vehicle for precautionary controller |
| $\mathbf{u}_{\text{ctrl}}^{\mathbf{e}}$ | Control input vector of ego vehicle for precautionary controller |
| $\mathbf{x}_{ref}$ | Reference state vector of ego vehicle for precautionary controller |
| $v_0^t$ | Initial velocity of target vehicle |

| | |
|---|---|
| $x_0^t$ | Initial longitudinal position of target vehicle |
| $v_c^e$ | Velocity of ego vehicle at collision point |
| $v_c^t$ | Velocity of target vehicle at collision point |
| $x_c^t$ | Longitudinal position of target vehicle at collision point |
| $x_k^e$ | Longitudinal position for sampled point K |
| $y_k^e$ | Lateral position for sampled point K |
| $v_k^e$ | Critical velocity for sampled point K |

# Functions

| | |
|---|---|
| $h(\cdot)$ | Objective function |
| $m(\cdot)$ | Equality constraint function |
| $n(\cdot)$ | Inequality constraint function |
| $V_N(\cdot)$ | Cost function of MPC |
| $l_f(\cdot)$ | Final cost function of MPC |
| $l(\cdot)$ | Process cost function of MPC |
| $l(\cdot)$ | Process cost function of MPC |
| $f(\cdot)$ | System dynamic of MPC |
| $g(\cdot)$ | Inequality constraint functions of MPC |
| $\mathbf{B}(\cdot)$ | Function of bézier curve |
| $\mathcal{J}_s^l$ | Cost function for calculating lower safety-set |
| $\mathcal{J}_s^u$ | Cost function for calculating upper safety-set |
| $f_b(\cdot)$ | System dynamic of kinematics bicycle model |

# Contents

# List of Figures

# List of Tables

# 1
# Introduction

In the age of autonomous driving and advanced driver assistance systems (ADAS), the safety of road users remains paramount. Collision Avoidance Systems (CAS) serve as a key technology designed to prevent or mitigate the severity of a collision. These systems intervene by automatically initiating emergency maneuvers such as braking or steering when an imminent threat is detected. However, real-world scenarios present complex challenges, including sensor inaccuracies, occlusion and unpredictable road user behaviour, which may affect the effectiveness of these systems. To handle these challenges, the implementation of a precautionary safety CAS system may be an effective method for the mitigation of potential collision threats.

## 1.1 Background

With the growing number of vehicles on the road, traffic accidents are increasing at a rapid pace [1]. Worldwide, more than one million people die in road traffic accidents every year, with the majority of accidents caused by human error. Autonomous driving technology has the potential to change transportation by providing safer and more efficient mobility solutions. A key technical challenge in this area is to ensure that advanced driver assistance systems (ADAS) enable autonomous vehicles (AVs) to consistently avoid collisions, particularly in scenarios involving hidden or unexpected obstacles. Moreover, it is a crucial requirement for the widespread adoption and societal trust in autonomous technologies.

Automated emergency braking (AEB) system is a critical component within the broader suite of CAS technologies. Mattias, Erik and Jonas [2] introduces a model-based algorithm for estimating potential maneuvers (steering, braking, or accelerating) a driver can employ to avoid collisions with arbitrary objects. It utilizes a linear bicycle model for vehicle motion and a polygonal representation for object perimeters, facilitating the assessment of evasive maneuvers and the need for driver assistance. Their research suggests a method to describe the motion of the ego vehicle and to set a reasonable limit for acceleration to avoid possible future collisions.

In [3], a research conducted at NVIDIA by Nistér and Lee introduces the Safety Force Field (SFF) concept as a safety layer for autonomous driving vehicles. The SFF aims to prevent dangerous behaviors toward other road users by utilizing a physics-based forward simulation. This involves calculating safe stopping trajecto-

ries for every perceived object in traffic. The future path of each object, or Claimed Set, is analyzed for overlaps that indicate collision risks. SFF employs a safety potential function to determine the best real-time control commands by evaluating overlaps in dynamic traffic scenarios, optimizing collision avoidance strategies.

Similarly, Shalev-Shwartz et al. from Mobileye have proposed a white-box, interpretable, mathematical model for safety assurance known as Responsibility-Sensitive Safety(RSS) [4]. The RSS model is designed to formalize and contextualize human judgment regarding all driving situations and dilemmas. In [5], similar methods was proposed, which could guarantee that the AVs will not crash as long as other road users behave according to certain assumed behaviours, and mitigate collisions for unforeseen behaviour of other road users.

To achieve the desired level of safety, it is essential to implement control strategies that enable the vehicle to respond effectively to dynamic and unpredictable environments. This is where the nonlinear programming and Model Predictive Controller (MPC) comes in. MPC has emerged as an important control approach for autonomous vehicles, primarily due to its ability to anticipate future states by solving an optimisation problem over a finite prediction horizon. It provides real-time decision making based on current vehicle states, future trajectory predictions, and environmental information. The flexibility of MPC lies in its ability to incorporate system constraints such as speed limits, acceleration limits, jerk limits, and dynamic obstacle avoidance, all of which are critical to maintaining safety in both normal comfort operation and emergency scenarios.

A specific aspect of safety in AVs is precautionary collision avoidance. This involves ensuring that the vehicle is not only capable of reacting to imminent collisions but also maintains a state that is consistently prepared for emergency maneuvers. By offline computing an N-step controllable set (safety-set) through MPC, the vehicle's safety velocity on different positions can be planned such that the vehicle remains within a predetermined safety-set. The key challenge here is to balance the need for maintaining efficient motion with the imperative to remain within safe bounds under all circumstances.

In summary, ensuring the safety of autonomous vehicles requires not only reactive measures, but also a precautionary approach that anticipates potential hazards and equips the vehicle to proactively avoid collisions. MPC is well suited for this purpose due to its predictive capabilities and flexibility in dealing with constraints. By focusing on maintaining a safety-set for potential threats, this work aims to contribute to the development of safer, more reliable autonomous vehicles capable of operating in complex, real-world environments.

## 1.2 Aim

The aim of the thesis is to provide a more foresighted active safety function, which could respond to potential threats so that early avoidance and mitigation of risks

could be implemented. This precautionary safety function is designed to handle potential threats by detecting and assessing collision risks in various scenarios and then defining motion restrictions based on the level of risk. The goal of this project is to enhance the capabilities of CAS and contribute to the overall safety of all road users. To achieve this, the following objectives have been established:

1. Define agent motion models used by the precautionary CAS. These models should account for future motions and the necessary safety maneuvers the agent must execute to maintain safety.
2. Formulate optimization problems to determine motion constraints for collision avoidance, and use fitting to represent the safety-set as a continuous function.
3. Design an optimal controller based on the safety-sets to execute precautionary and reasonable operations, enabling the ego vehicle to avoid collision risks in advance.

## 1.3 Research questions

The main research questions during this project are

- How to design and implement a precautionary collision avoidance system that has the foresight to detect and mitigate collision threats? This includes two sub-problems:
  - What approach could be employed to perform safety assessment regarding collision threats?
  - What strategies can be implemented to enable autonomous vehicles to respond appropriately to mitigate threats in advance and ensure safety?
- How should the motions of visible or occluded traffic participants be described for the purpose of threat assessment?
- How to validate the performance of our precautionary CAS?

## 1.4 Scope and limitations

This project focuses on the hidden vulnerable road users (VRU) scenario and makes an exploration in Lane change scenario (Figure 1.1). And in our scenarios, only a single pedestrian or target vehicle is considered, representing relatively simple situations used as a starting point for developing and evaluating our approach. And in the scenarios, the ego vehicle's emergency maneuvers for obstacle avoidance are considered to include the strongest possible braking and turning angles within the vehicle's limits.

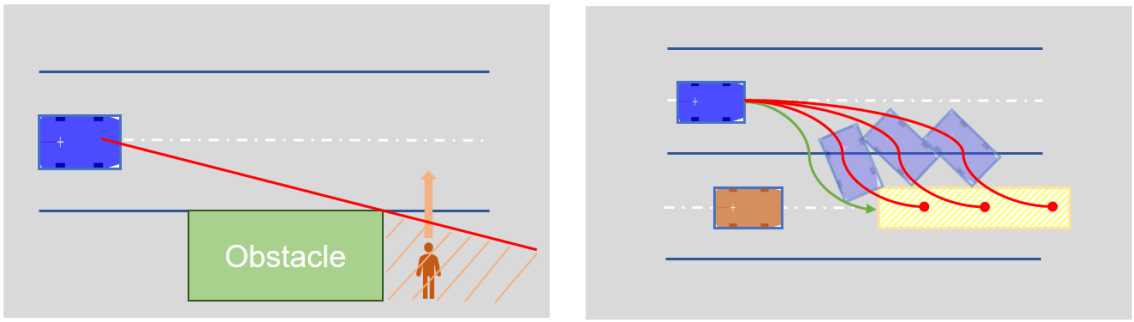



**Figure 1.1:** The Hidden pedestrian scenario on the left and Lane change scenario on the right.

The vehicle dynamics are simulated using a simplified kinematic model, without considering mechanical characteristics. While many vehicle-related parameters are difficult to estimate or obtain, we make reasonable assumptions for all data to build a model of a hypothetical vehicle.

The developed control system is tested solely in a simulation environment, which considers only geometric constraints from a bird's-eye view. Additionally, all vehicle sensors and actuators are assumed to have zero delay and zero error.

## 1.5 Ethical and sustainability aspects

The preventive safety approach is grounded in reducing risk and proactively avoiding accidents. This method emphasizes not merely reacting to hazards after they occur but preventing them before they can cause harm. Therefore, it aligns with the ethical principle of harm minimization, which seeks to reduce potential harm wherever possible. By maintaining the vehicle within a predefined safety-set, AVs continuously monitor their environment and make adjustments to minimize the likelihood of incidents. This proactive behavior helps protect not only the passengers inside the vehicle but also pedestrians and other road users, fostering a safer overall traffic system.

From a sustainability perspective, the safety-set concept can also enhance energy efficiency. Optimizing vehicle behavior to maintain safe velocity limits reduces the need for rapid braking or acceleration, leading to decreased fuel consumption or energy usage in electric vehicles. Additionally, effective collision avoidance prevents the destruction of large material values, minimizing the environmental impact associated with repairing or replacing damaged vehicles. This contributes to the overall sustainability of AV systems while improving road safety through more effective risk prediction and mitigation.

Nevertheless, the implementation of these systems raises key ethical considerations. While data privacy and security are important, the primary focus should be on ensuring that ADAS systems make decisions that are ethical and aligned with traffic safety responsibilities. These systems must be designed to appropriately assign

responsibility in the event of an accident, based on the actions of all road users. Additionally, it's essential to maintain a seamless driving experience for the vehicle users. Balancing these ethical and legal responsibilities with a smooth user experience is critical for fostering public trust and supporting the widespread adoption of this technology. Furthermore, the algorithms employed by these systems must be explainable to ensure accountability, particularly in the event of an accident.

## 1.6 Thesis overview

The thesis is structured as follows.

Chapter 2 presents the theoretical background of the project. It starts with an introduction to NLP and MPC, along with the relevant theories, followed by a discussion on polynomial regression. Additionally, it covers the fundamentals of Bézier curves and Autonomous emergency braking(AEB).

Chapter 3 explains in detail how precautionary safety is implemented. It begins with an overview of the system architecture, followed by a discussion of the various methods used to achieve each step.

Chapter 4 focuses on the design and results of two experiments, both conducted in the first scenario involving a hidden pedestrian. The first experiments 4.1.2 compare the safety performance of a control system equipped solely with an AEB system to that of a control system enhanced 4.1.3 by a precautionary safety-set, in response to the sudden appearance of a pedestrian from behind an obstacle. The second experiments compare the time required for a control system assisted by the precautionary safety-set versus a control system that navigates online passing through the area of which is partly hidden. The goal of this comparison is to highlight the efficiency of handling the hidden area.

Chapter 5 concludes the thesis by summarizing the overall results and suggesting possible directions for future work.

# 2

# Background Theory

## 2.1 Model predictive control

### 2.1.1 Basics of MPC

Model Predictive Control(MPC) is an advanced control strategy widely used in process control and various engineering applications. As an optimal control problem, the core concept of MPC is to use an internal model to predict the future states of the system and determine the optimal control inputs by solving an optimization problem that includes multiple constraints. At each time step, MPC solves an optimal control problem over a finite time horizon to obtain a sequence of control actions for future time steps.

**Optimization problem** An optimization problem involves finding the best solution, typically the minimum value, of an objective function while satisfying a set of constraints. The standard formulation of a general optimization problem is to:

$$\begin{aligned} &\underset{\mathbf{z}}{\text{minimize}}\ h(\mathbf{z}), \\ &\text{subject to}\ m_i(\mathbf{z}) \leq 0, \quad i \in \mathbb{I}, \\ &\qquad\qquad n_j(\mathbf{z}) = 0, \quad j \in \mathbb{E}, \\ &\qquad\qquad \mathbf{z} \in \mathbb{R}^n. \end{aligned} \tag{2.1}$$

where:

- $\mathbf{z} \in \mathbb{R}^n$ : vector of decision variables $x_i,\ i = 1, \ldots, n$,
- $h : \mathbb{R}^n \to \mathbb{R} \cup \pm\infty$ : objective function,
- $m_i, n_j : \mathbb{R}^n \to \mathbb{R}$ : constraint function defining restriction on $\mathbf{z}$,
- $m_i \geq 0,\ i \in \mathbb{I}$ : inequality constraints,
- $n_j = 0,\ j \in \mathbb{E}$ : equality constraints

Nonlinear programming (NLP) is the sub-field of mathematical optimization that deals with nonlinear problems. With some nonlinear functions $f$, $g_i$, $i \in \mathbb{I} \cup \mathbb{E}$, the optimization problem could be seen as NLP. For systems with nonlinear dynamics, or when the constraints or cost function are nonlinear, solving MPC becomes equivalent to solving a nonlinear programming (NLP) problem.

**Basic formulation of MPC** The general MPC problem could also be seen in this form, an optimal control problem with horizen $N$ at time $k$ is:

$$\begin{aligned} \min_{\mathbf{u}(k:k+N-1|k)} \quad & V_N(\mathbf{x}_0, \mathbf{u}(k:k+N-1|k)) \\ \text{subject to} \quad & \mathbf{x}(k+i+1|k) = f(\mathbf{x}(k+i|k), \mathbf{u}(k+i|k)), \quad \mathbf{x}(k) = \mathbf{x}_0, \\ & g(\mathbf{x}(k+i|k), \mathbf{u}(k+i|k)) \leq 0, \\ & \mathbf{x}(k+i|k) \in \mathbb{X}, \ \mathbf{u}(k+i|k) \in \mathbb{U}, \quad \text{for all } i = 0, \ldots, N-1, \\ & \mathbf{x}(k+N|k) \in \mathbb{X}_f \subseteq \mathbb{X}. \end{aligned} \tag{2.2}$$

where:

- The cost function could be defined as:

$$V_N(\mathbf{x}_0, \mathbf{u}(k:k+N-1|k)) = l_f(\mathbf{x}(k+N|k)) + \sum_{i=0}^{N-1} l(\mathbf{x}(k+i|k), \mathbf{u}(k+i|k)) \tag{2.3}$$

  with the final cost $l_f$ and the process cost $l$.

- $\mathbf{x}(k+i|k)$ is the state vector at time step $i$, $\mathbf{u}(i)$ is the control input vector, $\mathbf{x}_0$ is the initial state
- $\mathbf{x}(k+i+1|k) = f(\mathbf{x}(k+i|k), \mathbf{u}(k+i|k))$ represents the discrete time state equation
- $g(\mathbf{x}(k+i|k), \mathbf{u}(k+i|k)) \leq 0$ represents the inequality constraints,
- It is assumed that $\mathbb{U}$, $\mathbb{X}$, and $\mathbb{X}_f$ all contain the origin.

By solving the optimal problem, the optimal control and state sequences are obtained:

$$\begin{aligned} \mathbf{u}^*(k:k+N-1|k) &= \{\mathbf{u}^*(k|k), \mathbf{u}^*(k+1|k), \ldots, \mathbf{u}^*(k+N-1|k)\}, \\ \mathbf{x}^*(k:k+N|k) &= \{\mathbf{x}^*(k|k), \mathbf{x}^*(k+1|k), \ldots, \mathbf{x}^*(k+N|k)\} \end{aligned} \tag{2.4}$$

However only the first step of the optimal control solution is implemented, and then the plant state is sampled again and the calculations are repeated from the new current state. The prediction horizon keeps being shifted forward and the strategy is known as receding horizon, as illustrated in Figure 2.1.

