Combined braking and steering maneuver for collision avoidance

Master's thesis in Complex Adaptive Systems

VILGOT LÖÖF WETTERVIK, GUSTAV MATTSSON

DEPARTMENT OF ELECTRICAL ENGINEERING

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

Master's thesis 2025: EENX30

Combined braking and steering maneuver for collision avoidance

VILGOT LÖÖF WETERVIK, GUSTAV MATTSSON

CHALMERS
UNIVERSITY OF TECHNOLOGY

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

Combined braking and steering maneuver for collision avoidance
VILGOT LÖÖF WETTERVIK, GUSTAV MATTSSON

Supervisor: Markus Carlander, Aptiv
Examiner: Jonas Sjöberg, Systems and Control, Electrical Engineering

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

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2025

Combined braking and steering maneuver for collision avoidance
VILGOT LÖÖF WETTERVIK, GUSTAV MATTSSON,
Department of Electrical Engineering
Chalmers University of Technology

Abstract

In modern vehicles today, safety features such as Automatic Emergency Braking (AEB) and Automatic Emergency Steering (AES) are important for collision avoidance and reducing traffic fatalities. AEB and AES are constrained by their inability to handle complex scenarios. Therefore, further development is required to explore new ways to avoid collisions in complex scenarios.

This thesis proposes two algorithms, Rule-based and Spline-based. Both utilize combined braking and steering with multi-target threat assessment for collision avoidance. The Rule-based algorithm relies on multi-target threat assessment and free space calculations, to determine when to steer or brake. The Spline-based algorithm samples several quintic spline trajectories every time step. If the algorithm finds collision-free, dynamical feasible trajectories, the safest trajectory is selected with a cost function.

Both algorithms manage to generate collision-free trajectories and prevent collisions if the trajectory is dynamically feasible. The two algorithms in some test scenarios choose different combinations of braking and steering, to avoid collision.

Keywords: Multi-target threat assessment, collision avoidance, quintic splines, dynamical feasibility

Acknowledgements

We would like to express our gratitude to our supervisor Marcus Carlander at Aptiv. We want to thank you for all the help and guidance during this intriguing project. It has been enriching for us.

Vilgot Lööf Wettervik, Gustav Mattsson, Gothenburg, June 2025

List of Acronyms

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

AEB	Automatic Emergency Braking
AES	Automatic Emergency Steering
COG	Center Of Gravity
MPC	Model Predictive Control
SAT	Separating Axis Theorem
SBST	Spline-Based-Search-Three
TTC	Time To Collision
VCS	Vehicle Coordinate System

Nomenclature

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

Indices

k	Index for iteration step
t	Index for time step

Parameters

BTN	Brake threat number
STN	Steer threat number
TTC	Time to collision number
SAT	Collision number

Variables

s	Distance manual calculation $[m]$
v_0	Initial speed $[m/s]$
t	Time $[s]$
a	Acceleration $[m/s^2]$
v	Vehicle velocity $[m/s]$
R_{lat}	Lateral distance required for the host vehicle to traverse in order to safely overtake the vehicle ahead $[m]$
t_c^2	Time to collision$[s]$
L_f	Length from Center Of Gravity to the front $[m]$
L_r	Length from Center Of Gravity to the rear $[m]$

δ	Steering angle of host vehicle [$degree$]
L_w	Wheelbase [m]
c	Curvature [$1/m$]
α_r	Rear tire slip angle [$degree$]
α_f	Front tire slip angle [$degree$]
M	Mass of vehicle [kg]
C_r	Rear cornering stiffness [$N/degree$]
C_f	Front cornering stiffness [$N/degree$]
R	Trajectory radius for small slip angles [m]
L	Length of vehicle [m]
$a_{rec,lat}$	Host vehicle required lateral acceleration [m/s^2]
$a_{max,lat}$	Host vehicle maximal lateral acceleration [m/s^2]
$a_{rec,long}$	Host vehicle required longitudinal acceleration [m/s^2]
$a_{max,long}$	Host vehicle maximal achievable longitudinal deceleration [m/s^2]
h	Host vehicle heading [$degree$]
w	Host vehicle yaw rate [$degree/s$]
T	Time[s]
v_x	Host vehicle longitudinal velocity [m/s]
v_x	Host longitudinal velocity [m/s]
v_y	Host vehicle lateral velocity [m/s]
x	Host vehicle longitudinal position [m]
y	Host vehicle lateral position [m]
$p_{long,t}$	Target vehicle longitudinal position [m]
$p_{lat,t}$	Target vehicle lateral position [m]
$p_{0lat,t}$	Target vehicle lateral position [m]
θ_t	Target vehicle heading [$degree$]
θ_{0_t}	Target vehicle initial heading [$degree$]
$\dot{\theta}$	Target vehicle yaw rate [$degree/s$]
d_{long}	Longitudinal distance between host and target vehicle [m]
$v_{long,rel}$	Relative longitudinal velocity between host and target vehicle [m/s]
v_{rel}	Relative speed between host and target vehicle [m/s]
a_{rel}	Relative acceleration between host and target vehicle [m/s^2]
d_0	Distance separating host and target vehicle [m]
$a_{y,min}$	Host vehicle minimum lateral acceleration [m/s^2]

w_s	Host vehicle width $[m]$
P_x	Longitudinal distance between host and target vehicle $[m]$
P_y	Lateral distance between host and target vehicle $[m]$
γ_{min}	Host vehicle minimum yaw rate $[degree/s]$

Contents

List of Figures

List of Tables

List of Algorithms

1
Introduction

The introduction provides a brief overview of the report's topic. Furthermore, the introduction presents the background of today's vehicles emergency systems and different trajectory-finding algorithms.

1.1 Background

The number of vehicles has increased every year. According to the World Health Organization, more than 50 million people get injured and 2.5 % die yearly due to traffic accidents [1], more than 50 % of the fatalities include pedestrians, cyclists and motorcyclists [2]. In 2023, the Swedish police reported 229 traffic fatalities connected to road traffic accidents. The percentage of collisions resulting in death regarding rear-end collisions was 3.49 %, pedestrians 10.0 % and oncoming vehicles 21.8 % [3]. This highlights the importance of vehicle safety and investigating the potential of combining braking and steering from safety interventions, such as AEB and AES, together with a trajectory planner which could reduce those numbers.

In modern vehicles, Automatic Emergency Braking (AEB) has become a standard safety feature, driven by governmental and industrial priorities for collision prevention. AEB is designed to intervene as a last resort by automatically applying brakes when a collision is imminent [4]. However, at high speeds, AEB's effectiveness is limited because stopping distance scales quadratically with velocity, requiring more deceleration space as speed increases. In contrast, lateral displacement for steering evasion grows approximately linearly with speed. This means that while both braking and steering require longer maneuvers at higher speeds, braking demands disproportionately more distance, making AEB interventions increasingly insufficient compared to steering avoidance in critical high-speed scenarios [5]. To avoid false activations, AEB systems are often calibrated conservatively, but this conservative logic can delay braking until the stopping distance exceeds available space, leaving steering as the only viable option.

The counterpart functionality, Automatic Emergency Steering (AES), aims to avoid collisions by executing a steering maneuver closer to the potential collision time. This has proven to be successful even for distracted drivers [6]. However the AES is frequently constrained by the presence of other road users and physical road boundaries.

In previous studies, different trajectory-finding algorithms have been applied to guide autonomous vehicles. There are some diverse types of trajectory finding algorithms which are often split into two categories: global and local trajectory planning. Global trajectory planning aims to find the shortest trajectory between start and goal, by knowing the map and obstacles in advance. Local trajectory planning is real-time trajectory planning in the local environment, often used when avoiding dynamic obstacles [7]. Examples of local trajectory planning are Model Predictive Control (MPC), State Lattice and Spline-Based-Search-Three (SBST).

In previous research, MPC has been utilized for collision avoidance. MPC is a multivariable control [8], which applies a model to compute optimal control sequences-based future events in the finite prediction horizon. The MPC navigates the drivable area with dynamic obstacles in the scene and makes the vehicle drive along the drivable area. A drawback is that it is time-consuming to compute a trajectory because of the large number of parameters the MPC can be optimized over [9]. The parameters optimized are vehicle dynamics parameters such as jerk, longitudinal and lateral accelerations. Other parameters such as relative speed between targets and host vehicle, proximity to other objects and road boundaries are also optimized [10]. Sampling time is another parameter that can affect the computational time. Due to the need for real-time capabilities, a high sampling rate is necessary.

Similar to the MPC, the State Lattice [11] has been widely applied in the field of trajectory predictive algorithms. It is a graph-based trajectory planning algorithm that takes the vehicle dynamic constraints and kinematics into account. The sampled state space contains the position, orientation, and steering angle of the vehicle, which are connected through controllable edges. These edges can be computed offline and stored in lookup tables to make it possible for efficient real-time planning. In complex scenarios, it can handle several objects, but the main drawback is the precision of the State Lattice which is directly related to the resolution of the lattice.

In addition to MPC and State Lattice, in [12] studies about emergency collision avoidance systems have also been conducted. This type of system generally struggles with decision-making and maneuvering at high speed. In the study, different methods are presented to address this problem. One method utilizes maximum braking and steering maneuvers, and another method applies steering maneuvers combined with longitudinal motion planning to avoid collisions in complex scenarios. Scenarios in [12] were simulated 4000 times with different initial velocities and distances. The algorithm avoided collisions in 89 % of the cases. Failures mostly occurred because of the noticeably short distance to the car in front, and no feasible gap could be found at that time.

Aside from MPC and State Lattice, braking-steering maneuver, SBST [11] is a fast real-time trajectory planner. It utilizes cubic splines, constructed by polynomials, and the basis is the 3D derivative of the position and jerk, which is constant through the segment. The jerk can switch the sign and amplitude between the segments, which results in a minimized jerk, resulting in an optimal path. Aside from cubic

splines, quintic splines can be applied with the aim of smoothening the path further and eliminating irregularities in the jerk. However, smoothening comes at the expense of a slight increase in jerk. [13] proposes quintic polynomials instead, which are computationally fast and widely used for trajectory planning.

1.2 Contribution

The project proposes two algorithms, both, combining braking and steering for collision avoidance in traffic scenarios where AEB and AES indicate a crash is unavoidable. The first algorithm, referred to as the Ruled-based algorithm, is introduced in Section 2.8.2, utilizing trajectory predictions, Section 2.3, to predict object's trajectories. They are evaluated with three threat assessments to estimate when to steer or brake, Section 2.4.1-2.4.3. The other is a Spline-based algorithm, presented in Section 2.8.4, where quintic splines are assessed together with the same set of threat assessments previously applied, Section 2.4.1-2.4.3. Together with a cost function, Section 2.8.3.3 to select a feasible trajectory.

To analyze the two proposed algorithms, five different scenarios are generated. Scenario one represents a single target case and scenarios two to five represent multi-target cases.

1.2.1 Scope

In this thesis, five real-world scenarios are proposed. The scenarios are designed for the host vehicle to have a possible collision-free trajectory, explained in section 2.9.1. They are employed to test the two developed collision avoidance algorithms to evaluate their ability to avoid collision with other targets in the scene. The following scenarios are proposed:

1. Single-lane driving, expanding into multilane driving on the highway with one target in the scene. AES and AEB are not available.

2. Multi-lane driving on the highway, with two targets in the scene. AES and AEB are not available.

3. Multi-lane driving on the highway, with three targets in the scene. AES and AEB are not available.

4. Urban driving, with two pedestrians in the host lane and one target in the adjacent lane in the opposite direction in the scene. AES and AEB are not available.

5. Multi-lane driving on the highway, with two targets in the scene. AEB is not available.

In scenarios where braking occurs, the applied constant braking is $8m/s^2$.

1.2.1.1 Scenario one, one target vehicle, braking

In the first scenario, a target vehicle travels ahead of the host vehicle at a constant velocity. As it reaches the end of the guardrails, the target vehicle suddenly initiates braking. This maneuver results in a blocked path for the host vehicle. In this situation, AES is unavailable, and the braking alone is insufficient to prevent a collision. Additionally, AEB is also not available.

Fig. 1.1: Scenario 1, the target (blue) in front of the host (red) suddenly brakes, guardrail to the left and right, road splits up from a single lane into two lanes.

1.2.1.2 Scenario two, two target vehicles, one braking

Similar to the first scenario, a target vehicle is traveling ahead of the host vehicle at a constant velocity. Simultaneously, another vehicle is maintaining a constant velocity in the adjacent lane, thereby obstructing the potential for an AES maneuver. When the lead vehicle suddenly decelerates, the host vehicle's braking alone proves insufficient to prevent a collision. Furthermore, AEB is not available in this situation.

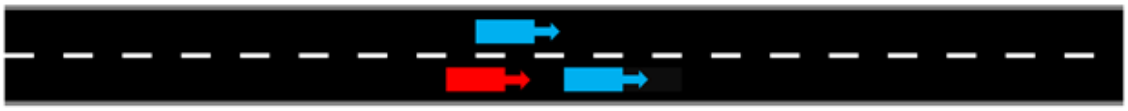


Fig. 1.2: Scenario 2, the target (blue) in front of the host (red) suddenly brakes, incoming target (blue) in the adjacent lane.

1.2.1.3 Scenario three, three target vehicles, one braking

Similar to the second scenario, a target vehicle is traveling ahead of the host vehicle at a constant velocity. In the adjacent lane, two additional target vehicles are also moving at constant velocities. At the moment of interest, one of these vehicles obstructs the possibility of executing an AES maneuver. The lead vehicle in front of the host vehicle suddenly decelerates. In this situation, braking alone is insufficient for the host vehicle to avoid a collision, and AEB is not available.

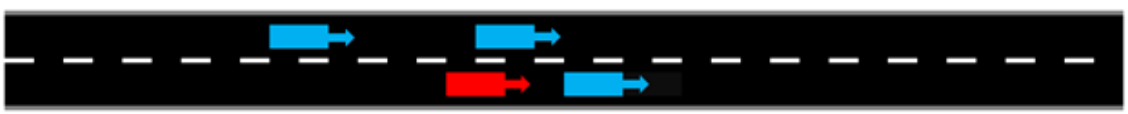


Fig. 1.3: Scenario 3, the target (blue) in front of the host (red) suddenly brakes, and two incoming targets (blue) in the adjacent lane at constant velocity.

1.2.1.4 Scenario four, one target vehicle and two pedestrians

In contrast to scenarios one through three, the target vehicle in this case is traveling in the opposite direction at a constant velocity in the adjacent lane. Additionally, two pedestrians are entering the roadway at constant walking speeds, one positioned ahead of the other. The combined presence of the oncoming vehicle and the pedestrians obstructs the host vehicle's ability to effectively employ either an AES maneuver or an AEB intervention.

Fig. 1.4: Scenario 4, two pedestrians (green) walk out onto the road in front of the host (red), target (blue) approaches the host in the adjacent lane in the opposite direction.

1.2.1.5 Scenario five, two target vehicles, two braking

Similar to scenario two, this case involves two target vehicles: one directly ahead of the host vehicle and another in the adjacent lane, both traveling at a constant velocity. In this scenario, the target vehicle in the adjacent lane is positioned further ahead than the one directly in front of the host. When the lead target vehicle suddenly brakes, relying solely on braking would lead to a collision, as AEB is not available. The target vehicle in the adjacent lane is also braking. However, evasive steering remains a viable option due to the available lateral and longitudinal margin relative to the target vehicle in the adjacent lane

Fig. 1.5: Scenario 5, the target (blue) in front of the host (red) suddenly brakes, target (blue) in the adjacent lane.

1.3 Thesis outline

Following the introduction, Chapter 2, Algorithm development and functional description, presents the methodology applied in this thesis and outlines the previous research. In addition, it presents two algorithms and how they are designed. Simulations and results of the algorithms, combined with Analyses, are presented in Chapter 3, Results and Analyses, followed by Chapter 4, Conclusions.

2

Algorithm development and functional description

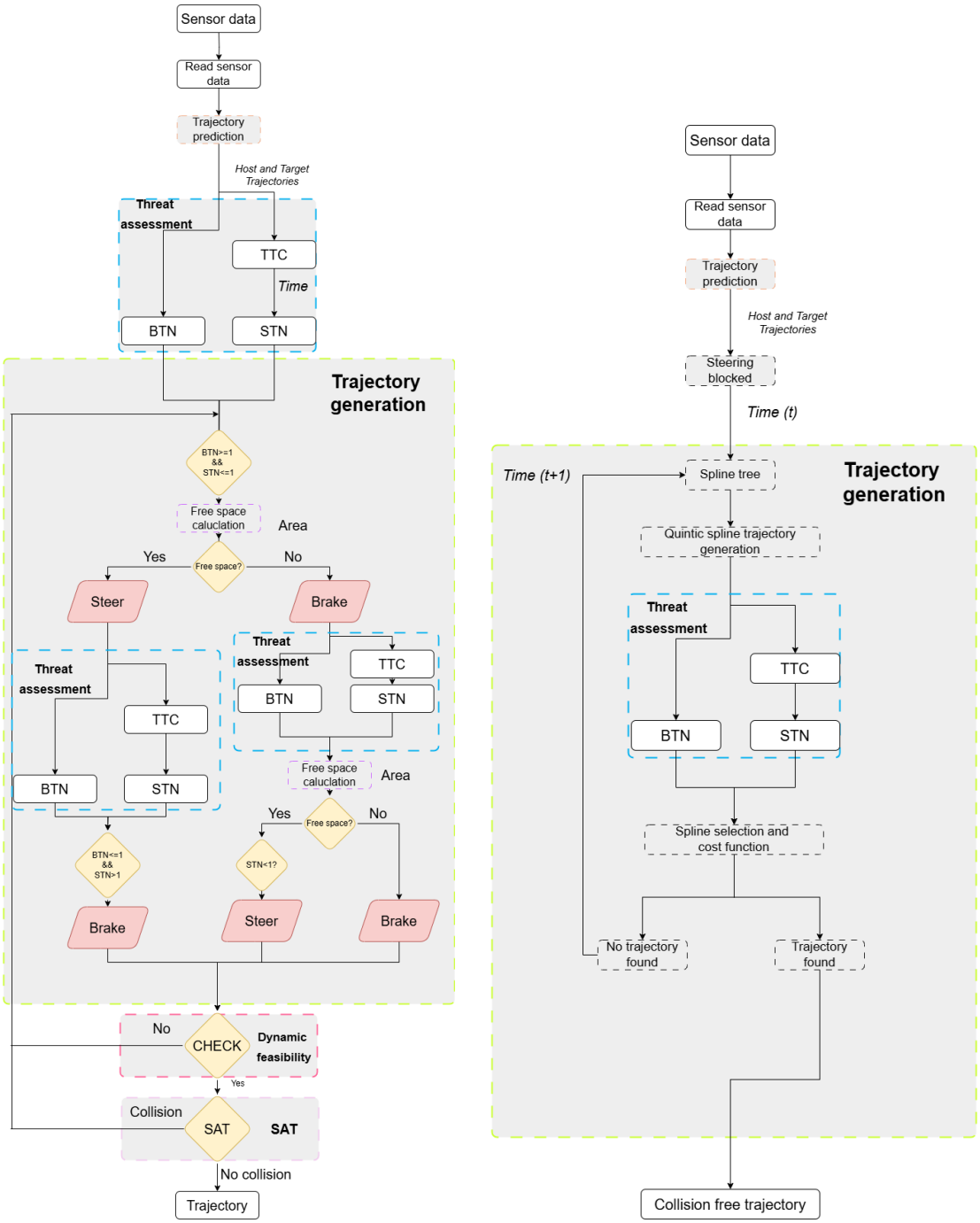
This chapter introduces the methodology applied throughout the study. The structure of the chapter reflects the sequential process used in the development and evaluation of the algorithms. Section 2.1 explains which methods were utilized as inspiration for the development of the two algorithms. Following Section 2.2, outlines an overview of the key features of the algorithms. Section 2.3 outlines the prediction of trajectories for both the host and target vehicles. Section 2.4 details the threat assessment approach, including the specific metrics employed. Section 2.5 explains the method for calculating the available free space around the host and target vehicles. It explains the conditions under which steering becomes available for the host vehicle. Section 2.6 provides an overview of the host vehicle's dynamic model and physical characteristics. Section 2.7 describes the collision detection method, including how collision checks are conducted. Section 2.8 introduces the two proposed algorithms, emphasizing their core components and operational principles. Lastly, Section 2.9 describes the framework used to generate various scenarios for testing.