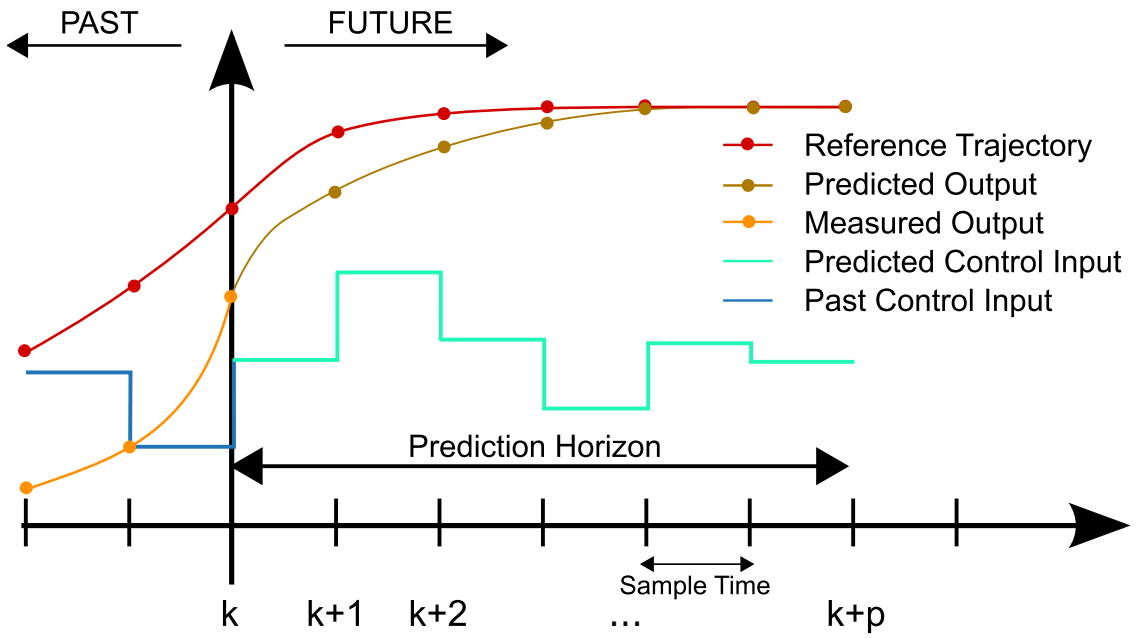



**Figure 2.1:** The receding horizon concept of MPC.

When the system's dynamic model is nonlinear, MPC is referred to as Nonlinear Model Predictive Control (NMPC). The optimization problem in NMPC involves nonlinear constraints and objective functions, which are typically solved using NLP. In other words, the core of NMPC lies in solving an NLP problem at each iteration to determine the control inputs, aiming to find the optimal control strategy within the prediction horizon.

### 2.1.2 Feasibility of MPC

The feasibility problem in optimization refers to determining whether there exists a solution that satisfies the given constraints. If there is an optimal variable that satisfies all the constraints of the problem, then the optimization problem is solvable. Therefore, the feasibility of an optimization problem depends entirely on the constraints. When the constraints are the same, the feasible set of the optimization problem will also be identical. In the case of an MPC problem, we can also observe that the feasibility depends only on the system dynamics (equality constraints) and other constraints.

**Feasible set of MPC** The feasible set $\mathbb{F}$ is defined as the set of all $\mathbf{x}$ that satisfy all the constraints of the optimization problem. Based on Equation 2.1, the feasible set $\mathbb{F}$ is given by:

$$\mathbb{F} = \{\mathbf{x} \in \mathbb{R}^n \mid g_i(\mathbf{x}) \leq 0, \quad i \in \mathbb{I}_{mpc} \text{ and } f_j(\mathbf{x}) = 0, \quad j \in \mathbb{E}_{mpc}\}. \tag{2.5}$$

For the MPC, not all control sequences are feasible. Only control sequences that satisfy both the control and state constraints, while steering the state trajectory to the target set $\mathbb{X}_f$ in the final step of the horizon, are considered valid. The set of

valid control sequences typically depends on the initial state $\mathbf{x}(0) = \mathbf{x}_0$. Formally, the implicit constraint on the control sequences can be expressed as follows:

$$\mathbf{u}(0 : N-1) \in \mathbb{U}_N(x_0) \tag{2.6}$$

where

$$\mathbb{U}_N(\mathbf{x}_0) = \{\mathbf{u}(0 : N-1) \mid \mathbf{x}(i+1) = f(\mathbf{x}(i), \mathbf{u}(i)), \mathbf{x}(0) = \mathbf{x}_0, i = 0, \ldots, N-1, \mathbf{u}(i) \in \mathbb{U}, \mathbf{x}(i) \in \mathbb{X}, \mathbf{x}(N) \in \mathbb{X}_f\} \tag{2.7}$$

is the set consists of all control sequences that guide the initial state $\mathbf{x}_0$ to the terminal set $\mathbb{X}_f$ while satisfying the control and state constraints at every time step. For certain initial states, it may be impossible to identify any control sequence that meets these constraints. As a result, the optimization problem is meaningful only for initial states within a specific subset $\mathbb{X}_N$ of the state space, known as the **feasible set of initial states**,

$$\mathbb{X}_N = \{\mathbf{x}_0 \in \mathbb{X} \mid \mathbb{U}_N(\mathbf{x}_0) \neq \emptyset\} \tag{2.8}$$

And the feasible set could alternatively be defined as the N-step controllable set of the target set $\mathbb{X}_N = \mathbb{K}_N(\mathbb{X}_f)$, which provides a method for determining the feasible set.

The **N-step controllable set** $\mathbb{K}_N(\mathbb{S})$ is the set of initial states that can reach the target set $\mathbb{X}_f$ within N steps, while satisfying both state and control constraints at all times. It can be computed recursively as follows.

$$\mathbb{K}_{i+1}(\mathbb{X}_f) = \text{pre}(\mathbb{K}_i(\mathbb{X}_f)) \cap \mathbb{X}, \quad \mathbb{K}_0(\mathbb{X}_f) = \mathbb{X}_f. \tag{2.9}$$

Here, the $pre(\mathbb{S})$ represent Precursor set, it could be describe as:

$$\text{pre}(\mathbb{S}) = \{\mathbf{x}(k) \in \mathbb{R}^n \mid \exists \mathbf{u}(k) \in \mathbb{U}, \mathbf{x}(k+1) = f(\mathbf{x}(k), \mathbf{u}(k)) \in \mathbb{S}\} \tag{2.10}$$

### 2.1.3 Vehicle modeling for constraints

In MPC, the state-space model, serves as internal model in MPC problem, which describes the system's dynamic behavior and predicts its future evolution. The model is also seen as an equality constraint within the optimization problem, ensuring that the predicted states adhere to the system's dynamics.

In our project, two types of vehicle kinematic models are employed to simulate vehicle motion and guide the control process.

**Kinematics bicycle model** A kinematic bicycle model, with the reference point located at the center of the rear axle, is illustrated schematically in Figure 2.2.

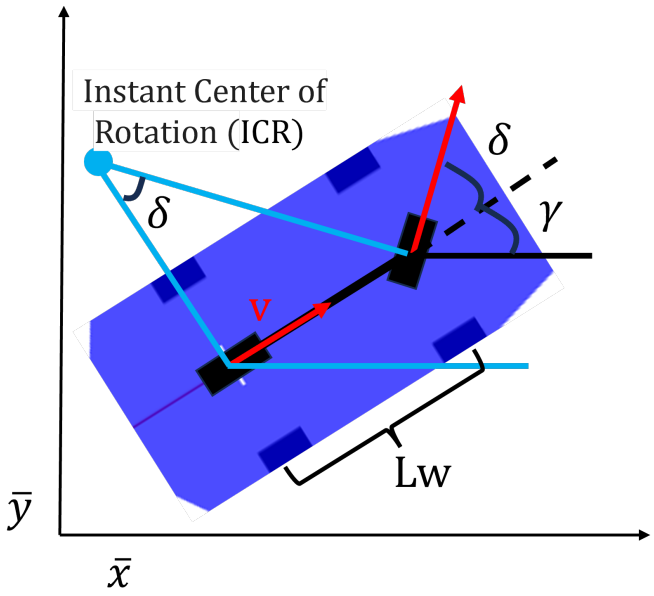



**Figure 2.2:** Kinematics bicycle model of vehicle vehicle, $Lw$ is the wheelbase, $\bar{x}$ and $\bar{y}$ are the axes of the coordinate system., $\gamma$ is the yaw of the vehicle , $\delta$ is the steering angle.

The corresponding state representation of ego vehicle, $\dot{\mathbf{x}}_b$, is described by:

$$\dot{\mathbf{x}}_b(t) = \begin{bmatrix} \dot{x}_b(t) \\ \dot{y}_b(t) \\ \dot{v}_b(t) \\ \dot{a}_b(t) \\ \dot{\gamma}_b(t) \\ \dot{\delta}_b(t) \\ \ddot{\delta}_b(t) \end{bmatrix} = \underbrace{\begin{bmatrix} v_b(t) \cdot \cos(\gamma_b(t)) \\ v_b(t) \cdot \sin(\gamma_b(t)) \\ a_b(t) \\ j_b(t) \\ v_b(t) \cdot \frac{tan(\delta_b)}{L} \\ \dot{\delta}_b(t) \\ \ddot{\delta}_b(t) \end{bmatrix}}_{f_b \text{ : the nonlinear dynamics}}, \mathbf{u}_b(t) = \begin{bmatrix} j_b(t) \\ \ddot{\delta}_b(t) \end{bmatrix} \tag{2.11}$$

where $x_b$ and $y_b$ are the coordinates represented from the middle of the rear axle of the vehicle, $v_b$ is the longitudinal velocity, $a_b$ is the longitudinal acceleration, $j_b$ is the longitudinal jerk, $\gamma_b$ is the yaw, $\delta_b$ is the steering angle, and $L$ is the length of the vehicle.The control signal for the ego vehicle are $j$ and $\ddot{\delta}_b$, so the vehicle states are controlled by the longitudinal jerk and the steering angular acceleration.

Using the center of the rear axle as the reference point in a kinematic bicycle model provides a range of benefits, including simpler control dynamics, better stability, and ease of mathematical modeling. This setup is particularly effective for low-speed maneuvering, and applications where accurate lateral and heading control is crucial. Jerk control allows for more gradual changes in acceleration, which results in smoother vehicle motions. Besides, jerk control can help achieve more precise control of dynamic systems.

**Point mass model** Second-order accurate kinematic equations, in both longitudinal and lateral directions, are employed to model the motion of the ego vehicle

as a point mass. This approach provides a more precise linear model suitable for real-time control in simulations. The states and control inputs are defined as follows:

$$\begin{aligned} \mathbf{x}_p &= \begin{bmatrix} p_x & p_y & v_x & v_y & a_x & a_y \end{bmatrix}^\intercal \\ \mathbf{u}_p &= \begin{bmatrix} j_x & j_y \end{bmatrix}^\intercal \end{aligned} \tag{2.12}$$

where $p_x$ and $p_y$ denote longitudinal and lateral position, $v_x$ and $v_y$ denote longitudinal and lateral velocity, $a_x$ and $a_y$ denote longitudinal and lateral acceleration, $j_x$ and $j_y$ denote longitudinal and lateral jerk,respectively. The dynamics of the ego vehicle can be expressed in a linear, discrete state-space form as:

$$\mathbf{x}_p(k+1) = \mathbf{A}_p \mathbf{x}_p(k) + \mathbf{B}_p \mathbf{u}_p(k) \tag{2.13}$$

where

$$\mathbf{A}_p = \begin{bmatrix} 1 & 0 & dt & 0 & \frac{1}{2}dt^2 & 0 \\ 0 & 1 & 0 & dt & 0 & \frac{1}{2}dt^2 \\ 0 & 0 & 1 & 0 & dt & 0 \\ 0 & 0 & 0 & 1 & 0 & dt \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{B}_p = \begin{bmatrix} \frac{1}{6}dt^3 & 0 \\ 0 & \frac{1}{6}dt^3 \\ \frac{1}{2}dt^2 & 0 \\ 0 & \frac{1}{2}dt^2 \\ dt & 0 \\ 0 & dt \end{bmatrix} \tag{2.14}$$

Here $dt$ represents time interval.

This model reduces the vehicle's complexity by representing it as a single point with mass, making it computationally efficient and suitable for fast simulations and straightforward implementation

### 2.1.4 Hyperplane separation theorem

Soltan[6] introduced the Hyperplane Separation Theorem, which was later utilized by Fan et al[7]. to formulate smooth collision avoidance constraints by separating pairs of convex sets. This approach effectively reduces the size of the NLP problem and enhances computational performance in optimal control problems (OCP).

**Hyperplane separation theorem** Let $\mathbb{K}_1$ and $\mathbb{K}_2$ represent two compact convex sets in $\mathbb{R}^N$. Then,

$$\begin{aligned} \mathbb{K}_1 \cap \mathbb{K}_2 = \emptyset \iff \exists \boldsymbol{\lambda} \in \mathbb{R}^N \backslash \mathbf{0}, \mu \in \mathbb{R} : \\ \forall \boldsymbol{q} \in \mathbb{K}_1, \forall \boldsymbol{p} \in \mathbb{K}_2, \boldsymbol{\lambda}^\top \boldsymbol{q} \geq \mu, \boldsymbol{\lambda}^\top \boldsymbol{p} \leq \mu. \end{aligned} \tag{2.15}$$

According to Equation 2.16, when compact sets $\mathbb{K}_1$ and set $\mathbb{K}_2$ are disjoint, there exists a separating hyperplane between them, i.e., points belonging to $\mathbb{K}_1$ are on one side of the hyperplane while points of $\mathbb{K}_2$ are on the other side.

When the convex sets are polytopes, it is a much more efficient way to only check the vertices of the $\mathbb{K}_1$ and $\mathbb{K}_2$.

**Separation between two polytopes** Let sets $\mathbb{B}_m, \mathbb{O}_n$ be denoted using polytopes $P_1$ and $P_2$. Let $V_{P_1}$ and $V_{P_2}$ be matrices of vertices. Then,

$$\begin{aligned} \mathbb{B}_m \cap \mathbb{O}_n = \emptyset \iff \exists \boldsymbol{\lambda} \in \mathbb{R}^N, \mu \in \mathbb{R} : \\ \boldsymbol{\lambda}^\top V_{P_1} \geq \mu \mathbf{1}^\top, \quad \boldsymbol{\lambda}^\top V_{P_2} \leq \mu \mathbf{1}^\top, \quad \|\boldsymbol{\lambda}\| > 0. \end{aligned} \tag{2.16}$$

In the collision avoidance controller, both the ego vehicle and obstacles can be simplified as union of polytopes in a bird's-eye view. By using the separation between polytopes to construct the collision avoidance constraints in the MPC, the nonlinearity of the optimization problem is reduced, making this an efficient approach for obstacle avoidance. The illustration is shown in the Figure 2.3.

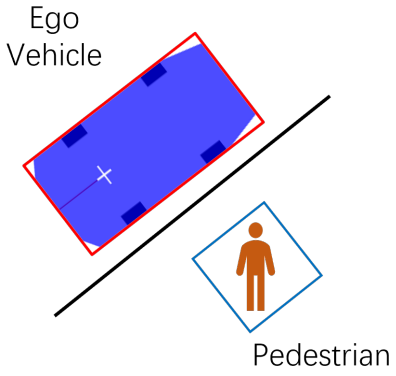



**Figure 2.3:** For readability, this figure illustrates the theorem in two dimensions. Pedestrian is in blue and ego vehicle is in red. Both of them are in type of polytopes.

### 2.1.5 Interior point optimizer

In this project, IPOPT is used as a solver for solving optimisation problems in MPCs. IPOPT (Interior Point OPTimizer)[8] is an open source optimisation algorithm designed to solve large-scale nonlinear programming problems.

The interior point method is a widely used algorithm for nonlinear optimisation. Instead of searching along the boundary of the feasible region, it finds the optimal solution by exploring the interior of the feasible space. This method has better convergence and numerical stability than traditional methods such as the sequential quadratic programming with active set methods, and is particularly suitable for large-scale problems.

IPOPT allows both the objective function and the constraints to be nonlinear or even nonconvex, making it very flexible in solving complex problems. Unlike algorithms that approximate nonlinearities through linear or quadratic models, IPOPT evaluates the nonlinear function directly, resulting in more accurate solutions. In addition, its internal scaling mechanism significantly improves numerical stability and efficiency, especially when dealing with variables of different sizes. The linear search filter method further improves the performance by reducing oscillations during the iteration process, resulting in faster convergence and improved robustness.

## 2.2 Polynomial regression

The polynomial regression model assumes a linear relationship between the input variables $X$ and the output variable $y$:

$$y = X\beta + \varepsilon$$

where:

- $y$ is an $n \times 1$ vector of observed values.
- $X$ is an $n \times p$ matrix of input features (with $n$ samples and $p$ features).
- $\beta$ is a $p \times 1$ vector of coefficients to be estimated.
- $\varepsilon$ is an $n \times 1$ vector of error terms (assumed to be normally distributed with mean zero).

The goal is to estimate $\beta$ such that the residual sum of squares (RSS) between the observed values and the values predicted by the linear model is minimized:

$$\min_{\beta} \text{RSS}(\beta) = \min_{\beta} \|\varepsilon\|^2 = \min_{\beta} \|y - X\beta\|^2 \tag{2.17}$$

where $\|\cdot\|^2$ denotes the squared Euclidean norm.

To find the value of $\beta$ that minimizes the RSS, we derive the normal equations.

The RSS function is quadratic in $\beta$, and its minimum can be found by setting its gradient with respect to $\beta$ to zero:

$$\frac{\partial}{\partial\beta}\text{RSS}(\beta) = -2X^\top(y - X\beta) = 0 \tag{2.18}$$

Solving for $\beta$:

$$X^\top X\beta = X^\top y \tag{2.19}$$

Assuming $X^\top X$ is invertible, the solution is:

$$\hat{\beta} = (X^\top X)^{-1}X^\top y \tag{2.20}$$

This solution minimizes the RSS and provides the best linear unbiased estimator under the Gauss-Markov theorem.

## 2.3 Bézier curves

Bézier curves are commonly used in path planning for autonomous vehicles due to their smoothness and flexibility. In this section, Bézier curve of degree $n$ is defined as a parametric curve constructed using a set of control points. The curve is given by the following equation:

$$\mathbf{B}(t) = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i} t^i \mathbf{P}_i \tag{2.21}$$

where:

- $\mathbf{B}(t)$ is the position vector on the curve at parameter $t \in [0, 1]$.
- $\mathbf{P}_i$ are the control points in 2D or 3D space.
- $\binom{n}{i}$ is the binomial coefficient, given by:

$$\binom{n}{i} = \frac{n!}{i!(n-i)!} \tag{2.22}$$

- $t$ is the parameter, where $t = 0$ corresponds to the start of the curve and $t = 1$ corresponds to the end of the curve.
- $n$ is the degree of the Bézier curve, determined by the number of control points $n + 1$.