2.1 Method selection based on previous research

The two algorithms proposed in this report are inspired by prior work mentioned in Section 1.1, particularly the developments in emergency collision avoidance systems [12] and the application of quintic polynomials for trajectory planning [13]. The rule-based decision-making framework from the emergency collision avoidance system described in [12], inspired the Rule-Based Algorithm, disregarding the MPC. This is based on the disadvantages with MPC, outlined in [10]. To further enhance trajectory smoothness, quintic splines, as proposed in [13], are adopted in this report in place of cubic splines for the Spline-based algorithm.

2.2 Algorithms overview

An overview of both algorithms and their key components are presented in Fig. 2.1a and Fig. 2.1b.



(a) Rule-based algorithm. **(b)** Spline-based algorithm.

Both algorithms are expected to generate a dynamic feasible trajectory that should avoid collision with all target vehicles and objects in the scenarios mentioned in Section 1.2.

2.3 Trajectory prediction

Trajectory predictions compute the trajectories for all targets on the road, including the host vehicle. It is implemented in both algorithms, displayed in Fig. 2.1a and Fig. 2.1b, block *"Trajectory prediction"*. The host vehicle heading [14], is calculated by

$$h(k+1) = h(k) + wT \tag{2.1}$$

where w denote the yaw rate, T represent time and the longitudinal velocity is

$$v_x\left(k+1\right) = v_x\left(k\right) \cdot \cos\left(wT\right) - v_y\left(k\right) \cdot \sin\left(wT\right) + a_t \cdot T \cdot \cos\left(h\left(k\right) + wT\right) \tag{2.2}$$

and the lateral velocity

$$v_y\left(k+1\right) = v_y\left(k\right) \cdot \cos\left(wT\right) + v_x\left(k\right) \cdot \sin\left(wT\right) + a_t \cdot T \cdot \sin\left(h\left(k\right) + wT\right) \tag{2.3}$$

which are applied to compute the longitudinal position

$$\begin{aligned} x\left(k+1\right) = \\ x\left(k\right) + \frac{\left(\left(v_x\left(k\right) \cdot \sin\left(wT\right) + v_y\left(k\right) \cdot \left(\cos\left(wT\right) - 1\right)\right)\right)}{w} + \\ \frac{\left(a_t \cdot \left(\cos\left(h\left(k\right) + wT\right) - \cos\left(h\left(k\right)\right)\right) + a_t \cdot wT \cdot \sin\left(h\left(k\right) + wT\right)\right)}{w^2} \end{aligned} \tag{2.4}$$

and the lateral position

$$\begin{aligned} y\left(k+1\right) = \\ y\left(k\right) + \frac{\left(\left(v_y\left(k\right) \cdot \sin\left(wT\right) - v_x\left(k\right) \cdot \left(\cos\left(wT\right) - 1\right)\right)\right)}{w} + \\ \frac{\left(a_t \cdot \left(\sin\left(h\left(k\right) + wT\right) - \sin\left(h\left(k\right)\right)\right) - a_t \cdot wT \cdot \cos\left(h\left(k\right) + wT\right)\right)}{w^2} \end{aligned} \tag{2.5}$$

To predict the targets trajectories [15], longitudinal position of the targets in the scene is computed accordingly

$$position_{long,\ t} \ = \ \left(v_t t + a_t t^2\right) \cos\left(\theta_t + \dot{\theta}_t t\right) + position_{0\ long,\ t} \tag{2.6}$$

together with the lateral position

$$position_{lat,\ t} \ = \ \left(v_t t + a_t t^2\right) \sin\left(\theta_t + \dot{\theta}_{0_t} t\right) + position_{0\ lat,\ t} \tag{2.7}$$

and the heading

$$\theta_t \ = \left(\dot{\theta}_t t + \theta_{0_t}\right) \tag{2.8}$$

2.4 Threat assessment

Threat assessment is calculated based on the trajectory predictions detailed in Section 2.3 portrayed in Fig. 2.1a and 2.1b, block *"Threat assessment"* for both algorithms. This process incorporates information about surrounding obstacles. Threat assessments are initiated for both the target directly in front of the host vehicle and all other surrounding targets. The threat assessment metrics applied include the Steering Threat Number (STN), Braking Threat Number (BTN), and Time To Collision (TTC) [16]. STN will indicate when to steer, BTN indicates when to brake and TTC indicate when a collision occurs.

2.4.1 Time To Collision

TTC is calculated to estimate the temporal interval remaining before a potential collision occurs between the host vehicle and the target vehicles

$$TTC = \frac{d_{long}}{v_{long,rel}} \tag{2.9}$$

where d_{long} represents the longitudinal distance between the host vehicle and the target vehicles, and $v_{long,rel}$ denotes the relative longitudinal velocity between the host and the target vehicles.

TTC can be computed considering varying accelerations

$$TTC = -\frac{v_{rel}}{a_{rel}} \pm \sqrt{\frac{v_{rel}^2}{a_{rel}} - \frac{d_0}{a_{rel}}} \tag{2.10}$$

where v_{rel} and a_{rel} represent the relative speed and acceleration, respectively, between the host and target vehicles, and d_0 denotes the distance separating them.

2.4.2 Brake Threat Number

BTN is computed to determine whether the minimum longitudinal acceleration is sufficient to prevent a collision. It can be derived as

$$BTN = \frac{a_{rec,long}}{a_{max,long}} \tag{2.11}$$

where $a_{rec,long}$ represents the minimum longitudinal acceleration required to ensure that the relative velocity between the target and host vehicles becomes zero at the moment of impact $v_t - v_h = 0$, and $a_{max,long}$ denotes the maximum achievable longitudinal deceleration of the vehicle.

2.4.3 Steer Threat Number

STN is calculated to determine the latest possible time at which the host vehicle must execute a steering maneuver to avoid collision with the target or targets ahead

$$STN = \frac{a_{rec,lat}}{a_{max,lat}} \tag{2.12}$$

where $a_{rec,lat}$ represents the required lateral acceleration to avoid collision described in (2.26), $a_{max,lat}$ denotes the maximum lateral acceleration described in (2.18).

2.5 Free space calculation

Based on the location of the static obstacles, gaps are calculated where steering is possible. The free space between the target vehicles is computed based on the trajectory prediction mentioned in Section 2.3. By combining the free space calculation of static and moving objects, a possible safe steering opportunity is calculated.

2.5.1 Steering blocked calculation

Steering blocked calculates when steering will be available for the host vehicle. Based on the static obstacles and predicted trajectories, Section 2.3, of the moving objects around the host vehicle, the algorithm outputs points in time where steering would possibly be available. Together with the free space calculations, Section 2.5, a decision can be made to potentially steer.

2.6 Dynamic feasibility and vehicle model

Vehicle dynamics are utilized to generate host vehicle trajectories Section 2.8, in both algorithms displayed in Fig. 2.1a, block *"Dynamic feasibility"* and incorporated in 2.1b, block *"Quintic spline trajectory generation"*. To describe the host vehicle, the motion of the linear bicycle model is utilized. A visualization of the model is portrayed in Fig. 2.2, [17].

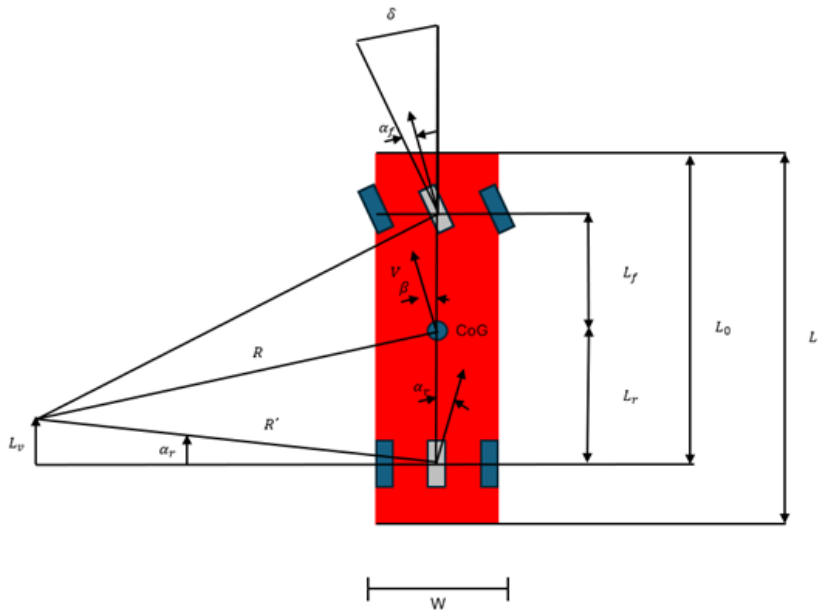


Fig. 2.2: Bicycle model, describing the dynamics of the host vehicle.

The length from the Center Of Gravity (COG) to the front is L_f and the length from COG to the rear is L_r. δ represents the steering angle of the car and it can be approximated with

$$\delta = L_w c + \alpha_f - \alpha_r \tag{2.13}$$

where L_w denote the wheelbase $L_r + L_f$ and the curvature

$$c = \frac{1}{R} \tag{2.14}$$

The rear tire slip angle α_r can be computed with

$$\alpha_r = \frac{ML_f}{L_w C_r}\frac{v^2}{R} \tag{2.15}$$

where M is the mass of the vehicle, C_r is the rear cornering stiffness and v is the velocity of the vehicle. The front slip angle α_f can be computed similarly

$$\alpha_f = \frac{ML_r}{L_w C_f}\frac{v^2}{R} \tag{2.16}$$

where C_f is the front cornering stiffness, together with the variables described in (2.15). According to [18], the maximum lateral acceleration a normal car can handle is $0.7g = 6.87m/s^2$. The maximum lateral acceleration will be utilized to check dynamical feasibility. From the bicycle model, it is possible to approximate the trajectory radius

$$R = \frac{L_w}{\delta} \tag{2.17}$$

where L_w is the wheelbase of the car [19]. Maximum lateral acceleration can be calculated with

$$a_{max,lat}(t) = \frac{v_0^2}{R(t)} = c(t)v_0^2 \tag{2.18}$$

where v_0 represents the initial velocity.