The weights in the Bézier curve are determined by Bernstein basis polynomials of degree $n$:

$$B_{i,n}(t) = \binom{n}{i}(1-t)^{n-i}t^i \tag{2.23}$$

Thus, the Bézier curve can also be written as:

$$\mathbf{B}(t) = \sum_{i=0}^{n} B_{i,n}(t)\mathbf{P}_i \tag{2.24}$$

**Cubic Bézier curves** Four points $\mathbf{P}_0$, $\mathbf{P}_1$, $\mathbf{P}_2$ and $\mathbf{P}_3$in the plane or in higher-dimensional space define a cubic Bézier curve. As the Figure 2.4 shows, the curve starts at $\mathbf{P}_0$ going toward $\mathbf{P}_1$ and arrives at $\mathbf{P}_3$ coming from the direction of $\mathbf{P}_2$. The explicit form of the curve is:

$$\mathbf{B}(t) = (1-t)^3\mathbf{P}_0 + 3(1-t)^2t\mathbf{P}_1 + 3(1-t)t^2\mathbf{P}_2 + t^3\mathbf{P}_3, \quad 0 \le t \le 1 \tag{2.25}$$

The derivative of the cubic Bézier curve with respect to t is as follows, and its geometric meaning is the direction of the curve at any given point.

$$\mathbf{B}'(t) = 3(1-t)^2(\mathbf{P}_1 - \mathbf{P}_0) + 6(1-t)t(\mathbf{P}_2 - \mathbf{P}_1) + 3t^2(\mathbf{P}_3 - \mathbf{P}_2), \quad 0 \le t \le 1 \tag{2.26}$$

The second derivative of the Bézier curve with respect to t is:

$$\mathbf{B}''(t) = 6(1-t)(\mathbf{P}_2 - 2\mathbf{P}_1 + \mathbf{P}_0) + 6t(\mathbf{P}_3 - 2\mathbf{P}_2 + \mathbf{P}_1) \tag{2.27}$$

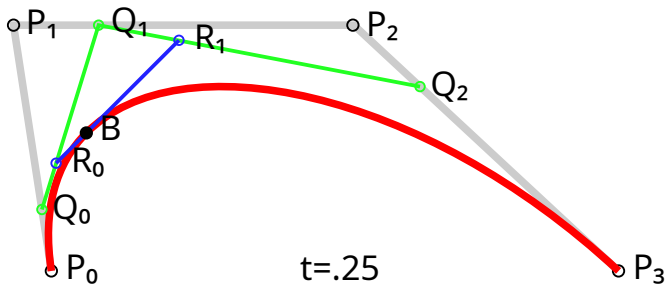



**Figure 2.4:** Construction of a cubic Bézier curve. B at the point when t is 0.25. Intermediate points $\mathbf{Q}_i$ describe linear Bézier curves and intermediate points $\mathbf{R}_i$ describe quadratic Bézier curves.

**Curvature of Bézier curves** The curvature of Bézier curves $\kappa$ could be calculated using the following formula:

$$\kappa = \frac{|\mathbf{B}'(t) \times \mathbf{B}''(t)|}{|\mathbf{B}'(t)|^3} \tag{2.28}$$

The curvature of the curve is of significant importance for verifying the feasibility of the planned path.

## 2.4 Autonomous emergency braking system

Mattias, Erik and Jonas [2] introduces an algorithm, which approximates potential evasive maneuvers and derives analytical expressions to estimate the range of possible driver actions to avoid a collision. This algorithm assesses whether the driver needs immediate assistance to prevent or mitigate an accident. Inspired by this related work about CAS, an AEB system is designed in the thesis and used as a comparison for the experimental session. The autonomous control system continuously predicts whether a detected pedestrian is about to appear in the front area of ego vehicle laterally in the time horizon, as shown in Figure 2.5:

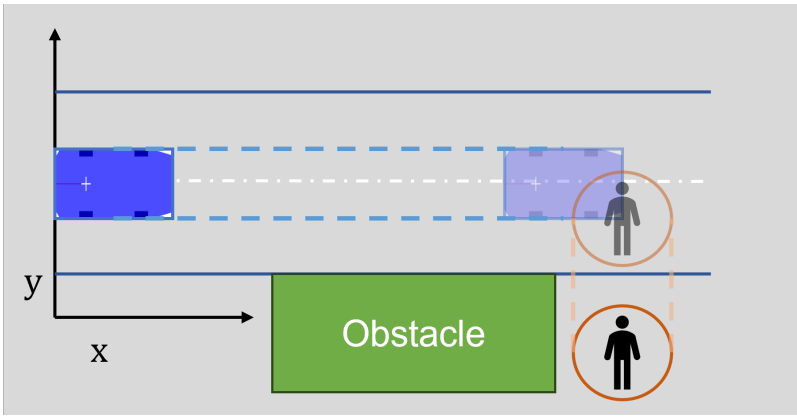



**Figure 2.5:** Predicted positions of ego vehicle and pedestrian

In the prediction phase, the motion of the ego vehicle is described by a CA model and the motion of the pedestrian is described by a CV model. Then the required acceleration $a_{\text{req}}$ is calculated through:

$$a_{\text{req}} = \frac{2 \times (x_P - x_E - v_E \cdot t - r_p - 0.5 \cdot \text{length}_E - \epsilon)}{t^2} \tag{2.29}$$

where $t$ is the detected time, $x_E, x_P$ are longitudinal positions of vehicle and pedestrian respectively, $v_E$ is longitudinal velocity of vehicle, $r_P$ is range of pedestrian motion, $\text{length}_E$ is the length of vehicle, and $\epsilon$ is a constant distance margin. If $a_{\text{req}}$ is less than $-5m/s^2$ (the limitation of control system), braking system would be activated immediately.

# 3

# Methodology of Precautionary Safety Function

To provide a more foresighted active safety function, the precautionary safety function is designed to respond to potential collision risks. The essential aim of the precautionary safety function is to ensure that the ego vehicle can always avoid potential threats using emergency maneuvers. This requires identifying potential collision risks in the environment and calculating a safety-set of vehicle states based on the threat assessment. The safety-set represents motion restrictions under which the vehicle can safely avoid collision along its predicted path. By ensuring the vehicle remains within this safety-set, the system guarantees that it can successfully execute emergency actions, such as AEB. When used alongside traditional active safety functions, the precautionary safety function significantly enhances the vehicle's ability to avoid potential hazards.

## 3.1 Overview of precautionary safety function

**Architecture** The Architecture of the precautionary safety function could be seen as the following Figure 3.1. The general approach of this method mainly contains two tasks: **Calculation of safety-set**, **Vehicle control based on safety-set**. Computing the safety-set involves the following steps: first, spatial sampling is performed; Next, since the safety-set is defined as a region in the three-dimensional state space of lateral position, longitudinal position, and velocity, the critical safety velocities at the sampled points are calculated to define the boundary of this region. Finally, polynomial regression is applied to fit the boundary points with a continuous function, representing the safety-set.

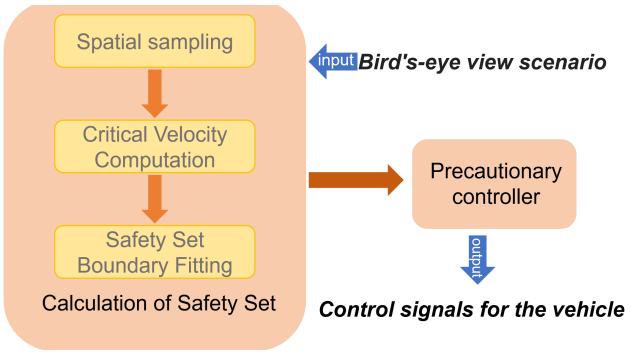



**Figure 3.1:** A general architecture of precautionary safety function.

The specific details of implementing these steps will be explained below. In some cases, the exact methods may vary depending on the scenario.

**Scenarios** This project investigated two scenarios: the Hidden Pedestrian Scenario and the Lane Change Scenario. The primary focus was addressing the challenges in the Hidden pedestrian scenario, where the entire process for the precautionary safety function was developed and thoroughly analyzed through experiments. To demonstrate the general applicability of the proposed approach and to enhance the methods, the precautionary safety function was also applied to the lane change Scenario, albeit with fewer experiments.

As shown in the Figure 3.2, in the Hidden Pedestrian Scenario, a large obstacle on the side of the road blocks the view, and a pedestrian may cross the street at any moment from the blind spot. We need to ensure that the ego vehicle can completely avoid colliding with pedestrians

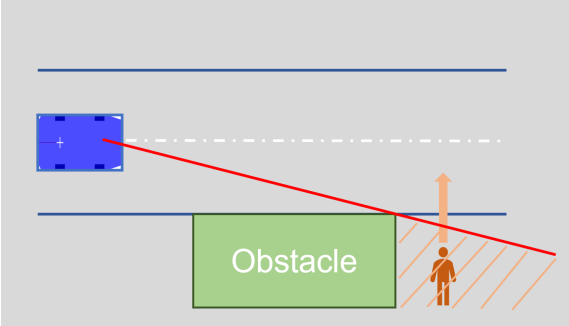



**Figure 3.2:** The Hidden Pedestrian Scenario.

As shown in the Figure 3.3, in the Lane Change Scenario, the ego vehicle is attempting to merge into the lane below. A vehicle is approaching from behind in the target lane and may suddenly accelerate or brake. Our goal is to avoid a collision with this vehicle during the lane change.

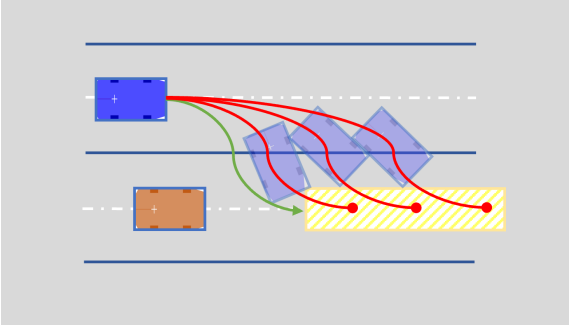


**Figure 3.3:** The Lane Change Scenario

## 3.2 Spatial sampling

Defining the safety set begins with sampling positions in regions where potential risks may exist. By analyzing these samples, the features of the safety set within the area can be determined.

In the Hidden Pedestrian scenario, the sampling on a grid method is employed. A grid is drawn over the road ahead of the blind spot, and samples are taken to represent the entire drivable area. This approach is straightforward and effective.

The safety-set is defined only on the basis of the vehicle's position and velocity, without taking into account other states such as heading and steering angle. In hidden pedestrian scenarios, where vehicles are mainly travelling straight ahead, this simplification has little impact, but in scenarios where precise turning dynamics are required, such as lane changing, this simplification is insufficient. To address this limitation, we introduce the along-path sampling method. This method integrates path planning and sampling by first planning multiple paths and then sampling along these paths. This method improves adaptability and generality in a variety of situations.

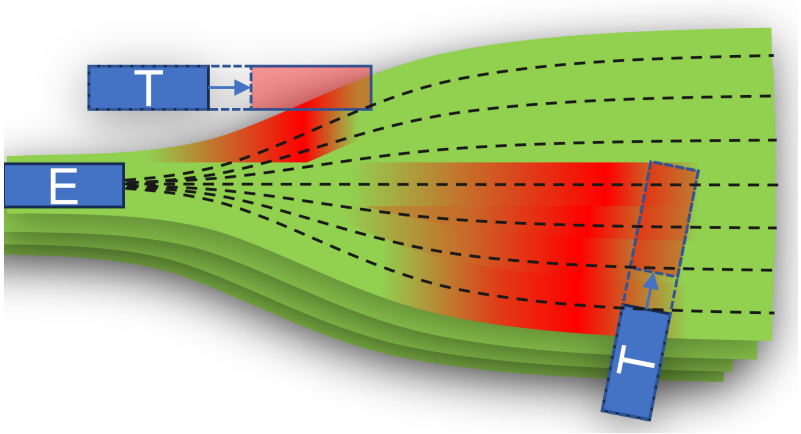



**Figure 3.4:** The E represents ego vehicle, T represents target vehicles. By sampling along the paths, the threat assessment is performed on the region of possible paths, taking into account the vehicle attitude.

### 3.2.1 Sampling on a grid

In order to obtain sample points, the sampling on a grid method was employed in the context of a hidden pedestrian scenario.

For an ego vehicle traveling on the lane before the blind zone, the entire area ahead should be drivable. This area is rectangular in shape, with the length starting at the front of the vehicle and ending at the rear of the obstacle, and the width being the same as the width of the lane.

Typically, the risk increases closer to hidden areas, so a logarithmically spaced vector is generated within this region, with denser sampled points nearer to the hidden areas. This arrangement is illustrated in the figure below:

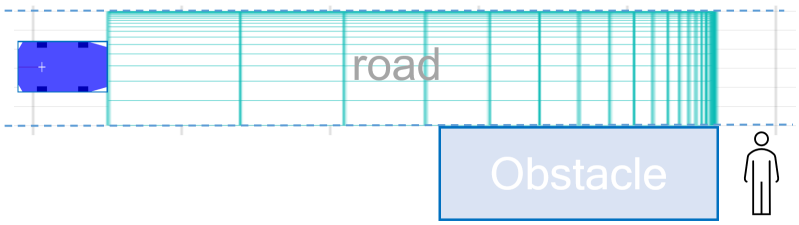



**Figure 3.5:** Sampling through logarithmically spaced vector. Each dot on the cyan grid is a sampled point.

### 3.2.2 Sampling along the paths

For the lane change scenario, we first generate multiple driving paths using Bézier curves and evaluate the feasibility of each path. Then, for the feasible paths, we uniformly sample points along the trajectories, which are used as the sampled points for calculating the safety-set.

As shown in Figure 3.3, in this bird's eye view, the ego vehicle and the target vehicle are both traveling from left to right. The driving goal of the ego vehicle is to transition from the center of the upper lane to the center of the lower lane. Therefore, at each time instance, we plan 10 different lane-change trajectories, along with another 10 trajectories to capture the safety characteristics of the entire drivable area as comprehensively as possible. We then sample points along these trajectories, as illustrated in the Figure 3.6.

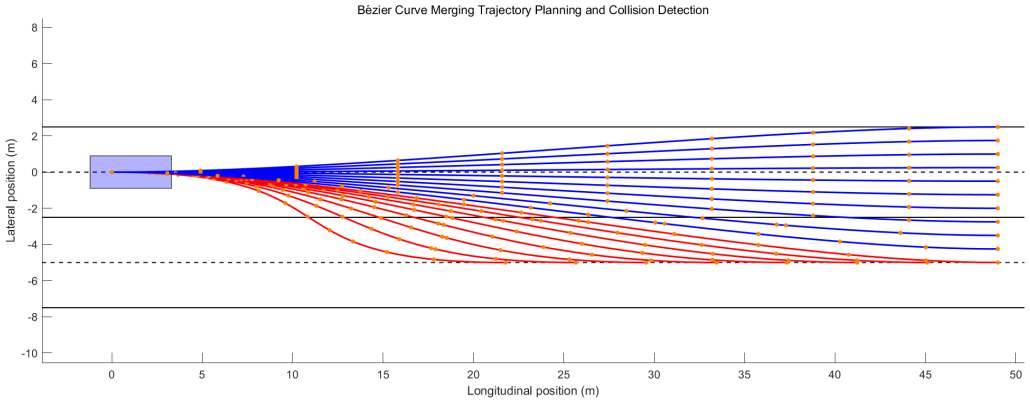



**Figure 3.6:** Blue rectangle represents the ego vehicle, red curves indicate lane-change trajectories, blue curves depict other possible trajectories, and orange points represent the sampled points

**Bézier curve for path planning** Specifically, each path is generated using a Cubic Bézier curve. The predicted lane-change trajectory starts from the center line of the upper lane and merges into the center line of the lower lane. At the start, the trajectory is tangent to the vehicle's current direction, and by the end, it is tangent to the center line of the lower lane. This requires the control points of the third-order Bézier curve to be defined as follows: $P_0$ is the current position of the vehicle, $P_1$ lies along the vehicle's current direction, and both $P_2$ and $P_3$ are on the target lane's center line, with $P_3$ representing the position where the vehicle merges into the lane. The Figure 3.7 below shows the predicted trajectory at the beginning of the lane change.

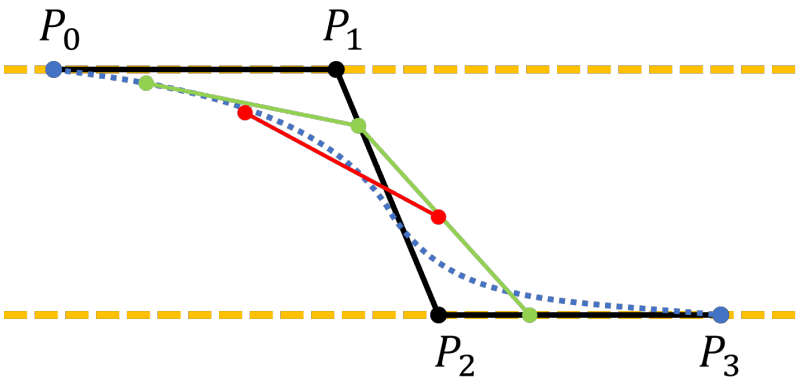



**Figure 3.7:** $P_0$ is the starting point for ego vehicle and $P_3$ is the ending point. $P_1$ and $P_2$ are also control points along the path. The yellow dash lines are the current center line and target center line

Assume that the ego vehicle's initial longitudinal velocity is $V_{init}$. The longitudinal distance between points $\mathbf{P}_0$ and $\mathbf{P}_3$ is given by $S = V_{init} \cdot T$, where $T \in [2, 10]$. By selecting ten values of $T$ within this range, ten different lane-changing trajectories can be generated. Furthermore, the longitudinal distance between $\mathbf{P}_0$ and $\mathbf{P}_1$ (as well as between $\mathbf{P}_2$ and $\mathbf{P}_3$) is defined as $S_{control} = V_{init} \cdot 2$. The corresponding mathematical representation of the curve can be expressed as follows:

$$\begin{aligned}
&\mathbf{B}(t) = (1-t)^3\mathbf{P}_0 + 3(1-t)^2 t\mathbf{P}_1 + 3(1-t)t^2\mathbf{P}_2 + t^3\mathbf{P}_3, && 0 \le t \le 1 \\
&\mathbf{B}'(t) = 3(1-t)^2(\mathbf{P}_1 - \mathbf{P}_0) + 6(1-t)t(\mathbf{P}_2 - \mathbf{P}_1) + 3t^2(\mathbf{P}_3 - \mathbf{P}_2), && 0 \le t \le 1 \\
&\mathbf{B}''(t) = 6(1-t)(\mathbf{P}_2 - 2\mathbf{P}_1 + \mathbf{P}_0) + 6t(\mathbf{P}_3 - 2\mathbf{P}_2 + \mathbf{P}_1), && 0 \le t \le 1
\end{aligned} \tag{3.1}$$

where $\mathbf{B}(t)$ represent for the generated path and its derivative $\mathbf{B}'(t)$ represent for the direction of the ego vehicle on this path, and $\mathbf{B}''(t)$ could be useful for calculating the curvature.

**Feasibility of the paths** It is also essential to evaluate whether the generated trajectories are feasible for the vehicle to follow. To simplify the problem, we examine whether the vehicle can safely pass the point of maximum curvature for each

trajectory at the initial velocity. This can be determined by checking if the vehicle's maximum lateral friction force is sufficient to provide the necessary centripetal force along the path.

Our path expression can be specifically represented by the following equation in the Cartesian coordinate system:

$$\begin{aligned} &\text{For the x-coordinate:} x^{\mathrm{c}}(t) = (1-t)^3 x_0^{\mathrm{c}} + 3(1-t)^2 t x_1^{\mathrm{c}} + 3(1-t)t^2 x_2^{\mathrm{c}} + t^3 x_3^{\mathrm{c}} \\ &\text{For the x-coordinate:} y^{\mathrm{c}}(t) = (1-t)^3 y_0^{\mathrm{c}} + 3(1-t)^2 t y_1^{\mathrm{c}} + 3(1-t)t^2 y_2^{\mathrm{c}} + t^3 y_3^{\mathrm{c}} \end{aligned} \tag{3.2}$$

At this point, the curvature at each location on the path is given by:

$$\kappa(t) = \frac{\left| \frac{dx^{\mathrm{c}}(t)}{dt} \cdot \frac{d^2 y^{\mathrm{c}}(t)}{dt^2} - \frac{dy^{\mathrm{c}}(t)}{dt} \cdot \frac{d^2 x^{\mathrm{c}}(t)}{dt^2} \right|}{\left( \frac{dx^{\mathrm{c}}(t)}{dt}^2 + \frac{dy^{\mathrm{c}}(t)}{dt}^2 \right)^{3/2}} \tag{3.3}$$

By selecting the maximum curvature along the curve, we verify the following equation. If the condition is satisfied, the path is considered feasible:

$$v_{init}^{c}{}^2 \kappa_{max} \leq G \tag{3.4}$$

Here, we have $v_{init}^{c}$ as the initial velocity when entering the curve, $K_{max}$ as the maximum curvature, and the maximum lateral friction coefficient provided by the ego vehicle is assumed to be $G$.

## 3.3 Critical safety velocity computation

The safety-set represents the constraints on vehicle states that ensure the continued existence of safety. When defining this set, we focus only on the **lateral position**, **longitudinal position** and **velocity of the vehicle**. The safety-set can be represented as a volume within a three dimensional state space. The goal of precautionary safety can also be seen as keeping the vehicle states within the safety-set.

With the sampling positions in the evaluation area, we only need to determine the corresponding critical velocities to outline the boundary of the safety-set, as illustrated in the Figure 3.8. To calculate the critical velocity required for obstacle avoidance, we first need to model the target to be avoided and define its claim area.

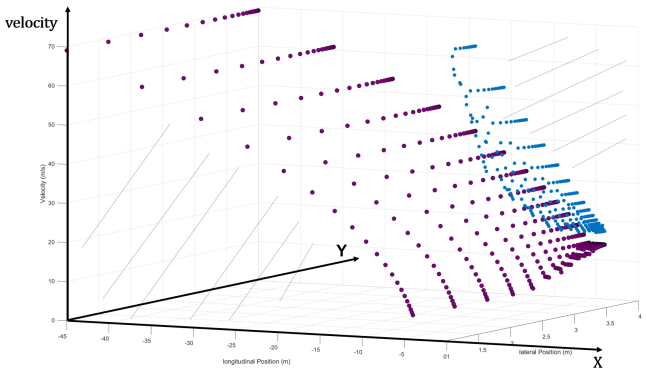



**Figure 3.8:** The blue and purple points represent the boundary of the safety-set. The grey dashed area above the blue boundary and below the purple boundary indicates the safety-set.

### 3.3.1 Modeling of claim area

The claim area of a target refers to the region that the target may potentially occupy during the prediction time. With the claim area model established, our problem can be transformed into avoiding overlap with the target's claim area.

**Claim area of hidden pedestrian** The centre of the vehicle sensor is assumed to be in the middle forward, in the first scenario the sight line emanates from the centre of the sensor, this triangular area is the part of the vehicle that cannot be sensed by the vehicle and changes with the position of the vehicle.

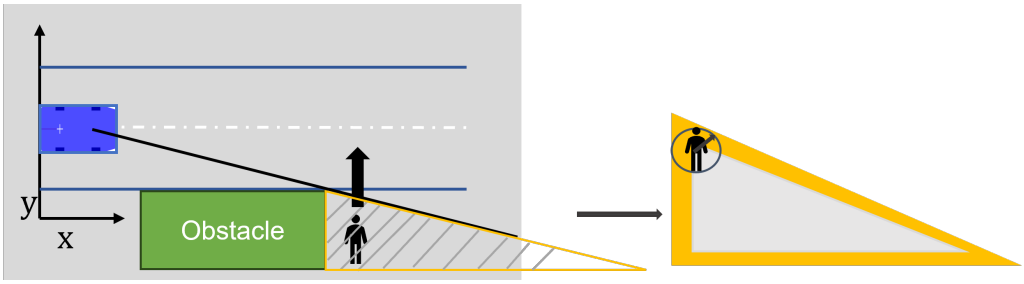



**Figure 3.9:** The hidden area is in type of triangle of which edge is in yellow. The right part of this figure details the possible area where the pedestrian could appear, which is also of shape in triangle and fulfilled with gray.

The motion radius of a pedestrian is $r_p$. The distances between the vertices of internally similar triangles and the adjacent vertices of external triangles are all $r_p$. The initial position of the hidden pedestrian is uniformly distributed in the inner triangle and the initial velocity is also uniformly distributed between 0 and $v^p$. The position of the pedestrian can be expressed by constant velocity model as:

$$\begin{bmatrix} p_y^p \\ p_x^p \end{bmatrix}_k = \begin{bmatrix} p_y^p \\ p_x^p \end{bmatrix}_{k-1} + \begin{bmatrix} dt & 0 \\ 0 & dt \end{bmatrix} \begin{bmatrix} v_y^p \\ v_x^p \end{bmatrix}_{k-1} \tag{3.5}$$

where $k$ represent the time step. With the constant velocity model, the progression of the space occupied by pedestrians over time can be well calculated by forward reachability analysis. The initial pedestrian position can be regarded as the initial set and the distribution of velocitys could be used as a constraint for the state.

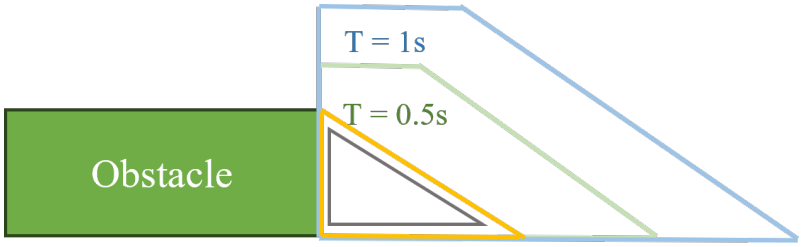



**Figure 3.10:** A general illustration of forward reachable sets of hidden pedestrian. The green polygon shows the area of the hidden area at 0.5s in the future, and the blue polygon shows the area of the hidden area at 1s in the future.

Collision avoidance focuses on the ego vehicle's responsibility to proactively mitigate risk. However, relying solely on the ego vehicle's efforts is insufficient to prevent collisions. For instance, if the ego vehicle stops in front of a pedestrian, the pedestrian might still collide with the vehicle's side due to an inability to stop. The goal is to prevent collisions where the ego vehicle actively hits the pedestrian, as solving the feasibility is necessary. Therefore, the velocity limit for pedestrians is established as follows:

$$\alpha_{pedes} = \frac{1}{1 + exp(-100 \times (v^e - 0.1))} \tag{3.6}$$

$$\begin{bmatrix} v_y^p \\ v_x^p \end{bmatrix} = \alpha_{pedes} \begin{bmatrix} v_{ymax}^p \\ v_{xmax}^p \end{bmatrix} \tag{3.7}$$

where $v_x^p$ and $v_y^p$ denote the longitudinal and lateral velocity constraints of the pedestrian, respectively. $v_{ymax}^p$ and $v_{xmax}^p$ represent the maximum velocitys of the pedestrian's movement, and $v^e$ represents the velocity of the ego vehicle. $\alpha_{pedes}$ is derived from a modified sigmoid function. This approach assumes that when the ego vehicle's velocity is zero, the pedestrian will also stop moving. This prevents scenarios where the pedestrian actively collides with the stationary vehicle.

**Claim area of target vehicle** For the lane change scenario, we consider the target vehicle's reasonable worst-case behaviors within the prediction horizon $T_{claim}$, including sudden braking and acceleration. An example claim area is illustrated in the figure below:

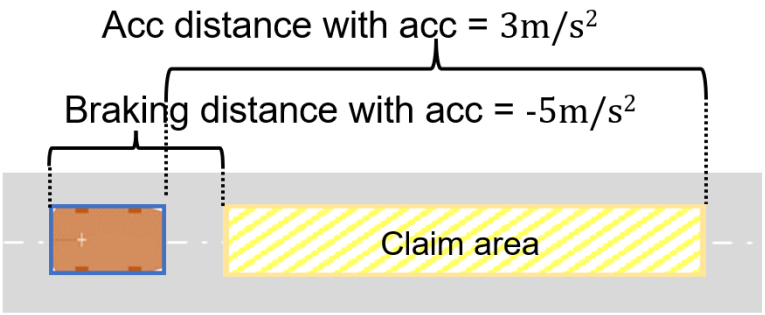



**Figure 3.11:** The predicted claim area of a target vehicle.

With the initial velocity of the target vehicle $v_0^t$, and the Braking deceleration $a_b$ and acceleration rate$a_a$. The braking distance $D_b$ and the accelerating distance $D_a$ would be:

$$D_b = \begin{cases} v_o^t T_{claim} + \frac{1}{2} a_b {T_{claim}}^2 & \text{if } -a_b T_{claim} < v_0^t \\ -\frac{1}{2} a_b {T_{claim}}^2 & \text{if } -a_b T_{claim} > v_0^t \end{cases}$$
$$D_a = v_o^t T_{claim} + \frac{1}{2} a_a {T_{claim}}^2 \tag{3.8}$$

### 3.3.2 Optimization based computation

From a control theory perspective, if an emergency avoidance MPC controller is used to represent the vehicle's emergency manoeuvre during obstacle avoidance, ensuring that the MPC problem remains feasible indicates that there is an emergency strategy to avoid a collision. Therefore, the goal of the precautionary safety function can also be seen as ensuring that the emergency MPC remains continuously feasible. The safety-set can be understood as the N-step controllable set defined in terms of the vehicle's position and velocity. Since the feasibility of the optimisation problem depends only on the constraints, the safety-set can be determined indirectly by designing the optimization problem with the same constraints as the emergency MPC. The constraints for the emergency MPC typically include the vehicle dynamics model, various vehicle motion limitations and obstacle avoidance constraints.

In the hidden pedestrian scenario, the ego vehicle has two options when approaching a potentially dangerous area, as shown in Figure 3.12. The ego vehicle can either maintain a high enough initial velocity to pass through the danger zone before the hazard occurs, as shown by the blue path in the left figure, or travel at a low enough velocity to avoid the hazard by slowing down, as illustrated by the purple path in the same figure. From the right figure, it can be seen that the safety set is the union of the upper and lower safety-sets.

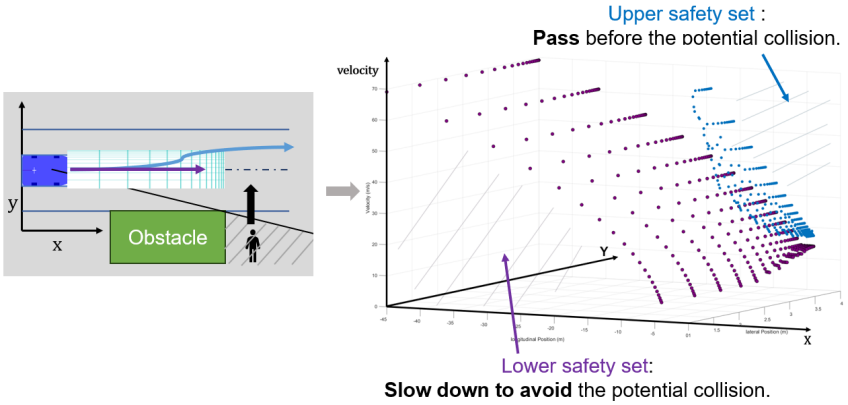



**Figure 3.12:** Two Strategies for Mitigating Hazards in hidden pedestrian scenario.

**Maximum initial velocities to slow down safely** For the lower set, an optimization problem (which could also be viewed as an MPC problem) is deigned to compute the maximum initial velocities that allow the vehicle to slow down and avoid hazards at each sampled point, defining one boundary of the safety-set.

$$\begin{aligned} \min_{\mathbf{u}(k),\forall k\in[1,N]} \quad & \mathcal{J}_s^l = -v_0^e + \sum (v_s^e(k) - v_0^e) \\ \text{s.t.} \quad & \mathbf{x_s^e}(k+1) = f_b(\mathbf{x_s^e}(k), \mathbf{u_s^e}(k)) \\ & g_s(\mathbf{x_s^e}(k), \mathbf{u_s^e}(k)) \leq 0 \\ & \mathbf{x_s^e}(k) \in \mathbb{X}_s^e, \quad \mathbf{u_s^e}(k) \in \mathbb{U}_s^e \\ & v_s^e(N) = 0 \end{aligned} \tag{3.9}$$

- By solving the optimization problem and minimizing the objective function $\mathcal{J}_s^l$, we can obtain the maximum initial velocity by minimizing the negative initial velocity, and the most aggressive braking behavior at each prediction step by minimizing the negative reduction in velocity relative to the initial velocity.
- $\mathbf{x_s^e}$ represents the vector of ego vehicle's states for calculating safety-set, $\mathbf{u_s^e}$ represents the vector of control input.
- $f_b$ represents the equality constraints, describing the system dynamics, which are modeled using the kinematic bicycle model.
- $g_s$ represents the inequality constraints, which include the use of the hyperplane separation theorem to separate the ego vehicle from the claim area, as shown in the figure below. This serves as the obstacle avoidance constraint.

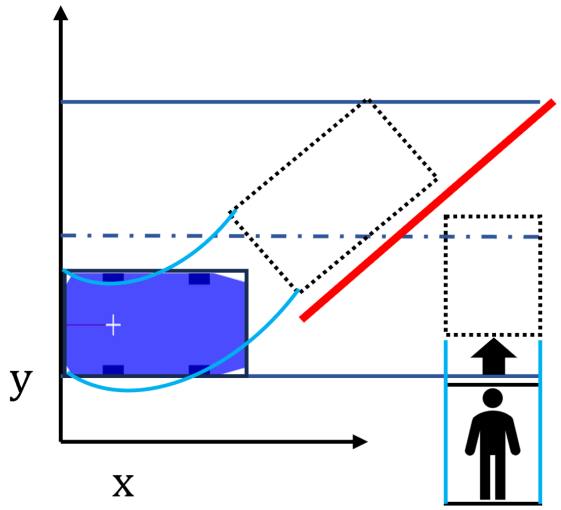



**Figure 3.13:** Hyperplane separation theorem for collision avoidance with pedestrian claim area.

- $\mathbf{x}(k) \in \mathbb{X}_s^e, \quad \mathbf{u}(k) \in \mathbb{U}_s^e$ These constraints represent the vehicle's performance limitations or restrictions imposed by road conditions, such as velocity limits, maximum acceleration, and others. Some of these constraints are as follows:

| State or input | Lower bound | Upper bound |
|---|---|---|
| $v$ | 0 | 19.44 $m/s$ |
| $a$ | $-5$ $m/s^2$ | 0 |
| $j$ | $-50$ $m/s^3$ | 50 $m/s^3$ |
| $\gamma$ | $-\frac{55\cdot\pi}{180}$ | $\frac{55\cdot\pi}{180}$ |
| $\delta$ | $-\frac{45\cdot\pi}{180}$ | $\frac{45\cdot\pi}{180}$ |
| $\dot{\delta}$ | $-\frac{20\cdot\pi}{180}$ | $\frac{20\cdot\pi}{180}$ |
| $\ddot{\delta}$ | $-\frac{80\cdot\pi}{180}$ | $\frac{80\cdot\pi}{180}$ |

**Table 3.1:** Operational constraints for some states or inputs

- It is important to note that the final vehicle velocity is set to zero, indicating that the ego vehicle has come to a complete stop through steering and braking in order to avoid collision with the pedestrian.