The geometry of the vehicle is approximated as a rectangular form, characterized by a width W and a length L, as depicted in Fig. 2.3. Within this representation, H denotes the host vehicle, indicated in red, while T signifies the target vehicle, marked in blue.

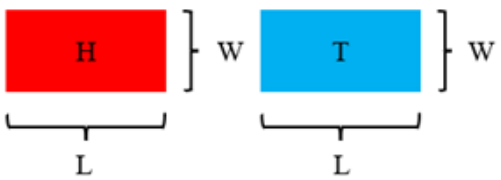


Fig. 2.3: Vehicle model, describing the host and target vehicle exterior.

2.7 Separating Axis Theorem

To check if a collision occurs when the host follows the trajectory, Separating Axis Theorem (SAT) for collision detection will be applied [20], portrayed in Fig. 2.1a, block *"SAT"*, and utilized in Fig. 2.1b, block *"Spline selection and cost function"*. The shape of the host and target vehicles are approximated as rectangles, mentioned in Section 2.6. The corners of both rectangles are projected onto the axes Y1 and X1 that are parallel to the edges of red rectangle H, an example of this is illustrated in Fig. 2.4, (a circle on the X1 axis indicates separation is possible) [15]. Overlaps are observed on axis Y1 but not on axis X1, indicating the presence of a separating axis perpendicular to X1. As a result, no collision is detected. Projecting the corners onto axes Y2 and X2 is unnecessary since a separating axis has already been identified.

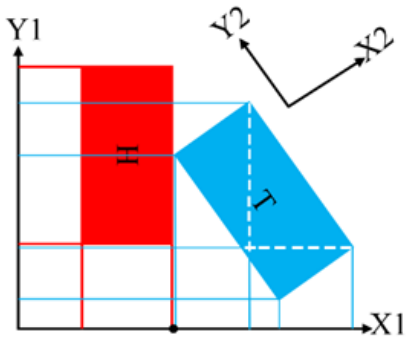


Fig. 2.4: Separating Axis Theorem (SAT).

The implemented SAT assumes that the target vehicle is moving in a straight trajectory, aligned with the world axes Y1 and X1, no rotation occurs. Because of this, the SAT does not have to check the target rotated axes, which results in a faster computation. It checks overlap over four axes, two axes from each vehicle. Because for rectangles, checking one pair of perpendicular axes per rectangle (4 axes total) covers all possible separating directions in SAT, since opposite-edge normals are redundant due to symmetrical projections. Pseudo code of the SAT method is presented in Algorithm 1.

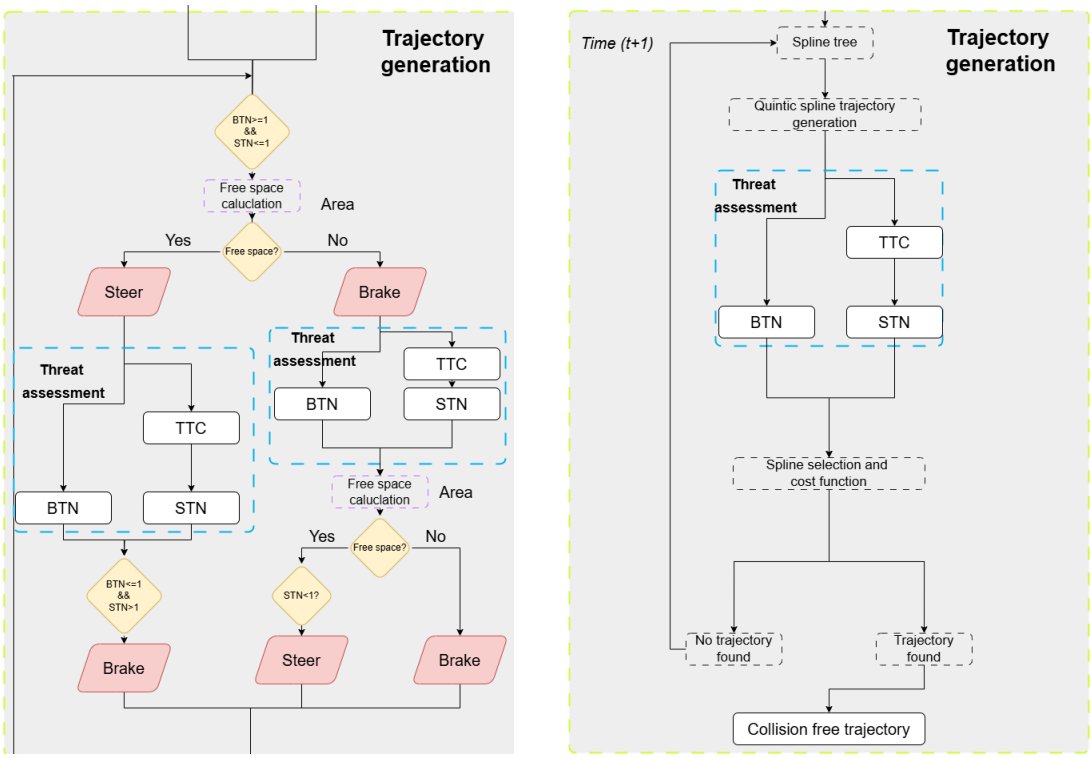
Algorithm 1: Separating Axis Theorem (SAT).

Input:
host ← vertices of host rectangle (4 × 2 matrix);
target ← vertices of target rectangle (4 × 2 matrix);
yaw_host ← orientation angle of host rectangle (radians);
Output:
flag ← interSection status (1 = true, 0 = false);

```
axisX2 ← [1, 0] ;                                    // Target's x-axis
axisY2 ← [0, 1] ;                                    // Target's y-axis
axisX1 ← [cos(yaw_host), sin(yaw_host)] ;    // Host's rotated x-axis
axisY1 ← [-sin(yaw_host), cos(yaw_host)] ;   // Host's rotated y-axis
axes ← [axisX2; axisY2; axisX1; axisY1];
flag ← 1 ;                                  // Assume interSection initially
for each axis in axes do
    projA ← host × axis^T;
    minA ← min(projA);
    maxA ← max(projA);
    projB ← target × axis^T;
    minB ← min(projB);
    maxB ← max(projB);
    if maxA < minB or maxB < minA then
        flag ← 0 ;                                   // Separating axis found
        return flag;
    end
end
return flag;
```

2.8 Trajectory generation

In the following Section a detailed explanation of how the trajectories of both algorithms are generated and selected. Fig. 2.5a illustrates the trajectory generation of the Ruled-based algorithm and Fig. 2.5b illustrates the trajectory generation of the Spline-based algorithm.



(a) Rule-based trajectory generation. **(b)** Spline-based trajectory generation.

Fig. 2.5: Trajectory generation block of both Rule and Spline-based algorithm. The whole block schemes of both algorithms are presented in 2.1a and 2.1b.

2.8.1 Ruled-based trajectory generation

Trajectories are generated as proposed in Fig. 2.5a, block *"Trajectory generation"*. If the host vehicle is predicted to collide based on threat assessment, Section 2.4, free space is calculated as described in Section 2.5. If free space is found, steering angles and time to steer are calculated with Algorithm 2. The algorithm will only steer if there is no other vehicle in front or no other collisions will be caused based on threat assessments, Section 2.4.

If the free space calculation estimates a free space further ahead in time, the host vehicle will max brake and then steer if possible, based on free space and threat assessments. The brake and steering signals will be converted to trajectories with the trajectory prediction equations mentioned in Section 2.3

Algorithm 2: Ruled-based steering signals

Input:
target_width
host_vehicle_properties
$TTC \leftarrow threat\_assessments$
$v \leftarrow vehicle\_speed$
Output:
(steering_angle, time_to_steer_one_direction)

if *STN=<1* **then**
- required_width = target_infront_width + safety_margin
- required_acceleration = $\frac{2 \cdot required\_width}{TTC^2_{vehicle-in-front}}$
- R = $\frac{v^2}{required\_acceleration}$ // R is the turning radius and v is host vehicle speed
- steering_angle = $\frac{wheelbase_{host}}{R}$
- steering_ratio = $\frac{max\_steering\_wheel\_angle_{host}}{arctan(\frac{2 \cdot wheelbase_{host}}{turning\_circle_{host}}) \cdot 57.2957}$
- steering_wheel_angle = $steering\_angle \cdot steering\_ratio$
- angle_to_clear_object = $arccos(1 - \frac{\frac{target\_infront\_width}{2}}{R})$
- time_to_steer_one_direction = $\frac{R * angle\_to\_clear\_object}{v}$

end

2.8.2 Ruled-based algorithm overview

Trajectories are generated as proposed in Fig. 2.5a, block *"Trajectory generation"*, for the host vehicle. Based on trajectory predictions, Section 2.3, and threat assessment, Section 2.4. These data are utilized together with the road map, road measurements from IPG carmaker, and static obstacles to identify unobstructed space between targets. This is to examine when steering potentially will be available.

If no trajectory is initially found, occurring in complex scenarios, where multiple maneuvers are required to avoid collisions successfully, trajectories with different lateral accelerations are sampled utilizing

$$a_{y,min} = \gamma_{min} v_x = \frac{v_x^2 (w_s + 2P_y)}{P_x^2 + P_y^2 - \frac{w_s^2}{4}} \tag{2.19}$$

w_s denotes the width of the vehicle. P_x represents the longitudinal distance between the target and the host vehicle, and P_y indicates the relative lateral distance between the host and the target vehicle. Additionally, v_x denotes the longitudinal velocity of the vehicle. The minimum yaw rate required to satisfy the collision avoidance condition can be expressed as follows

$$\gamma_{min} = \frac{v_x (w_s + 2P_y)}{P_x^2 + P_y^2 - \frac{w_s^2}{4}} \tag{2.20}$$

where the dynamically feasible trajectories, Section 2.6, and do not collide, checked by SAT, Section 2.7, for the host vehicle is selected.

2.8.3 Spline-based trajectory generation

An overview of the Spline-based algorithm trajectory generation is displayed in Fig. 2.5b, highlighting the features of the *"Trajectory generation"* block. The following Sections 2.8.3.1 - 2.8.4 explain the included parts of the Spline-based algorithm trajectory generation.

2.8.3.1 Spline tree

The spline tree, displayed in Fig. 2.5b, block *"Spline tree"*, are constructed of six dimensional nodes $n = (x, y, \theta, p, t, R)$. x represents the longitudinal position, y the lateral position, θ the heading, p the parent node, t the end time of the node, and R represents the turning radius. The tree is initialized with the host vehicle state as the root, and k number of starting nodes are randomly generated, each with the root representing the parent to initialize n number of starting trajectories. Nodes are then randomly generated with random parents until a certain number of trajectories have reached a maximum time threshold. Nodes in the tree structure are randomly generated based on vehicle dynamics and the road map with Algorithm 3.

Algorithm 3: Random new node

Input:
$roadmap$;
$a_{max} \leftarrow max_{lat\_acc}$;
$p \leftarrow parent\_node$;
$v \leftarrow vehicle\_speed$;
$max_{time} \leftarrow max\_time$;
Output:
random_node $\leftarrow$ n;
while $N \neq 0$ **do**
  $a_{lat} \leftarrow a_{lat} \sim U(-a_{max}, a_{max})$;
  $R \leftarrow \frac{v^2}{a_{lat}}$;
  $t \leftarrow t \sim U(0,1) \cdot \frac{max_{time}}{2}$;
  $angular_{vel} \leftarrow \frac{v}{R}$;
  $x \leftarrow R \cdot sin(angular_{vel} \cdot t)$;
  $y \leftarrow R \cdot (1 - cos(angular_{vel} \cdot t))$;
  $x = x \cdot 2 + p_x$;
  $y = y \cdot 2 + p_y$;
  if *y is inside roadmap* **then**
    $n \leftarrow (x, y, \theta, p, t, R)$;
    break
  end
end

2.8.3.2 Quintic spline trajectory generation

Based on the spline tree algorithm, Section 2.8.3.1, trajectories are generated by connecting the nodes with their parent node, displayed in Fig. 2.5b, block *"Quintic spline trajectory generation"*. The lateral and longitudinal motion between each node and parent are described with quintic polynomial trajectories [13] as follows

$$\begin{cases} x = a_5t^5 + a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0 \\ y = b_5t^5 + b_4t^4 + b_3t^3 + b_2t^2 + b_1t + b_0 \end{cases} \tag{2.21}$$

The coefficients are calculated based on the boundary condition which is a combination of the initial and final states. The twelve unknown coefficients are determined by six boundary conditions for each direction. With $x(t_i) = x_i$ and $y(t_i) = y_i$, the quintic polynomial model can be described by combining the boundary conditions as

$$\begin{cases} (x_s\dot{x}_s\ddot{x}_s x_e\dot{x}_e\ddot{x}_e)^T = MA \\ (y_s\dot{y}_s\ddot{y}_s y_e\dot{y}_e\ddot{y}_e)^T = MB \end{cases} \tag{2.22}$$

where x_s, y_s represent the initial state and x_e, y_e represent the final state. M represents the time parameter state matrix and can be described as

$$M = \begin{bmatrix} t_s^5 & t_s^4 & t_s^3 & t_s^2 & t_s & 1 \\ 5t_s^4 & 4t_s^3 & 3t_s^2 & 2t_s & 1 & 0 \\ 20t_s^3 & 12t_s^2 & 6t_s^1 & 1 & 0 & 0 \\ t_e^5 & t_e^4 & t_e^3 & t_e^2 & t_e & 1 \\ 5t_e^4 & 4t_e^3 & 3t_e^2 & 2t_e & 1 & 0 \\ 20t_e^3 & 12t_e^2 & 6t_e^1 & 1 & 0 & 0 \end{bmatrix} \tag{2.23}$$

The generated quintic polynomial for every parent node pair is then connected to form a quintic spline trajectory.

2.8.3.3 Spline selection and cost function

Each trajectory generated, described in Section 2.8.3.2, is evaluated with the Separating Axis Theorem, Section 2.7, to filter out trajectories that are predicted to cause a collision. The remaining collision-free trajectories are then further evaluated with the threat assessments, Section 2.4.

All of the remaining quintic spline trajectories are further evaluated or filtered based on the minimum distance threshold allowed to the other objects and then by the minimum number of nodes, with

$$min(trajectory_{number-of-nodes}) \tag{2.24}$$

The remaining quintic splines are the ones with the least amount of turns and a safe distance to the surrounding target vehicles and objects. To select the the most suitable trajectory, the trajectory with the highest safety distance to all target vehicles and objects are selected.

2.8.4 Spline-based algorithm overview

Trajectories are generated as proposed in Fig. 2.5b. Trajectories for each object in the scenarios are predicted as in Section 2.3. For each time step, a spline tree is constructed, Section 2.8.3.1, then converted to quintic spline trajectories, Section 2.8.3.2, followed by threat assessment evaluation, described in Section 2.4. The generated splines are evaluated based on the cost function, Section 2.8.3.3, and evaluated with SAT, Section 2.7, to remove splines that collide with other objects. To speed up the computation, the steering blocked algorithm, Section 2.5.1, calculates which time step to initialize the spline tree. Spline-based algorithm trajectory generation is described with Algorithm 4, outputting a collision-free trajectory.

Algorithm 4: Spline trajectory generation

Input:
predicted_trajectories;
max_{time};
$n_{host\_start} \leftarrow (x, y, \theta, p, t, R)$;
Output:
$collision\_free\_trajectory$

time ← steering _blocked(predicted_trajectories);
host_plong ← brake(time);

while *time*<max_{time} **do**
  $n_{host\_current} \leftarrow update(x, y, \theta, p, t, R)$;
  $tree \leftarrow Spline\_tree(n_{host\_current})$;
  $trajectories \leftarrow Spline\_generation(tree)$
  for *each trajectory in trajectories* **do**
    $valid\_trajectories \leftarrow SAT(trajectory)$
  end
  $collision\_free\_trajectory \leftarrow cost\_function(valid\_trajectories)$;
  if *collision_free_trajectory* **then**
    return
  end
end

2.9 Test and verification

The performance of the two algorithms are tested through simulations, conducted in IPG CarMaker. The initial conditions for the scenario generation are described in Section 2.9.1.

2.9.1 Scenario generation

The scenarios in Section 1.2.1 are generated through manual calculations, an illustrative example of which is provided in Appendix A. The scenarios are designed

to make it possible for the algorithms to generate trajectories that avoid collisions with combined braking and steering, where AEB and AES fail by themselves. Initial speed and constant lateral acceleration are determined for the host and the target object in front of the host. The longitudinal positions are determined by the equation

$$s = v_0 t + \frac{at^2}{2} \tag{2.25}$$

such that a collision would occur with max brake. Static obstacles and target vehicles are placed in positions to block the host vehicle for a certain amount of time. Then when there is space to turn, to avoid collision the lateral required acceleration is required to be less than $0.7g = 6.87m/s^2$. $a_{req,lat}$ can be formulated as

$$a_{req,lat} = 2 \cdot \frac{R_{lat}}{t_c^2} \tag{2.26}$$

where R_{lat} represents the lateral distance required for the host vehicle to traverse to safely overtake the vehicle ahead and t_c^2 denotes the TTC.

3

Results and analyses

This Chapter presents the starting conditions and the calculated threat assessments, Section 3.1. Followed by an evaluation of the two proposed algorithm's performance, the Ruled-based algorithm, Section 3.2, and the Spline-based algorithm, Section 3.3. Finally, Section 3.4 discusses the decisions made by both algorithms and their performance differences.

3.1 Scenario testing conditions

The properties of the car used are presented in Table 3.1.

Table 3.1: Properties of car used in simulations.

Car length $[m]$	Width $[m]$	Steer ratio $[rad]$	Wheel base $[m]$	Max brake force $[N]$
4.936	1.868	21.933	2.975	0.843

Table 3.2 together with Table 3.3 describe the initial scenario conditions with host and targets vehicles initial speeds and the initial calculated threat assessments, Section 2.4.

Table 3.2: Each scenario's initial velocities for target vehicles and their initial threat assessment metrics are BTN, STN, and TTC.

Scenario	Speed host vehicle $[km/h]$	Speed target vehicle i.f.o host $[km/h]$	Acceleration target vehicle $[m/s^2]$	BTN	STN	TTC $[s]$
1	90	70	-8	1.059	0.2216	1.18
2	90	70	-8	1.059	0.2216	1.18
3	100	70	-8	1.021	0.1318	1.53
4	50	7	-8	4.417	0.2174	0.75
5	90	70	-8	1.059	0.2216	1.18

Table 3.3: The initial speeds of adjacent lane target vehicles in scenarios two to five.

Scenario	Target one adjacent lane speed $[km/h]$	Target one adjacent lane acceleration $[m/s^2]$	Target two adjacent lane speed $[km/h]$	Target two adjacent lane acceleration $[m/s^2]$
2	100	0	-	-
3	90	0	90	0
4	70	0	-	-
5	70	-8	-	-

By examining the BTN values from Table 3.2, it is not possible to avoid collision by only braking, because BTN > 1. This is shown in Fig. 3.1, where the host and target vehicle intersect when applying AEB, indicating a collision.