Here are the results of the optimization problems at the sampled points, representing the boundary of the lower safety set. If the vehicle's state is below this boundary, it indicates that the vehicle can respond to a suddenly appearing pedestrian by slowing down.

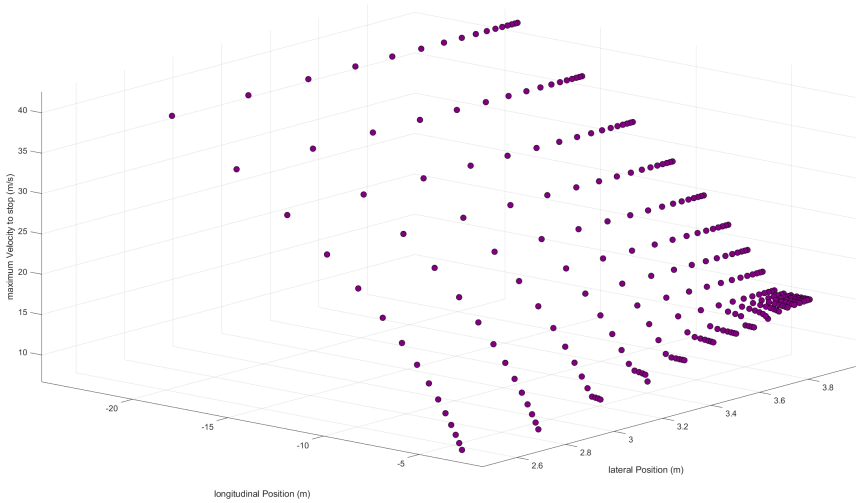



**Figure 3.14:** The lower safety-set for hidden pedestrian scenario.

**Minimum initial velocities for safe passage** For the upper set, the optimization problem is designed to obtain the minimum initial velocity for the ego vehicle to pass the area before collision. Here we made two changes to the Formulation 3.9:

- The objective function is set as: $\mathcal{J}_s^u = v_0^e$.
  By solving optimization problem, we could minimize the initial velocity at each point.
- The final constraint is set as: $x^e(N) \geq X_{final}^p$ instead of the constraint on $v^e(N)$.
  This ensures that the vehicle ultimately passes through the hazardous area.

And here is the result of the optimization problems on sampled points, representing the boundary of upper safety-set. If the vehicle's state is above this boundary, it means the vehicle can pass through the hazardous area at its current velocity without a collision.

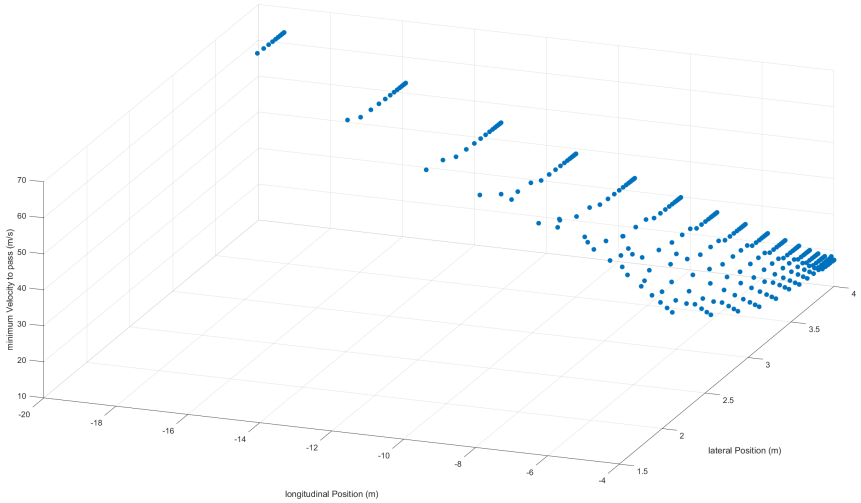



**Figure 3.15:** The upper safety-set for hidden pedestrian scenario.

### 3.3.3 Analytical calculation

Notably, the optimisation-based method requires significant computational time. Solving the complete safety-set requires solving 120 optimisation problems, which takes unreasonable time for online calculation. As the longitudinal position of the obstacle creating the blind spot remains constant, offline computation is possible. However, if the road scenario or the blind spot changes, recalculating the safety-set online becomes impractical. By analysing the scenario and applying kinematic principles, an analytical calculation of the safety-set can be derived. This could enable real-time safety-set calculation and have significant practical applications.

**Analytical Calculation for hidden pedestrian scenario** The lower set could be seen as to solve two problems:

- The critical velocity is the velocity at which a vehicle can safely come to a stop before reaching the pedestrian's longitudinal position.
- The critical velocity is the velocity at which a vehicle can come to a stop before the pedestrian reaches the vehicle's lateral position.

In the first case, the required velocity for the ego vehicle to stop just before entering the hidden area is calculated. Due to the analytical complexity of solving the jerk, a reaction time parameter $t_r$ is introduced to approximate the vehicle's dynamic response. By adjusting this parameter, the resulting solution closely aligns with the optimal solution. The formulation for the sampled point $k$ is presented as follows:

$$\begin{aligned} &\begin{cases} S &= p_x^p - x_k^e \\ S &= v_k^e t_r + v_k^e t + \frac{1}{2} a t^2 \\ t &= -\frac{v_k^e}{a} \end{cases} \\ &v_k^e = a * t_r - a\sqrt{t_r^2 + \frac{2(x_k^e - p_x^p)}{a}} \end{aligned} \tag{3.10}$$

Where $p_x^p$ represents the position of the left side of the pedestrian region, $x_k^e$ is the longitudinal position of the ego vehicle, $v_k^e$ denotes the critical velocity, and $a$ is the ego vehicle's minimum acceleration, $t_r$ is reaction time.

For the second case, we first calculate the time $t_{pedestrian}$ it takes for the pedestrian to reach the vehicle's lateral position.

$$t_{\text{pedestrian}} = \frac{y_k^e - B}{v_y^p}, \quad t_{\text{jerk}} = \frac{a}{j} \tag{3.11}$$

Here, $y_k^e$ is the lateral position of the ego vehicle,$v_y^p$ is the lateral velocity of the pedestrian, $B$ is the half width of the ego vehicle.

Then, considering the effect of jerk, we compute the maximum velocity at which the vehicle can brake within this time.

$$\begin{cases} t_{\text{pedestrian}} \geq t_{\text{jerk}} : v_k^e &= -\frac{j}{2} t_{\text{jerk}}^2 - a(t_{\text{pedestrian}} - t_{\text{jerk}}) \\ t_{\text{pedestrian}} \leq t_{\text{jerk}} : v_k^e &= -\frac{j}{2} t_{\text{jerk}}^2 \end{cases} \tag{3.12}$$

Here is the result of the analytical solution compared to the optimal solution:

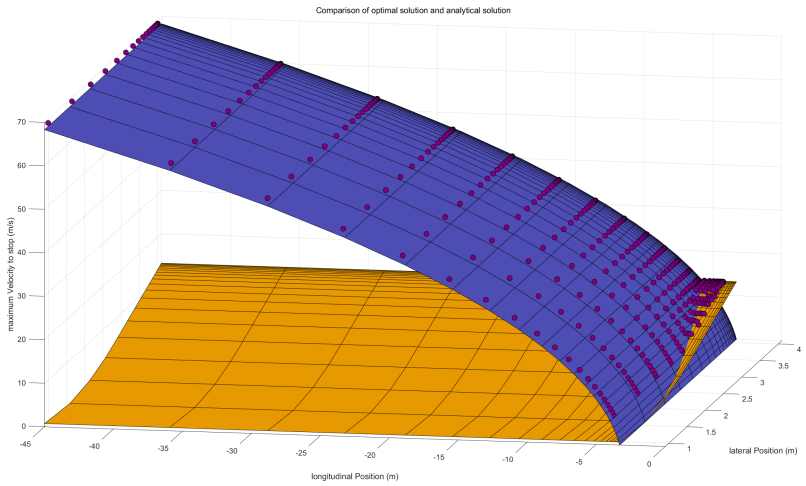



**Figure 3.16:** The comparison between analytical solution and optimal solution. The points are the lower set of optimal solution, and the purple and orange surfaces correspond to the analytical solutions of the two cases, respectively.

The upper set could be seen as trying to pass the area before the pedestrian could reach the lateral position of the ego vehicle, with the $V_{\text{pedestrian}}$, the expression is as followed:

$$v_k^e = v_x^p + \frac{S}{t_{\text{pedestrian}}} \tag{3.13}$$

Here the $v_x^p$ is the longitudinal velocity of the pedestrian. The result is as follows in Figure 3.17:

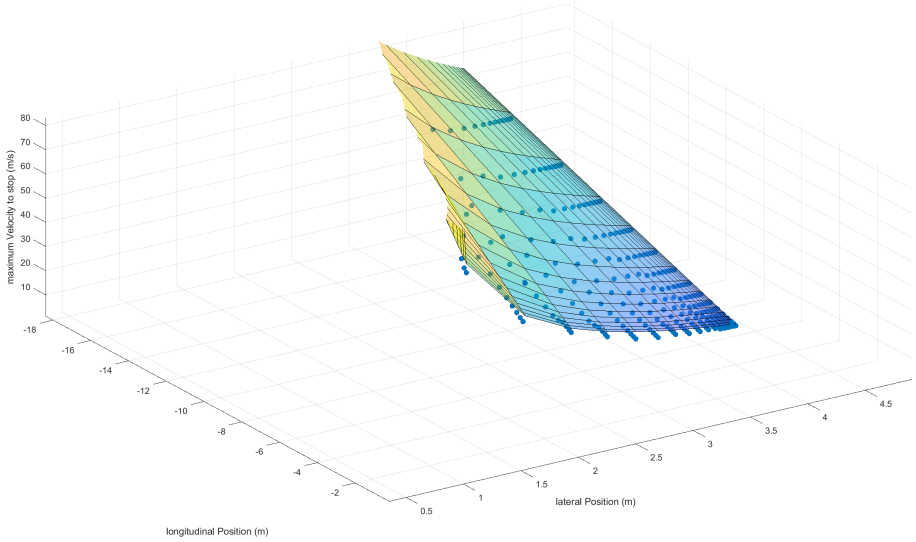



**Figure 3.17:** The comparison of upper set between analytical solution and optimal solution. The points are the upper set of optimal solution, and the surfaces correspond to the analytical solutions.

**Analytical Calculation for lane change scenario** In the lane change scenario, since the lateral position and velocity of the approaching vehicle may continuously change, the safety-set needs to be recalculated at each sampling time. Therefore, this scenario is solved online using analytical calculations.

We calculate the critical velocity by analyzing the relationship between the planned path and the target vehicle's claim area, as illustrated in the Figure 3.18.

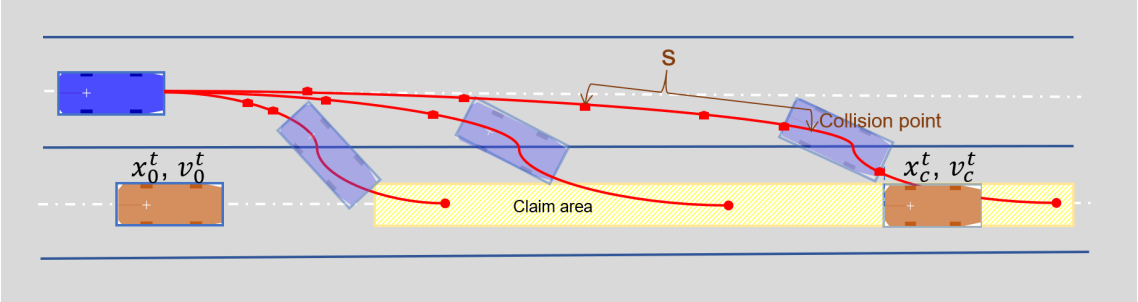



**Figure 3.18:** The blue rectangle represents the ego vehicle, while the orange rectangle represents the target vehicle approaching from behind. The yellow area indicates the claim area of the target vehicle over a horizontal time $T$. The red line represents examples of the planned paths, with the light blue rectangles along it denoting the collision points on the paths.

For a given planned path, collision points are identified where the ego vehicle's trajectory intersects with the claim area. At these collision points, it is assumed that a collision can be avoided if the ego vehicle's remains below the velocity of the target vehicle associated with the claim area ahead of the position. The critical velocity of the ego vehicle at a collision point is therefore defined as the velocity of the target vehicle immediately ahead of the collision point.

$$v_c^e = v_c^t = \max(\frac{2 \cdot (x_c^t - x_0^t)}{T_{claim}} - v_0^t, 0) \tag{3.14}$$

Where $v_c^e$ is the maximum velocity of the ego vehicle at the collision point, $v_c^t$ and $x_c^t$ represent the target vehicle's velocity and position just ahead of the collision point, $v_0^t$ and $x_0^t$ are the target vehicle's initial velocity and position, and $T_{claim}$ is the prediction horizon of the claim area.

Then, the critical velocity at the sampled points before the collision point along the path can be computed in reverse using kinematic equations.

And this is an example of a safety-set in the lane change scenario:

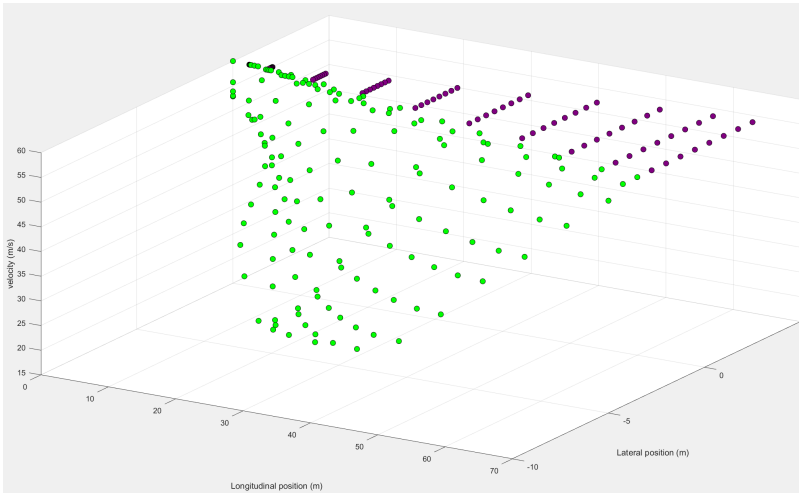



**Figure 3.19:** Safety set in a lane change scenario: green points indicate paths overlapping with the claim area, purple points represent non-overlapping paths.

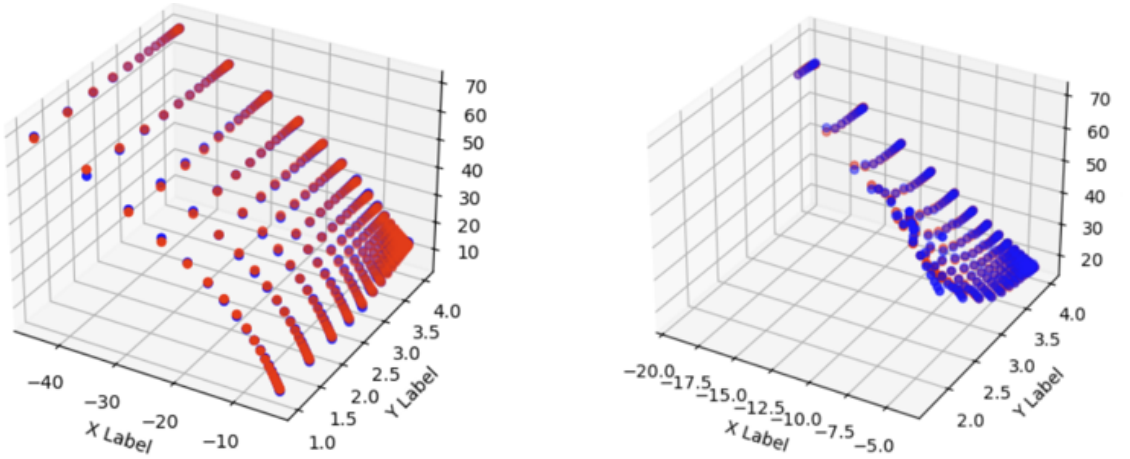
## 3.4 Safety-set boundary fitting

In the previous step, the critical velocity required to ensure collision avoidance was computed at each sampled point, forming the boundary of the safety set. Subsequently, polynomial regression is employed to fit these boundary points, representing the safety set as continuous functions. It is important to note that, the boundary of the safety set derived from analytical calculations is often represented by multiple functions. To simplify this representation and facilitate its use as constraints in the subsequent safety controller, the critical velocity is recalculated at each sampled point, followed by polynomial fitting.

The higher the degree of the polynomial, the better it can fit the boundary points. However, overly complex functions may lead to overfitting and can also complicate subsequent MPC calculations. After trade-off, we use a 5th-degree polynomial regression to fit the safety-set. The expression of the safety-set would be:

$$v_{\text{critical}} = f_{\text{safety-set}}(x_{\text{sample}}, y_{\text{sample}}) = \sum_{i=1,j=1}^{5} w_{ij} * x^{i}_{\text{sample}} y^{j}_{\text{sample}} \tag{3.15}$$

Below are examples of different shaped safety-sets being fitted using this approach:




**(a)** For one of the lower safety-set in the hidden pedestrian scenario.

**(b)** For one of the upper safety-set in the hidden pedestrian scenario.

**Figure 3.20:** The original points (in blue) and approximation result(in red). *XLabel* is longitudinal position and *YLabel* is lateral position.

The evaluation metrics for the examples are shown in the following table:

| Model Name | Mean Squared Error | $R^2$ Score |
|---|---|---|
| upper model | 0.8646 | 0.9949 |
| lower model | 0.0923 | 0.9991 |

**Table 3.2:** evaluation metrics for the regression models

## 3.5 Precautionary controller

One way to use the safety-set is by applying it as a velocity constraint in a MPC controller, ensuring that the vehicle's state always remains within the safe region. This serves as a precautionary controller, continuously keeping the vehicle's state within a safe range during its motion.

A relatively accurate kinematic bicycle model and kinematic analysis were used to derive the safety set. Based on this safety set, an MPC controller is being constructed, and its performance will be validated through simulations. In the simulations, the point mass model mentioned in Chapter 2 is used to represent the dynamics of the ego vehicle, and the MPC is developed based on this linear model.

Here is the formulation of the precautionary MPC controller:

$$\min_{\mathbf{u}^{\mathbf{e}}_{\text{ctrl}}(k),\forall k\in[0,N]} \sum_{k=0}^{N} \|(\mathbf{x}^{\mathbf{e}}_{\text{ctrl}}(k) - \mathbf{x}_{ref}(k))\|^2_{\mathbf{Q}} + \sum_{k=0}^{N} \|\mathbf{u}^{\mathbf{e}}_{\text{ctrl}}(k)\|^2_{R} + \|(\mathbf{x}^{\mathbf{e}}_{\text{ctrl}}(k+1) - \mathbf{x}_{ref}(k+1))\|^2_{\mathbf{Q_f}} + s * Q_s \tag{3.16a}$$

$$\text{s.t.} \quad \mathbf{x}^{\mathbf{e}}_{\text{ctrl}}(k+1) = \mathbf{A}_p \mathbf{x}^{\mathbf{e}}_{\text{ctrl}}(k) + \mathbf{B}_p \mathbf{u}^{\mathbf{e}}_{\text{ctrl}}(k), \quad \forall k \in [0, N] \tag{3.16b}$$

$$\mathbf{x}^{\mathbf{e}}_{\text{ctrl}}(k) \in \mathbb{X}^{e}_{\text{ctrl}}, \quad \mathbf{u}^{\mathbf{e}}_{\text{ctrl}}(k) \in \mathbb{U}^{e}_{\text{ctrl}}, \quad \forall k \in [0, N] \tag{3.16c}$$

$$g_s(\mathbf{x}^{\mathbf{e}}_{\text{ctrl}}(k \mid t), \mathbf{u}^{\mathbf{e}}_{\text{ctrl}}(k \mid t)) \leq 0, \quad \forall k \in [0, N] \tag{3.16d}$$

where

- The matrices $\mathbf{Q}$, $\mathbf{Q_f}$, $\mathbf{R}$, and $\mathbf{Q_s}$ represent the weighting matrices of process cost, terminal cost, control signal, and slack variable, respectively.
- The $s$ is the slack variable.
- The $\mathbf{A}_p$ and $\mathbf{B}_p$ represent the state space matrices of point mass model.

For Equation 3.16a, this is the typical cost function in MPC. In this case, our references are the target velocity and target lateral position. The target velocity is generally set to the initial velocity of the vehicle in the simulation. For the hidden pedestrian scenario, the target position is the centerline of the current lane, while for the lane change scenario, the target position is the centerline of the target lane. We also introduce slack variable to relax the constraint, making it easier for the controller to find a feasible solution.

For Equation 3.16b, The constraints here represent the vehicle's dynamic model. In this case, we use a linearized point-mass kinematic model, as described in Chapter 2.

For Equation 3.16c, the constraints represent the vehicle's physical limitations, as detailed in Table 3.1. Additionally, these constraints ensure that the vehicle remains within the drivable area.

For Equation 3.16d, here the Inequality constraints represents the velocity constraints from safety-set. First, it is necessary to determine which part of the safety-set the vehicle's state is in when the controller is activated, the constraint could be expressed as:

$$\begin{cases} \text{In the lower safety-set} : v^e_c(k) & \leq f_{\text{lower safety-set}}(x^e_c(k), y^e_c(k)) + s \\ \text{In the upper safety-set} : v^e_c(k) & \geq f_{\text{upper safety-set}}(x^e_c(k), y^e_c(k)) - s \end{cases} \tag{3.17}$$

Here $f_{\text{lower safety-set}}$ and $f_{\text{upper safety-set}}$ are the fitted functions representing the lower and upper bounds of the safety-set, respectively, and $S$ is the slack variable.

For example, the algorithm specified in the hidden lane scenario could be expressed as an algorithm in Appendix 1.

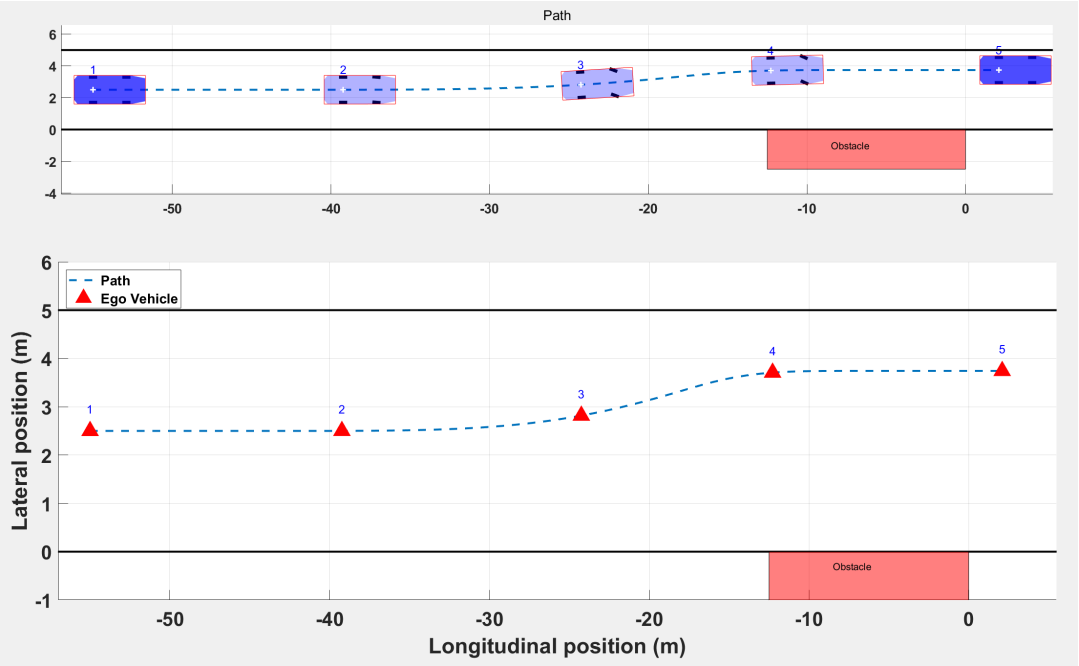
# 4

# Results and analysis

## 4.1 Case study: Hidden pedestrian scenario

### 4.1.1 Execution case in the hidden pedestrian scenario

In this scenario, the ego vehicle starts at the lane center with an initial position of 2.5 meters and 58 meters away from the blind spot, traveling at 10 m/s. The reference speed and lateral position are set to 10 m/s and the lane center, respectively. For the pedestrian claim area model, the maximum speed is 2.7 m/s vertical to the road and 1.5 m/s parallel to the road. During simulation, no actual pedestrian appears, and after responding to the potential threat, the vehicle continues to drive away.




| The timelapse is shown with five time instances | | | | |
|---|---|---|---|---|
| Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 |
| 0.0s | 1.6s | 3.3s | 4.9s | 6.5s |

**Figure 4.1:** This figure shows the trajectory of ego vehicle on the lane over time. It can be observed that the vehicle turns upward during its motion, laterally steering away from the pedestrian's blind spot.

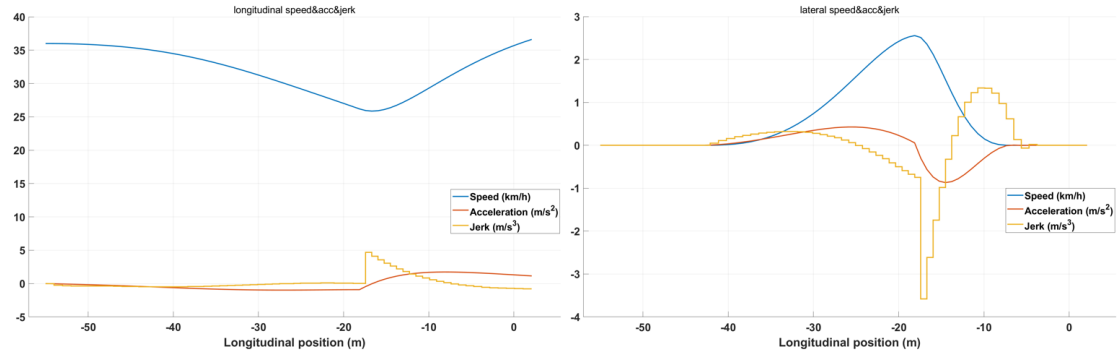



**Figure 4.2:** The left figure shows the longitudinal velocity, acceleration, and jerk, while the right figure displays the lateral ones. It can be seen that the vehicle first decelerates gently in the longitudinal direction and then accelerates after the blind spot disappears, moving away from the blind spot in the lateral direction.

For the second time sample of the above motion process. The images Figure 4.3 and Figure 4.4 below illustrate how the Precautionary MPC controller responds to potential threats and make decisions for one time step.

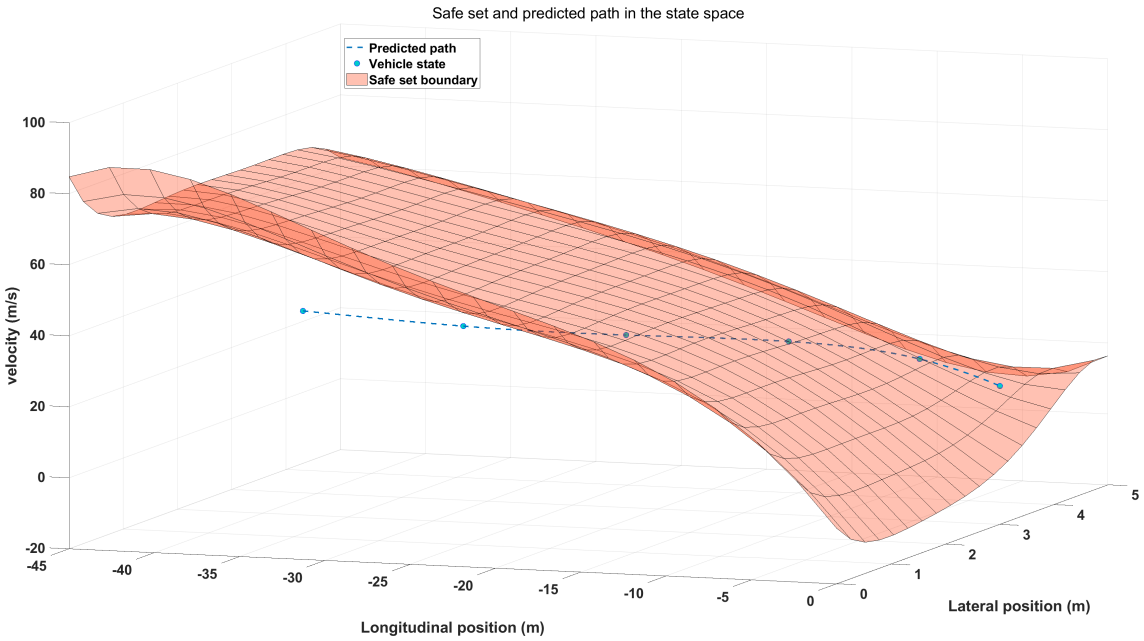



**Figure 4.3:** This figure simultaneously shows the optimal trajectory obtained by the MPC in the state space, along with the boundary surface of the safety-set at that moment.

In this simulation, since the vehicle's initial state is within the lower set, the lower set acts as a constraint for the MPC. We can observe that in the state space, the optimal state trajectory of the ego vehicle consistently remains below the boundary surface, ensuring that the ego vehicle stays within the safety-set at all times.

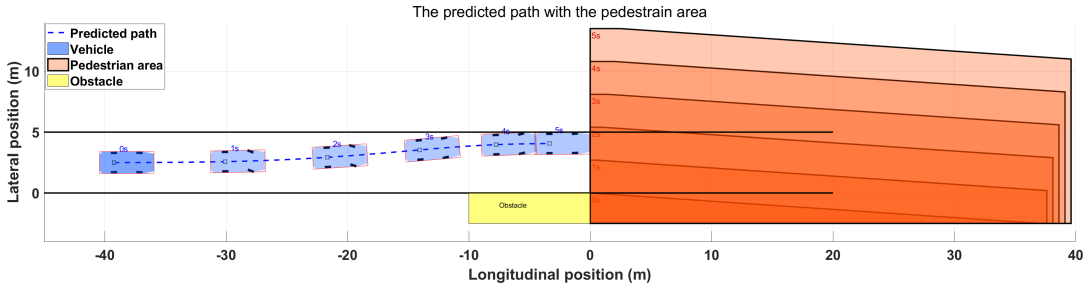



**Figure 4.4:** This figure shows the trajectory of the ego vehicle on the lane, along with the claim areas of the pedestrian at several corresponding time points.

An essential goal of this controller is to ensure that the vehicle does not enter the pedestrian's claim area. As shown in this figure, although only velocity constraints are used without directly preventing overlap between the vehicle and the claim area, the optimal trajectory obtained by the MPC still ensures that the vehicle does not enter the area potentially occupied by the pedestrian.

In other words, the MPC controller can achieve obstacle avoidance solely through velocity constraints imposed by the safety-set, which significantly simplifies the typical obstacle avoidance constraints. The online MPC does not require as long of a horizon, as its target set can be initialized with the offline-computed set corresponding to the position at the end of the horizon.

### 4.1.2 Collision avoidance robustness experiments

The first experiment simulated a wide range of cases of which settings is detailed in this section, implementing both the standard AEB system and the AEB system assisted by the precautionary safety-set in each scenario. The aim was to comprehensively compare their collision avoidance success rates. In Experiment I, only the initial longitudinal position of the vehicle is fixed at $-45$m and the rest of the parameters are randomly and uniformly generated over a specific range. The meaning of appearing distance is when the front of the vehicle is how many metres away from the front of the obstacle longitudinally, pedestrians are generated and detected by ego vehicle. The coordinate, initial positions of ego vehicle and the appearing distances are illustrated in Figure 4.5:

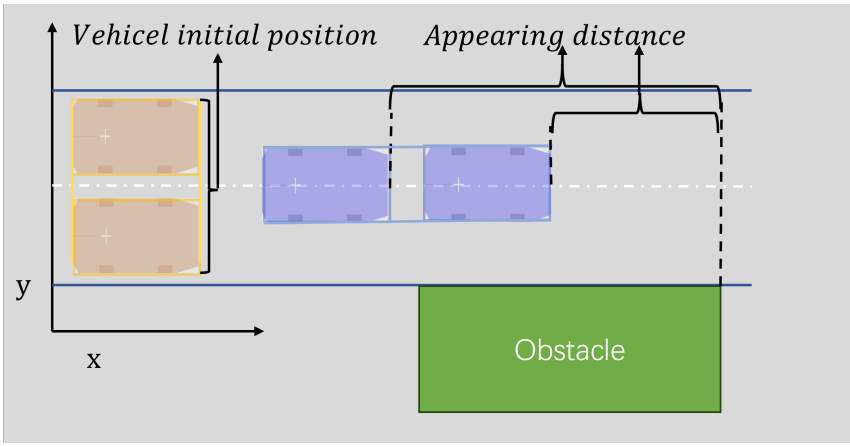



**Figure 4.5:** The coordinate (represented by $x$ and $y$), initial positions of ego vehicle (in orange) and the appearing distances (in blue) in Experiment I.

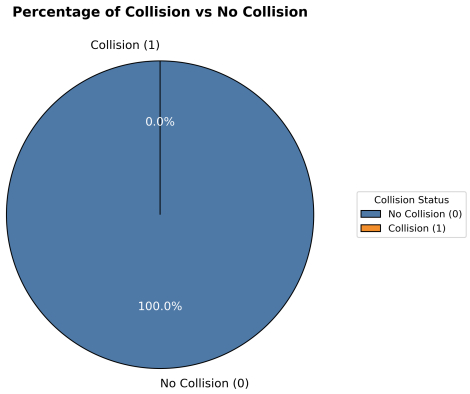
The specific ranges are given in the following table:

| Parameter | Lower bound | Upper bound |
|---|---|---|
| Ego vehicle lateral position | 1 m | 4 m |
| Ego vehicle velocity | 10 m/s | 19 m/s |
| Appearing distance | 12 m | 20 m |
| Pedestrian velocity | 0.8 m/s | 2.7 m/s |

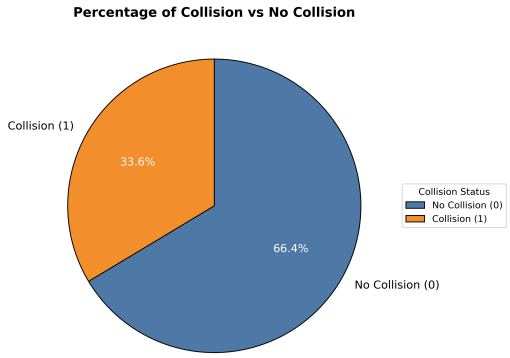
**Table 4.1:** Parameters range in Experiment I

The longitudinal position of the vehicle was fixed at -45 meters in the coordinate system, while the transverse position varied between 1 and 4 meters, ensuring that the vehicle could enter the same space in the experiment as when determining the precautionary safety-set. The initial range of pedestrian positions was defined in the pedestrian modeling section of the methodology chapter. Since this experiment focuses on comparing the performance of different systems in critical situations, the lower bounds for both vehicle and pedestrian speeds were set relatively high. Additionally, the upper bound for vehicle speed was set slightly below the maximum limit derived from the polynomial regression, ensuring that the system could fully utilize the precautionary safety-set.
After 1000 rounds of experiments, using the same parameters for each round, the success rate of collision avoidance is shown in Figure 4.6:
It is clearly noticeable that with the support of the precautionary safety-set, the ego vehicle is always able to avoid collisions with pedestrians. However, relying on the AEB system alone, in the 33.6 percent of the cases collision happened. Here are two specified cases worth studying:

In **case 1**, the initial position of ego vehicle is $[-45m, 2.5m]$, the pedestrian's appearing distance is $20m$ , the velocities of ego vehicle and pedestrian are $14.5m/s$




**(a)** Success rate to avoid collision with safety-set.




**(b)** Success rate to avoid collision without safety-set.

**Figure 4.6:** Success rate of collision avoidance with different systems. In the legend of each sub-figure, the number 0 represents no collision, while the number 1 represents a collision.

and $1.5m/s$, respectively. The trajectories of the ego vehicle and the pedestrians with different systems are shown in Figure 4.7 and Figure 4.8. For improved readability, the sampling times in Figure 4.7 are provided in Table 4.2, and the sampling times in Figure 4.8 are provided in Table 4.3.