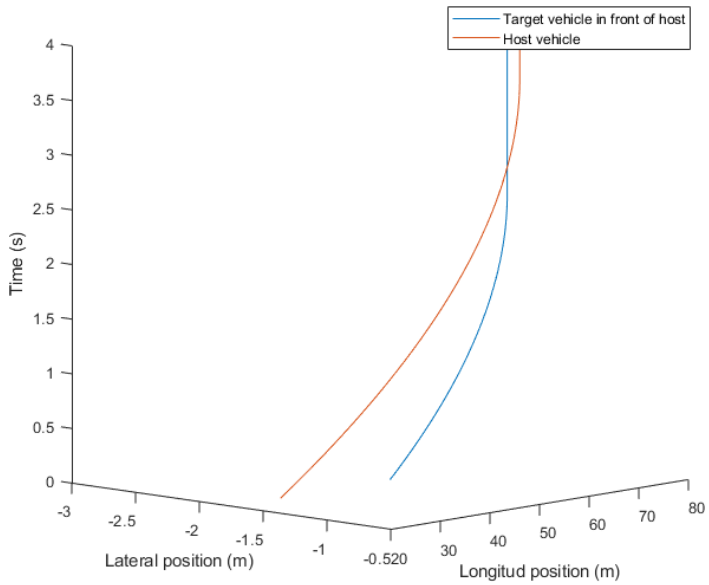


Fig. 3.1: Visualization of only utilizing AEB in scenario three.

In addition, a maneuver utilizing AES would result in collisions or become dynamically infeasible. The time for steering available is calculated with free space and steering blocked calculations, Section 2.5.

Table 3.4: Represent the STN values if steering would be available, - represent crash before available steering.

Scenario	Steering available $[s]$	STN
1	0.83	2.3811
2	-	-
3	-	-
4	0.41	1.454

For scenario five, Fig. 1.5, there are no cars directly blocking the host vehicle, but the car in front in the adjacent lane is also braking. If only AES is utilized collision would occur in adjacent lane.

Table 3.5: BTN value for adjacent lane target, after AES maneuver.

Scenario	BTN
5	1.059

3.2 Ruled-based algorithm evaluation

Based on the results in Table 3.6, the Rule-based algorithm managed to find collision-free trajectories for scenarios one to five, for lower speeds, highlighted in a green "Yes", indicating it is a feasible trajectory. In scenarios one to four the algorithm fails to predict collision-free trajectories, highlighted in a red "No", for speeds larger than the initial speeds, indicating it is not a feasible trajectory. This is because there are no dynamically feasible trajectories for speeds larger than the initial speeds, which can be examined in column $a_{rec_{lat}}$, Table 3.6. If the algorithm can't find any dynamically feasible trajectories, meaning $a_{rec_{lat}} > 6.87\ m/s^2$, the host vehicle will max brake and rear end the target vehicle in front, resulting in a collision.

Table 3.6: Ruled-based algorithm performance across different scenarios. The data in the table is extracted from Carmaker and Matlab, where the Speed represents the initial speed of the host vehicle. $a_{rec_{lat}}$ is the required lateral acceleration of the host vehicle to execute the collision avoidance maneuver. $a_{rec_{long}}$ is the required longitudinal acceleration to avoid a collision by braking. BTN_{init} is the initial Brake Threat Number and STN_{init} is the initial Steer Threat Number.

Scenario	Speed $[km/h]$	Collision avoided	Feasible	$a_{rec_{lat}}$ $[m/s^2]$	$a_{rec_{long}}$ $[m/s^2]$	BTN_{init}	STN_{init}
1	90	Yes	Yes	6.625	-8.760	1.059	0.222
	100	No	No	> 6.87	-10.81	1.306	0.315
	110	No	No	> 6.87	-13.12	1.586	0.437
2	90	Yes	Yes	5.601	-8.760	1.059	0.222
	100	No	No	> 6.87	-10.81	1.306	0.315
	110	No	No	> 6.87	-13.12	1.586	0.437
3	100	Yes	Yes	2.994	-6.84	1.021	0.132
	110	No	No	> 6.87	-10.22	1.235	0.172
	120	No	No	> 6.87	-12.16	1.470	0.218
4	50	Yes	Yes	5.372	-10.09	4.417	2.137
	60	No	No	> 6.87	-14.53	5.755	2.519
	70	No	No	> 6.87	-19.77	7.344	3.014
	80	No	No	> 6.87	-25.82	8.952	3.429
5	90	Yes	Yes	3.045	-8.760	1.059	0.222
	100	No	No	> 6.87	-10.81	1.306	0.315
	110	No	No	> 6.87	-13.12	1.586	0.437

The threshold values used to determine dynamic feasibility are general for normal cars. Incorporating models such as tire friction, road friction, and load transfer could possibly result in a more exact dynamic feasibility estimation, which potentially leads to a more realistic result for a real-world application.

3.2.1 Visualization Rule-based algorithm trajectory selection

Trajectory result from scenario three with the Rule-based algorithm. Fig. 3.2 represents the generated trajectory explained in Section 2.8.1.

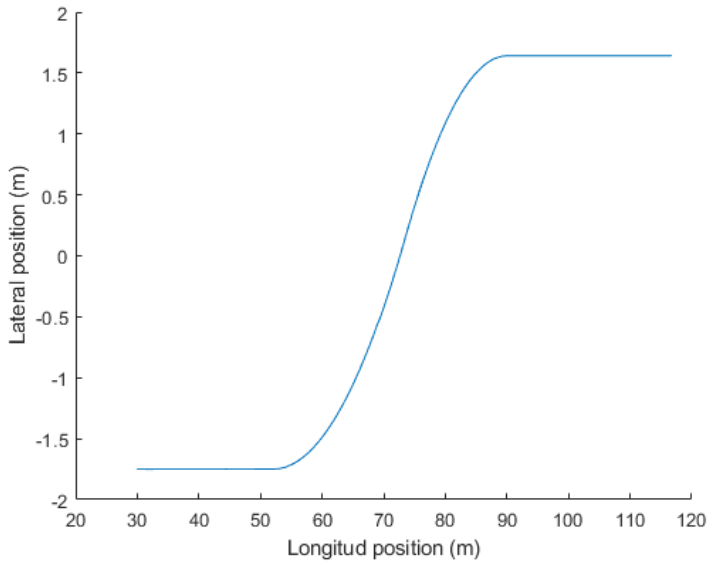


Fig. 3.2: Ruled-based collision avoidance algorithm trajectory, 90 km/h.

3.3 Spline-based algorithm evaluation

The Spline-based algorithm works for all scenarios, where the maneuver is dynamically feasible, highlighted in a green "Yes", based on the result in Table 3.7. Visualization of the spline selected for scenario three is presented in Fig. 3.3 to 3.5. In scenarios one to four, the algorithm fails to predict collision-free trajectories, highlighted in a red "No", for speeds higher than the initial speeds, Table 3.2. Similar to the Rule-based algorithm, if the algorithm can't find any dynamically feasible trajectories, the host vehicle will max brake and rear end the target vehicle in front, resulting in a collision.

Table 3.7: Spline-based algorithm, performance across different scenarios. The data in the Table is extracted from Carmaker and Matlab, where the Speed represents the initial speed of the host vehicle. $a_{rec_{lat}}$ is the required lateral acceleration of the host vehicle to execute the collision avoidance maneuver. $a_{rec_{long}}$ is the required longitudinal acceleration to avoid a collision by braking. BTN_{init} is the initial Brake Threat Number and STN_{init} is the initial Steer Threat Number.

Scenario	Speed $[km/h]$	Collision avoided	Feasible	$a_{rec_{lat}}$ $[m/s^2]$	$a_{rec_{long}}$ $[m/s^2]$	BTN_{init}	STN_{init}
1	90	Yes	Yes	< 6.87	-8.760	1.059	0.222
	100	No	No	> 6.87	-10.81	1.306	0.315
	110	No	No	> 6.87	-13.12	1.586	0.437
2	90	Yes	Yes	< 6.87	-8.760	1.059	0.222
	100	No	No	> 6.87	-10.81	1.306	0.315
	110	No	No	> 6.87	-13.12	1.586	0.437
3	100	Yes	Yes	< 6.87	-6.84	1.021	0.132
	110	No	No	> 6.87	-10.22	1.235	0.172
	120	No	No	> 6.87	-12.16	1.470	0.218
4	50	Yes	Yes	< 6.87	-10.09	4.417	2.137
	60	No	No	> 6.87	-14.53	5.755	2.519
	70	No	No	> 6.87	-19.77	7.344	3.014
	80	No	No	> 6.87	-25.82	8.952	3.429
5	90	Yes	Yes	< 6.87	-8.760	1.059	0.222
	100	Yes	Yes	< 6.87	-10.81	1.306	0.315
	110	Yes	Yes	< 6.87	-13.12	1.586	0.437

Similar to the discussion in Section 3.2, investigating factors such as tire friction, road friction, and load transfer models could play an essential role in a more exact dynamic feasibility estimation. This is further discussed in Section 4.2.

3.3.1 Visualization Spline-based algorithm trajectory selection

Result from scenario three with the Spline-based algorithm. Fig. 3.3 represents the generated quintic splines trajectories explained in Section 2.8.3.2.

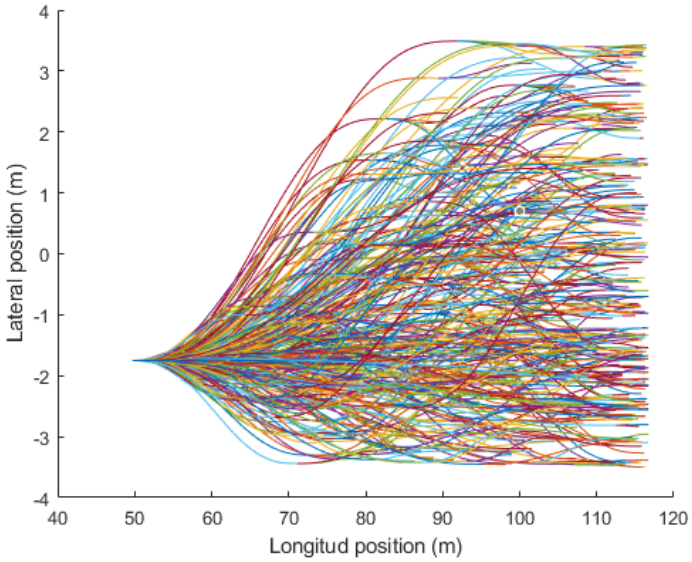


Fig. 3.3: Generated splines.

Fig. 3.4 represents trajectories that can avoid collisions with all target vehicles and road objects in the scenarios, Section 1.2. How these trajectories are selected is described in 2.8.3.3.