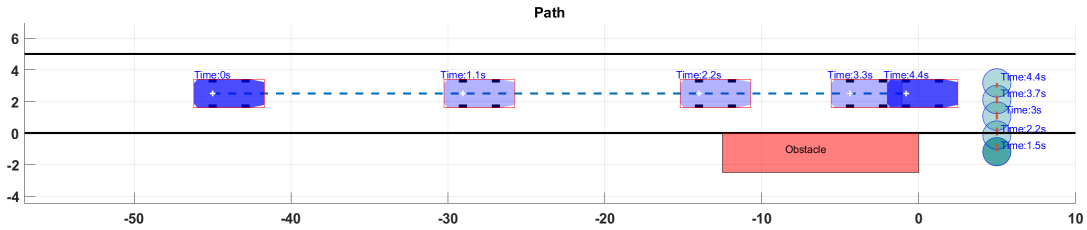



**Figure 4.7:** Case1: The trajectories of the ego vehicle and the pedestrians (in circle) with **only** AEB system.

| Participants | 1st ST | 2nd ST | 3rd ST | 4th ST | 5th ST |
|---|---|---|---|---|---|
| Ego vehicle (from left to right) | 0 s | 1.1 s | 2.2 s | 3.3 s | 4.4 s |
| Pedestrian (from down to up) | 1.5 s | 2.2 s | 3 s | 3.7 s | 4.4 s |

**Table 4.2:** Case1: Sampling time in Figure 4.7 with **only** AEB system. "ST" is the abbreviation of "sampling time".

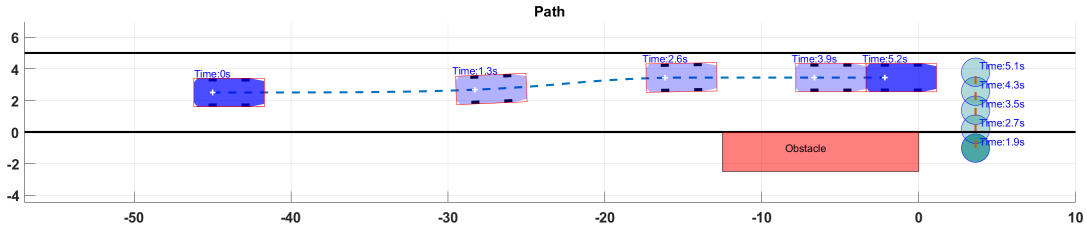



**Figure 4.8:** Case1: The trajectories of the ego vehicle and the pedestrians (in circle) with **both** AEB system and precautionary safety-set.

| Participants | 1st ST | 2nd ST | 3rd ST | 4th ST | 5th ST |
|---|---|---|---|---|---|
| Ego vehicle (from left to right) | 0 s | 1.3 s | 2.6 s | 3.9 s | 5.2 s |
| Pedestrian (from down to up) | 1.9 s | 2.7 s | 3.5 s | 4.3 s | 5.1 s |

**Table 4.3:** Case1: Sampling time in Figure 4.8 with **both** AEB system and safety-set.

Besides, the longitudinal velocity, acceleration and jerk of the ego vehicle vary with different systems are plotted in Figure 4.9 and figure 4.10:

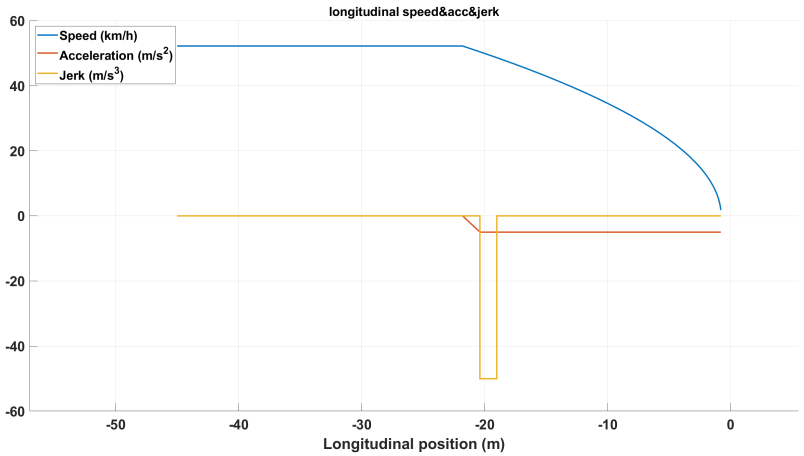



**Figure 4.9:** Case1: The curves representing the longitudinal velocity, acceleration and jerk of ego vehicle with **only** AEB system.

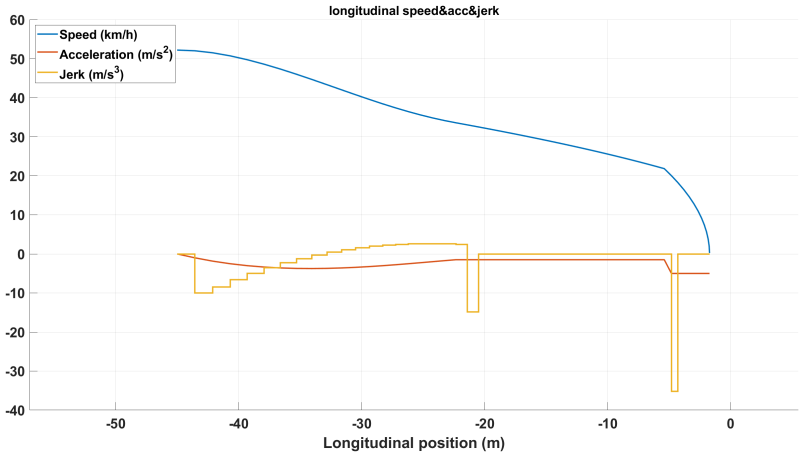



**Figure 4.10:** Case1: The curves representing the longitudinal velocity, acceleration and jerk of ego vehicle with **both** AEB system and precautiionary safety-set.

It can be seen that both systems achieve collision avoidance. The difference is that with the precautionary safety-set, the velocity of the ego vehicle is slowly reduced while the lateral position is increased, thus achieving risk avoidance to some extent, hence the longitudinal position of the ego vehicle is closer to the end of the obstacle when the AEB is activated, i.e. the AEB is activated later and the braking distance of the ego vehicle is shorter.

In **case 2**, the initial position of ego vehicle is $[-45m, 1.5m]$, somewhat lower than in case 1. The pedestrian's appearing distance is which is $17m$, more closer to the end of the obstacle. Moreover, the ego vehicle starts running with a higher velocity, $16m/s$ and the initial velocity of pedestrian is $1.2m/s$. The trajectories of the ego vehicle and the pedestrians with different systems are shown in Figure 4.11 and Figure 4.12. For clearer readability, the sampling times in Figure 4.11 are provided in table 4.4, and the sampling times in Figure 4.12 are provided in Table 4.5.

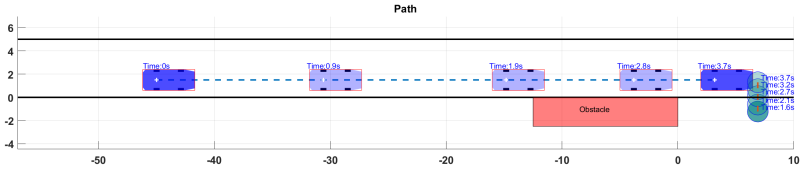



**Figure 4.11:** Case2: The trajectories of the ego vehicle and the pedestrians (in circle) with **only** AEB system.

| Participants | 1st ST | 2nd ST | 3rd ST | 4th ST | 5th ST |
|---|---|---|---|---|---|
| Ego vehicle (from left to right) | 0 s | 0.9 s | 1.9 s | 2.8 s | 3.7 s |
| Pedestrian (from down to up) | 1.6 s | 2.1 s | 2.7 s | 3.2 s | 3.7 s |

**Table 4.4:** Case2: Sampling time in Figure 4.11 with **only** AEB system.

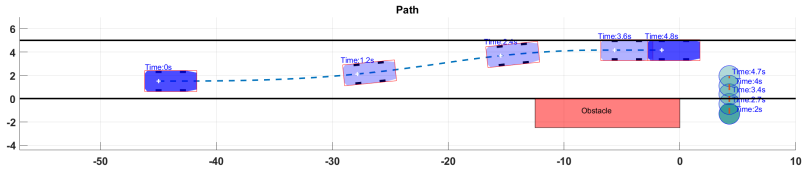



**Figure 4.12:** Case2: The trajectory of the ego vehicle with **both** AEB system and precautionary safety-set.

| Participants | 1st ST | 2nd ST | 3rd ST | 4th ST | 5th ST |
|---|---|---|---|---|---|
| Ego vehicle (from left to right) | 0 s | 1.2 s | 2.4 s | 3.6 s | 4.8 s |
| Pedestrian (from down to up) | 2 s | 2.7 s | 3.4 s | 4 s | 4.7 s |

**Table 4.5:** Case2: Sampling time in Figure 4.12 with **both** AEB and set.

Also, the longitudinal velocity, acceleration and jerk of the ego vehicle vary with different systems are plotted in Figure 4.13 and Figure 4.14:

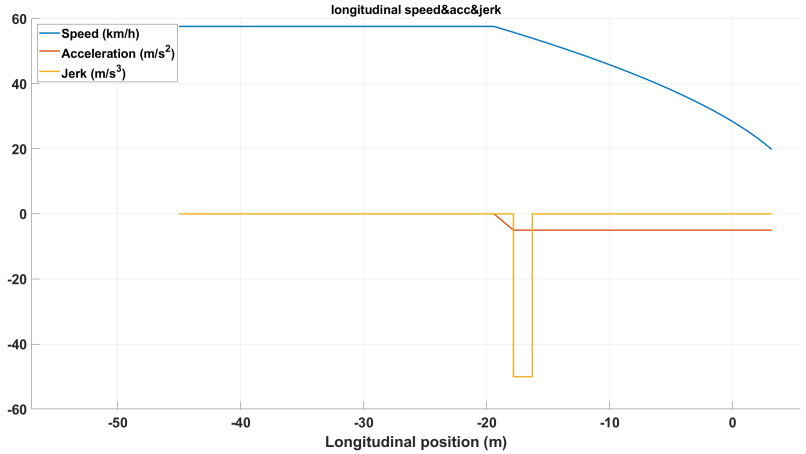



**Figure 4.13:** Case2: The curves representing the longitudinal velocity, acceleration and jerk of ego vehicle with **only** AEB system.

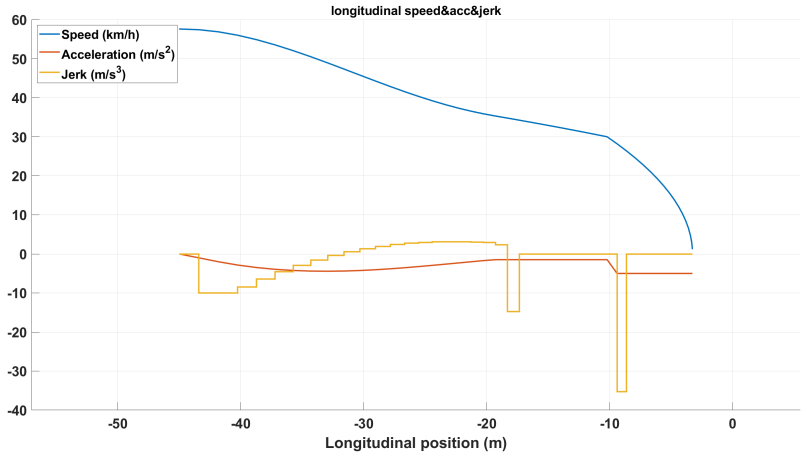



**Figure 4.14:** Case2: The curves representing the longitudinal velocity, acceleration and jerk of ego vehicle with **both** AEB system and precautiionary safety-set.

In scenarios where only the AEB system is active, the ego vehicle attempts to perform an emergency maneuver; however, a collision still occurs due to the insufficient longitudinal distance between the pedestrian and the ego vehicle at the moment the pedestrian appears, combined with the high speed of the ego vehicle.

### 4.1.3 Driving efficiency experiment

Driving efficiency in this context refers to the efficiency with which a vehicle drives through a road segment when faced with potential hazards. We want to ensure that the driving task is completed with the highest possible efficiency while maintaining safety. In fact, to avoid potential threats, good results can also be achieved by using an MPC that directly considers the claim area. In a typical MPC, obstacle avoidance can be ensured simply by preventing the ego vehicle from entering the claim area within the prediction horizon.

Therefore, in the driving efficiency experiment, the precautionary controller based on the safety set is compared with an MPC controller that directly considers the avoidance of the pedestrian claim area based on the hyperplane separation theorem. Since the purpose of the second experiment is to demonstrate the efficiency of the precautionary controller, the experiment was designed without pedestrians suddenly rushing out of the hidden area.
A standard MPC controller for obstacle avoidance was implemented. The formulation and parameters of this MPC are similar to the previously mentioned precautionary safety controller, as defined in Equation 3.16, except that the velocity constraint based on the reachable set in the inequality constraints is replaced by the hyperplane separation theorem from Equation 3.9. This ensures that the ego vehicle avoids collisions with pedestrians by steering and decelerating. In this context, we also refer to this controller as the direct controller.
Experiments were conducted in the hidden pedestrian scenario. The vehicle's initial lateral position was 58 meters away from the blind spot. For the pedestrian claim area model, the maximum speed was set to 2.7 m/s perpendicular to the road and

1.5 m/s parallel to the road. The lateral position of the ego vehicle was uniformly sampled across all possible lateral positions on the road, and the initial speed was uniformly sampled between 9 m/s and 19 m/s. Both controllers were run for each configuration, and the pass times, defined as the time taken for the vehicle to fully pass through the road segment, were compared.

In 25 experiments, regardless of variations in the initial lateral position and longitudinal speed of the ego vehicle, the passing time of the precautionary controller was consistently shorter under the same conditions. The table below presents a comparison of the two data sets under specific settings:

**Table 4.6:** Comparison of precautionary controller (A) and direct controller (B)

| Initial Speed | Lateral Position | Pass Time (A) | Pass Time (B) |
|---|---|---|---|
| 9 | 1.00 | **5.1** | 7.5 |
| 9 | 1.75 | **5.0** | 7.5 |
| 9 | 2.50 | **5.0** | 7.6 |
| 9 | 3.25 | **4.9** | 6.8 |
| 9 | 4.00 | **4.9** | 6.2 |
| 11.5 | 1.00 | **4.7** | 7.0 |
| 11.5 | 1.75 | **4.7** | 7.0 |
| 11.5 | 2.50 | **4.6** | 7.0 |
| 11.5 | 3.25 | **4.6** | 6.3 |
| 11.5 | 4.00 | **4.6** | 5.7 |
| 14 | 1.00 | **4.4** | 6.4 |
| 14 | 1.75 | **4.4** | 6.4 |
| 14 | 2.50 | **4.3** | 6.3 |
| 14 | 3.25 | **4.3** | 5.7 |
| 14 | 4.00 | **4.2** | 5.2 |
| 16.5 | 1.00 | **4.1** | 5.6 |
| 16.5 | 1.75 | **4.1** | 5.5 |
| 16.5 | 2.50 | **4.1** | 5.4 |
| 16.5 | 3.25 | **4.0** | 4.9 |
| 16.5 | 4.00 | **3.9** | 4.6 |
| 19 | 1.00 | **3.8** | 4.5 |
| 19 | 1.75 | **3.8** | 4.3 |
| 19 | 2.50 | **3.7** | 4.2 |
| 19 | 3.25 | **3.7** | 4.0 |
| 19 | 4.00 | **3.6** | 3.8 |

Selecting one of the cases of which settings are 14-m/s initial velocity and 2.5-m lateral position, the trajectory of the ego vehicle is shown in Figure 4.15:

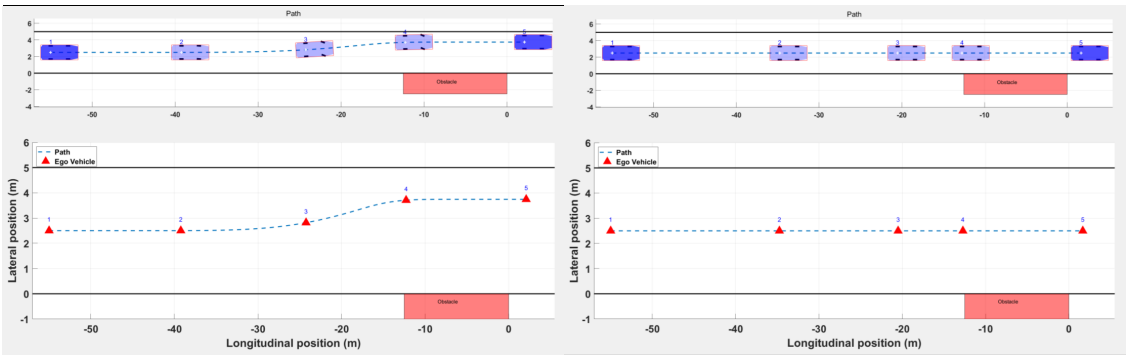



**Figure 4.15:** The trajectory of the ego vehicle with precautionary safety controller (left) and direct controller (right).

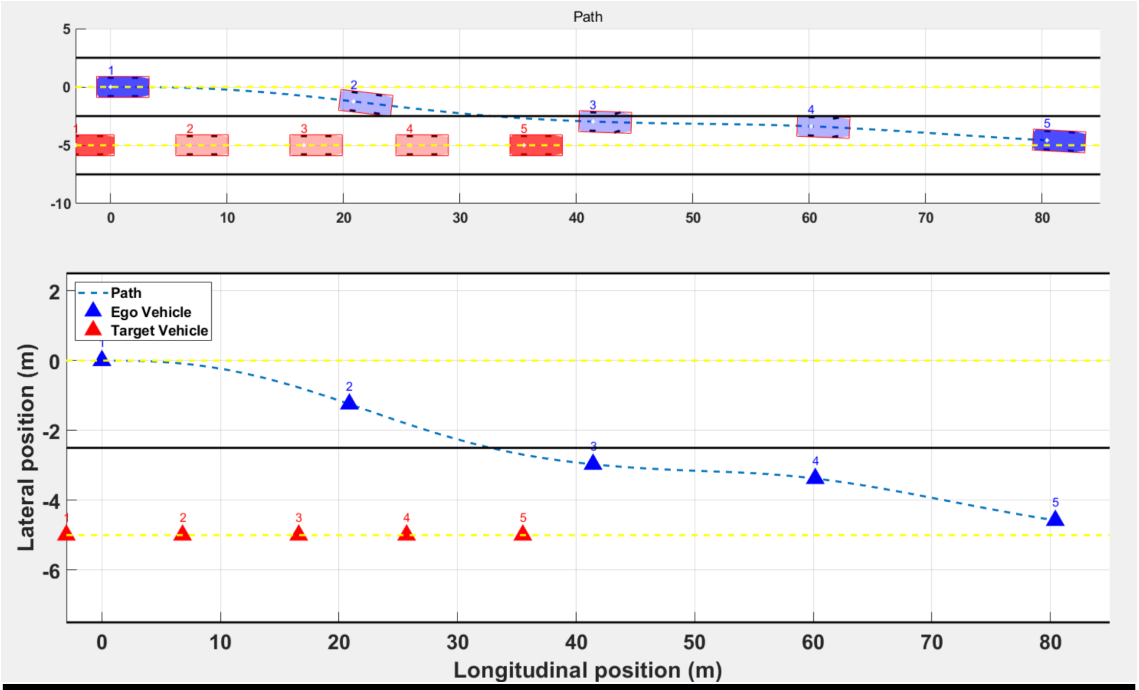
In Figure 4.15, the trajectory of the ego vehicle with direct controller is in a straight line, which is to be expected since the initial lateral position is right in the middle of the road, and the controller with constrains which is stopping outside the hidden area could be always functioning with such velocity settings. Trajectory of vehicle with precautionary controller is similar to these results in robustness experiment. In the final stage, the standard MPC predicts that the entire roadway is covered by the pedestrian claim area. However, the precautionary-based controller can predict that part of the space remains available for the ego vehicle to pass at a higher speed. In this sense, it extends the normal MPC's prediction horizon, enabling the ego vehicle to navigate the area more efficiently while still strictly avoiding potential threats.

## 4.2 Case study: Lane change scenario

### 4.2.1 Execution case in the lane change scenario

In this execution, the ego vehicle starts in the center of the upper lane, traveling at a speed of 15 m/s with the goal of maintaining this speed and changing lanes to the lower lane. At this point, there is a target vehicle 5 meters behind in the lateral position, moving forward at 6 m/s.

By combining the two figures, it can be seen that during the lane change maneuver, the ego vehicle's longitudinal speed remained almost unchanged, while it experienced two phases of lateral movement. Initially, driven by the reference to the lower lane center, the vehicle approached the target lane without entering the target vehicle's claim area laterally. Then, after calculating a feasible overtaking path, the vehicle further completed the lane change safely. Similar to the first case, without direct obstacle avoidance constraints, the vehicle successfully avoided the obstacle.This demonstrates the generality of the obstacle avoidance algorithm based on the safety-set, as well as the universality of the precautionary process.




| The timelapse is shown with five time instances | | | | |
|---|---|---|---|---|
| Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 |
| 0.0s | 1.6s | 3.0s | 4.3s | 5.7s |

**Figure 4.16:** This figure shows the trajectory of ego vehicle and target vehicle on the lane over time. It can be observed that the ego vehicle completed a lane change and overtaking of the target vehicle.

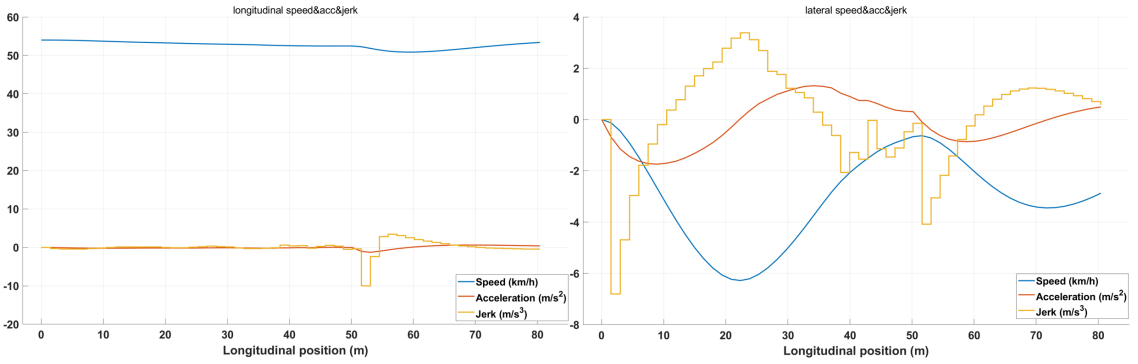



**Figure 4.17:** The left figure shows the longitudinal velocity, acceleration, and jerk, while the right figure displays the lateral ones.

For the second time sample of the above motion process. The images Figure 4.18 and Figure 4.19 below illustrate how the Precautionary MPC controller responds to potential threats and make decisions for one time step.

For each time step, here is how the paths are planned on the road and the collision detection based on claim area, as shown in Figure 4.18. And in the control process,

the optimal states stay under the safety-set boundary, in Figure 4.19.

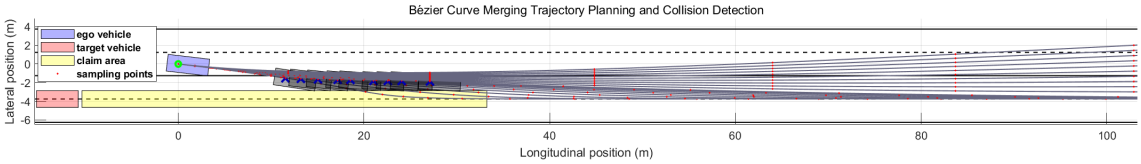



**Figure 4.18:** The figure shows the target vehicle with its claim area, the ego vehicle with its planned paths, and the collision points along those paths.

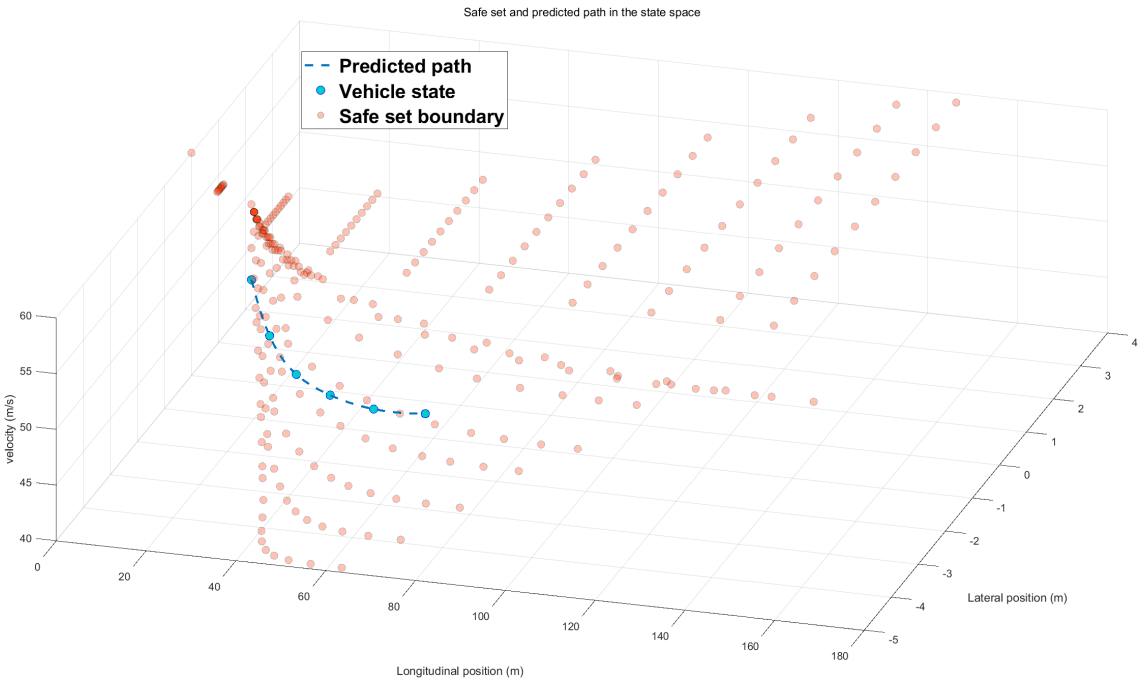



**Figure 4.19:** This figure simultaneously shows the optimal trajectory obtained by the MPC in the state space, along with the boundary points of the safety-set at that moment. Still, the trajectory remains under the boundary.

# 5

# Conclusion and discussion

## 5.1 Conclusion

The aim of this thesis is to improve the functionality of Collision Avoidance Systems (CAS) and enhance the safety of all road users. To achieve this, we propose a methodology for autonomous vehicles to implement precautionary collision avoidance systems. At its core, the methodology formalizes potential uncertain hazards and translates safety requirements into vehicle dynamic constraints, defined as the safety-set. By integrating this concept with Model Predictive Control (MPC), the system anticipates and mitigates collision risks, ensuring a proactive approach to maintaining safety in dynamic and unpredictable environments.

The proposed methodology was tested and evaluated in two critical scenarios: the hidden pedestrian scenario and the lane change scenario.
In the hidden pedestrian scenario simulation, with safety-set constraints in place, the system reliably avoided collisions and maintained smooth vehicle operation even under occluded visibility. Experimental results demonstrated that the precautionary safety function achieved an effective balance between safety and efficiency. In the collision avoidance robustness experiment, we set different pedestrian appearance timings and positions, as well as various initial states of the vehicle, conducting 1,000 rounds of testing. Here the AEB enhanced by the precautionary safety function is compare with standalone AEB. The former achieved a 100% success rate in avoiding hidden pedestrians, outperforming the latter by 33.6%. For the driving efficiency experiment, we designed an obstacle avoidance MPC for areas potentially occupied by pedestrians and compared it with our precautionary safety function. Under 25 different initial conditions, the precautionary safety function consistently resulted in shorter passage times compared to standalone MPC, demonstrating superior efficiency.
Furthermore, simulations in the lane change scenario demonstrated that, with safety-set constraints, the vehicle successfully completed lane changes and responded to sudden and unpredictable vehicle behaviors, underscoring the generality and adaptability of the proposed approach.

These findings confirm the system's significant potential to enhance the safety and operational efficiency of autonomous vehicles in complex environments.

From the perspective of MPC, this research introduces a novel method for defining

constraints in obstacle avoidance. Instead of avoiding objects spatially, safety-sets are computed to ensure avoidance using only velocity constraints. Experimental results further showed that the safety-set based MPC, considering the entire drivable area based on velocity constraints, identified "smarter" paths, maintaining higher speeds and reducing blind zones. This improved foresight and breadth of vision are attributed to the predictive design of the safety-set for the precautionary safety function.

Despite these advancements, the study has certain limitations. The system employs simplified kinematic models and assumes ideal sensor conditions, which may not fully reflect the complexity of real-world scenarios. Additionally, experimental validation was limited to simulation environments, making physical testing an essential next step.

By formalizing safety requirements and integrating them into vehicle control strategies, this work lays a foundation for more foresighted active safety functions in autonomous vehicles. Its alignment with ethical and sustainability principles not only enhances technical capabilities but also fosters safer and more responsible autonomous driving in real-world applications.

## 5.2 Future work

Future work includes the following points:

1. Explore the generalizability of the framework presented in this thesis under more complex scenarios, such as vehicles operating in unstudied environments or even in situations without formal road infrastructure. For scenarios involving multiple target vehicles or pedestrians, both the formulation of path planning and the definition of the claim set should be carefully addressed.
2. All optimization tasks and controllers in the project currently consider only a kinematic model. To achieve results that are more aligned with real-world conditions, more accurate dynamic models should be employed. Additionally, for some predicted trajectories in certain scenarios, a feasibility analysis using a dynamic model should be conducted.
3. Our precautionary safety function may be overly conservative in many scenarios. To ensure the comfort of the driver and minimize unnecessary interference, many parameters in the function need to be better tuned based on actual conditions.

# Bibliography


[1] Heather E Rosen, Imran Bari, Nino Paichadze, Margaret Peden, Meleckidzedeck Khayesi, Jesús Monclús, and Adnan A Hyder. Global road safety 2010–18: an analysis of global status reports. *Injury*, 2022.

[2] Mattias Brännström, Erik Coelingh, and Jonas Sjöberg. Model-based threat assessment for avoiding arbitrary vehicle collisions. *IEEE Transactions on Intelligent Transportation Systems*, 11(3):658–669, 2010.

[3] David Nistér, Hon-Leung Lee, Julia Ng, and Yizhou Wang. The safety force field. *NVIDIA White Paper*, 15, 2019.

[4] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. On a formal model of safe and scalable self-driving cars, 2018.

[5] Markus Koschi and Matthias Althoff. Set-based prediction of traffic participants considering occlusions and traffic rules. *IEEE Transactions on Intelligent Vehicles*, 6(2):249–265, 2021.

[6] Valeriu Soltan. Support and separation properties of convex sets in finite dimension. *Extracta mathematicae*, 36(2):241–278, 2021.

[7] Jiayu Fan, Nikolce Murgovski, and Jun Liang. Efficient collision avoidance for autonomous vehicles in polygonal domains. *arXiv preprint arXiv:2308.09103*, 2023.

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

**DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY**
**CHALMERS UNIVERSITY OF TECHNOLOGY**
Gothenburg, Sweden
www.chalmers.se


CHALMERS
UNIVERSITY OF TECHNOLOGY

# A

# Appendix 1

**Algorithm 1** Autonomous vehicle control system with precautionary safety-set in case 1

$\mathbf{Sp} = [Sp_1, \ldots, Sp_n] = [[x_1, y_1] \ldots, [x_n, y_n]] \leftarrow$ logspace()
**for** $Sp_i \leftarrow Sp_1$ **to** $Sp_n$ **do**
  Find minimum safe initial velocity $Vmin_i$: $\min\limits_{\mathbf{u}(k), \forall k \in [0,N]} V_0$
  $\mathbf{V}min[i] \leftarrow Vmin_i$
  Find maximum safe initial velocity $Vmax_i$: $\min\limits_{\mathbf{u}(k), \forall k \in [1,N]} - V_0 + \sum V_k - V_0$
  $\mathbf{V}max[i] \leftarrow Vmax_i$
**end for**
$\mathcal{V}_{upper} =$ PolynomialRegression$(\mathbf{Sp}, \mathbf{V}min)$
constraints$_1 \leftarrow \mathcal{V}_{upper}$
$\mathcal{V}_{lower} =$ PolynomialRegression$(\mathbf{Sp}, \mathbf{V}max)$
constraints$_2 \leftarrow \mathcal{V}_{lower}$
**Set initial state:** $x\_current =$ initial_state
**while** control **do**
  **for** $i = 1$ to $N$ **do**
    Predict future state: $x\_pred_1(i) = f(x\_current, u\_pred_1(i))$, $x\_pred_2(i) = f(x\_current, u\_pred_2(i))$
    Compute future control input: $u\_pred_1(i) =$ Optimization$(J,$ constraints$_1)$, $u\_pred_2(i) =$ Optimization$(J,$ constraints$_2)$
  **end for**
  Get control sequence: u_sequence$_1$ = Solve$(J_1, x_pred_1,$ constraints$_1)$, u_sequence$_2$ = Solve$(J_2, x_pred_2,$ constraints$_2)$
  **if** $J_1 \leq J_2$ **then**
    u_optimal = u_sequence$_1$
  **else**
    u_optimal = u_sequence$_2$
  **end if**
  **Apply** $u\_current =$ u_optimal$[0]$
  **Measure or estimate new state:** $x\_current =$ MeasureOrEstimateState$()$
  **Update** historical data for next iteration
**end while**
**End operation** PolynomialRegression$\mathbf{Sp}, \mathbf{V}$
**while** Training **do**
  Predicted velocity for the i-th training: $h_\theta(Sp_i) = \theta_0 + \theta_1 x_i + \theta_2 y_i + \theta_3 x_i y_i + \cdots + \theta_{m-1} x_i^m + \theta_m y_i^m$
  Cost function: $J(\boldsymbol{\theta}) = \frac{1}{2n} \sum_{i=1}^{n} \left(h_\theta(Sp_i) - V_i\right)^2$
**end while**
**Output:** $v = f(\theta, Sp)$