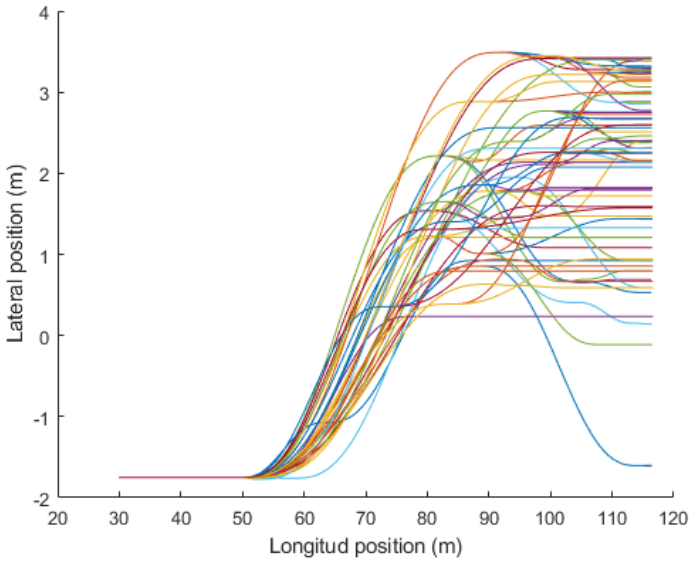


Fig. 3.4: Collision-free paths.

Fig. 3.5 represents the algorithm-selected trajectory based on the cost function, Section 2.8.3.3.

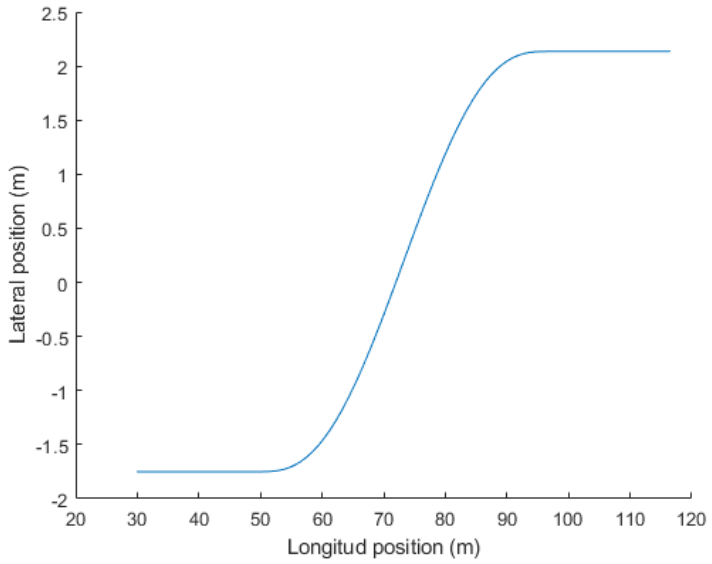
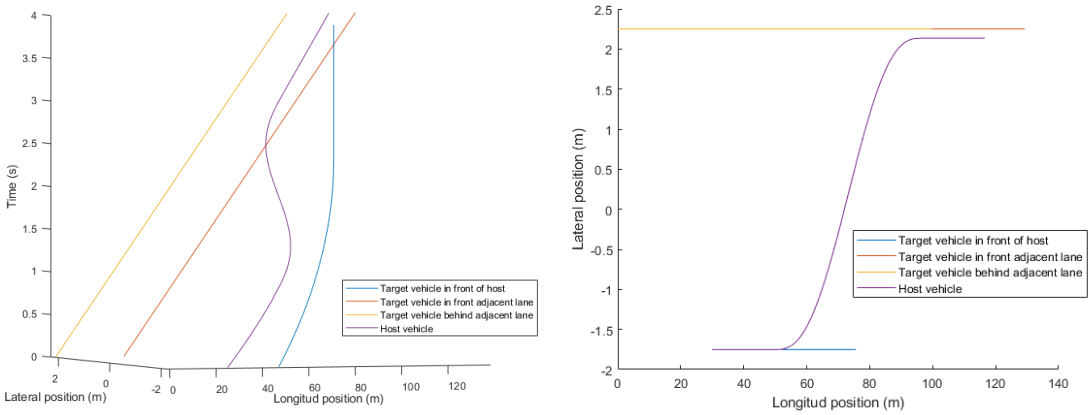


Fig. 3.5: Spline selected based on cost function, 90 km/h.

Fig. 3.6a and 3.6b present the selected trajectory and the target vehicle trajectories in scenario three, Section 1.2.



(a) 3D view. **(b)** 2D view.

Fig. 3.6: a) Represent the trajectories in time and b) represent the position.

3.4 The algorithms decision-making

The two different algorithms utilize different decision-making to predict collision-free trajectories, an overview of both algorithms is presented in Fig. 2.1a and 2.1b. In some scenarios, both algorithms result in different maneuvers. From scenario five the Rule-based algorithm collision avoidance maneuver is presented in Table 3.8 and visualized in Fig. 3.7. By examining the variable SteerAng, second column, and Brake variable, third column, the values describe when steering and braking are initiated. The collision avoidance maneuver for the Spline-based algorithm is presented in Table 3.9 together with a visualization of the trajectory in Fig. 3.8.

Table 3.8: Time is in seconds, SteerAng is the steering wheel angle in radians, Brake is how much the brake pedal is pushed in a scale from zero to one.

Time $[s]$	SteerAng $[rad]$	Brake
0.000	0.000	0.000
0.100	0.524	0.000
0.719	0.524	0.000
0.819	0.000	0.000
0.919	-0.524	0.000
1.538	-0.524	0.000
1.637	0.000	0.000
1.638	0.000	1.000
1.738	0.000	1.000
3.990	0.000	1.000

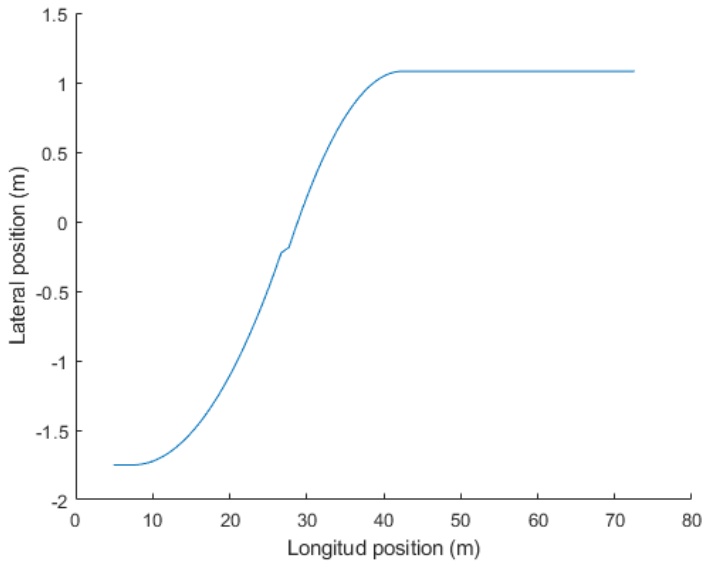


Fig. 3.7: Rule-based algorithm collision-free trajectory.

Table 3.9: Time is in seconds, SteerAng is the steering wheel angle in radians, and Brake is how much the brake pedal is pushed on a scale from zero to one.

Time $[s]$	SteerAng $[rad]$	Brake
0.000	0.000	0.000
0.010	0.000	0.000
0.020	0.383	0.000
0.750	0.383	0.000
0.800	-0.383	0.000
1.540	-0.383	0.000
1.560	0.000	0.000
1.570	-0.230	0.000
2.540	-0.230	0.000
2.640	0.230	0.000
3.620	0.230	0.000
3.690	0.000	0.000

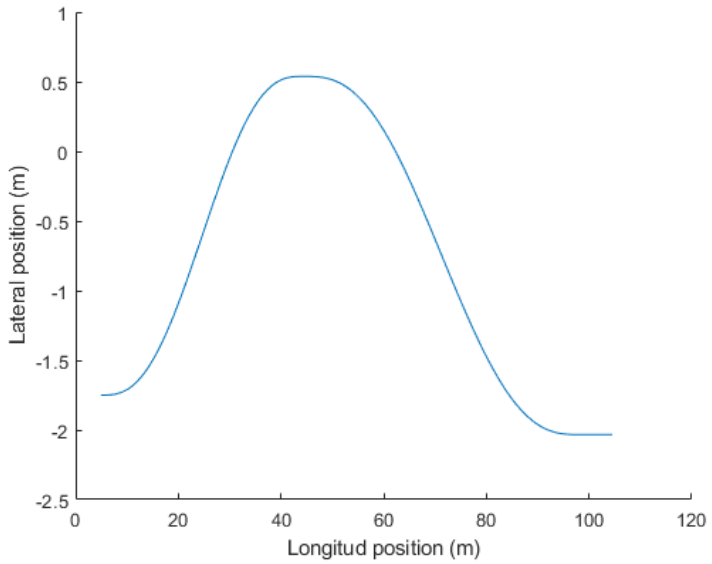


Fig. 3.8: Spline-based algorithm collision-free trajectory.

As can be seen in Table 3.9 and visualized in Fig 3.8 the generated trajectory by the Spline-based algorithm tends to only use steering to avoid collisions. The Rule-based algorithm on the other hand generates a trajectory that steers and then brake presented in Table 3.8 and Fig 3.7.

Because both algorithms result in two different maneuvers as mentioned previously, both algorithms result in different performances. Comparing the columns "Collision avoided" in Table 3.10 and Table 3.11 for both algorithms, it is evident that the Spline-based algorithm outperforms the other. The reason for this is that the Spline-

based algorithm finds a trajectory with a required lateral acceleration $a_{rec_{lat}} < 6.87$ m/s^2 for all speeds in scenario five, compared to the Rule-based. This highlights the robustness of the Spline-based algorithm compared to the Rule-based algorithm. However the robustness comes at the expense of computational time.

Table 3.10: Ruled-based algorithm performance across scenario five.

Scenario	Speed $[km/h]$	Collision avoided	Feasible	$a_{rec_{lat}}$ $[m/s^2]$	$a_{rec_{long}}$ $[m/s^2]$	BTN_{init}	STN_{init}
5	90	Yes	Yes	3.045	-8.760	1.059	0.222
	100	No	No	> 6.87	-10.81	1.306	0.315
	110	No	No	> 6.87	-13.12	1.586	0.437

Table 3.11: Spline-based algorithm, performance across scenario five.

Scenario	Speed $[km/h]$	Collision avoided	Feasible	$a_{rec_{lat}}$ $[m/s^2]$	$a_{rec_{long}}$ $[m/s^2]$	BTN_{init}	STN_{init}
5	90	Yes	Yes	< 6.87	-8.760	1.059	0.222
	100	Yes	Yes	< 6.87	-10.81	1.306	0.315
	110	Yes	Yes	< 6.87	-13.12	1.586	0.437

4
Conclusion

4.1 Contribution

Do the developed algorithms avoid colliding with other road users and objects in situations where AEB and AES would not? Yes, both algorithms will guarantee to avoid collisions in the scenarios mentioned in Section 1.2.1, if the generated trajectory is dynamically feasible.

4.2 Future work

For further development of the two algorithms, several key aspects need to be considered to make both algorithms more robust and adaptable to real-world dynamics.

4.2.1 Other road geometry

This thesis only considered straight roads when testing. Road curvature would be more suitable for real-world applications. A further development of this project would take the road curvature into account when generating trajectories, to make sure the host vehicle is able to safely continue after the predicted trajectories horizon and not end up with a heading pointing straight forward in a curve.

4.2.2 Prediction model for pedestrians

In this thesis a basic model has been applied to estimate pedestrian's position, section 2.3. This is not very accurate, because pedestrian's movements are difficult to predict. A better model would result in safer and more robust trajectories generated by the two algorithms.

4.2.3 Road friction and tire models

While this study has focused primarily on kinematic relationships, it has intentionally excluded dynamic factors including road surface conditions, tire-road friction characteristics, and load transfer. Future research should incorporate these variables to enhance the accuracy of braking distance calculations and better quantify the lateral required acceleration during steering maneuvers before reaching oversteer conditions.

A more comprehensive analysis is required to account for load transfer effects, as these significantly influence the available friction during both braking and steering maneuvers. Specifically, investigating how load transfer dynamics affect a vehicle's steering capability under varying weather conditions would provide valuable insights into the practical limitations of collision avoidance systems.

Bibliography

[1] L. Yang et al., "A systematic review of autonomous emergency braking system: Impact factor, technology, and performance evaluation", *Journal of Advanced Transportation*, vol. 2022, pp. 1–13, Apr. 2022. DOI: 10.1155/2022/1188089.

[2] W. health organization, "Road traffic injuries", vol. 2023, Feb. 2023. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries.

[3] T. analys, "Road traffic injuries", vol. 2024, Apr. 2024. [Online]. Available: https://www.trafa.se/vagtrafik/vagtrafikskador.

[4] B. Fildes et al., "Effectiveness of low speed autonomous emergency braking in real-world rear-end crashes", *Accident Analysis Prevention*, vol. 81, pp. 24–29, Aug. 2015. DOI: https://doi.org/10.1016/j.aap.2015.03.029.

[5] J. B. Cicchino and D. S. Zuby, "Characteristics of rear-end crashes involving passenger vehicles with automatic emergency braking", *Traffic Injury Prevention*, vol. 20, pp. 112–118, Aug. 2019. DOI: https://doi.org/10.1080/15389588.2019.1576172.

[6] M. Sieber, K.-H. Siedersberger, A. Siegel, and B. Färber, "Automatic emergency steering with distracted drivers: Effects of intervention design", in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Gran Canaria, Spain, 2015, pp. 2040–2045. [Online]. Available: https://ieeexplore.ieee.org/document/7313422, Acquired: 2025-02-12.

[7] P. Lin, E. Javanmardi, and M. Tsukada, "Clothoid curve-based emergency-stopping path planning with adaptive potential field for autonomous vehicles", *IEEE Transactions on Vehicular Technology*, vol. 73, no. 7, pp. 9747–9762, 2024. DOI: 10.1109/TVT.2024.3380745.

[8] MathWorks, *MATLAB Documentation*, 2024. [Online]. Available: https://se.mathworks.com/help/mpc/gs/what-is-mpc.html (Acquired 2025-02-21).

[9] D. J. Lamburn, P. W. Gibbens, and S. J. Dumble, "Efficient constrained model predictive control", *European Journal of Control*, vol. 20, no. 6, pp. 301–311, 2014. DOI: https://doi.org/10.1016/j.ejcon.2014.08.001.

[10] R. O. M. Ammour and M. Basset, "Collision avoidance for autonomous vehicle using mpc and time varying sigmoid safety constraints", vol. 54, pp. 39–40, 2021. DOI: https://doi.org/10.1016/j.ifacol.2021.10.138.

[11] D. Madås et al., "On path planning methods for automotive collision avoidance", in *2013 IEEE Intelligent Vehicles Symposium (IV)*, Gold Coast, Australia, Jun. 2013, pp. 931–937. [Online]. Available: https://ieeexplore.ieee.org/document/6629586, Acquired: 2025-02-12.

[12] Q. Cui, R. Ding, C. Wei, and B. Zhou, "A hierarchical framework of emergency collision avoidance amid surrounding vehicles in highway driving", *Control Engineering Practice*, vol. 109, pp. 2–21, Apr. 2021. DOI: https://doi.org/10.1016/j.conengprac.2021.104751.

[13] D. Zhang, X. Jiao, and T. Zhang, "Lane-changing and overtaking trajectory planning for autonomous vehicles with multi-performance optimization considering static and dynamic obstacles", *Robotics and Autonomous Systems*, vol. 182, 2024. DOI: https://doi.org/10.1016/j.robot.2024.104797.

[14] P. Lytrivis, G. Thomaidis, M. Tsogas, and A. Amditis, "An advanced cooperative path prediction algorithm for safety applications in vehicular networks", *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 669–679, 2011. DOI: 10.1109/TITS.2011.2123096.

[15] A. Andersson, "Multi-target threat assessment for autonomous emergency braking", Master thesis, Department of Signals and Systems, Chalmers University of Technology, Gothenburg, Sweden, 2016. [Online]. Available: https://publications.lib.chalmers.se/records/fulltext/245725/245725.pdf.

[16] M. Brannstrom, J. Sjoberg, and E. Coelingh, "A situation and threat assessment algorithm for a rear-end collision avoidance system", in *2008 IEEE Intelligent Vehicles Symposium*, 2008, pp. 102–107. DOI: 10.1109/IVS.2008.4621250.

[17] M. Brännström, E. Coelingh, and J. Sjöberg, "Model-based threat assessment for avoiding arbitrary vehicle collisions", *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 658–669, 2010. DOI: 10.1109/TITS.2010.2048314.

[18] T. E. Antonios, A. Konstantinos, and P. Basil, "Vehicles lateral acceleration and speed profiles investigation at the entry area of interchange ramps as a criterion of geometric road design", *IFAC-PapersOnLine*, vol. 69, pp. 13–20, 2023. DOI: https://doi.org/10.1016/j.trpro.2023.02.139.

[19] J. A. Matute, M. Marcano, S. Diaz, and J. Perez, "Experimental validation of a kinematic bicycle model predictive control with lateral acceleration consideration", *IFAC-PapersOnLine*, vol. 52, no. 8, pp. 289–294, 2019. DOI: https://doi.org/10.1016/j.ifacol.2019.08.085.

[20] P. Mileff, "Collision detection in 2d games", vol. 11, pp. 22–23, Sep. 2023. DOI: 10.32968/psaie.2023.3.2.

A

Appendix 1

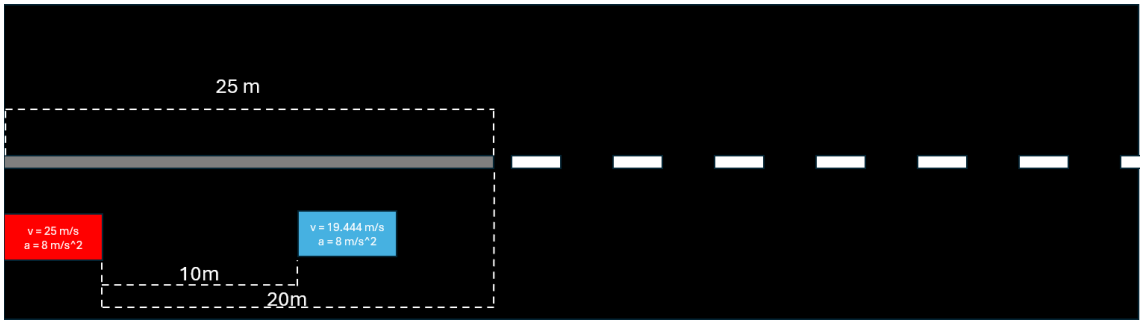


Fig. A.1

The guardrail terminates at a distance of 25 meters, indicating that a lane change maneuver may become feasible approximately 20 meters from the host vehicle's initial position. The time required for the host vehicle to traverse the longitudinal distance of 20 meters can be calculated based on its velocity and acceleration

$$20 = 25t - \frac{8t^2}{2} \Rightarrow t = 0.942s \tag{A.1}$$

$$s_{target} = 10 + 19.444 \cdot 0.942 - \frac{8 \cdot 0.942^2}{2} = 24.76720648m \tag{A.2}$$

At $t = 0.942$ seconds, the longitudinal distance to the target vehicle is calculated to be 4.767 meters, while the host vehicle speed at this instant is calculated to

$$v_{host} = 25 - 8 \cdot 0.942 = 17.464m/s \tag{A.3}$$

Target speed is calculated to

$$v_{target} = 19.4444 - 8 \cdot 0.942 = 11.9084m/s \tag{A.4}$$

TTC to target in front of the host can be calculated as follows

$$17.464 \cdot t - \frac{8t^2}{2} = 4.767 + 11.9084 \cdot t - \frac{8t^2}{2} \Rightarrow t = 0.858053s \tag{A.5}$$

The host required lateral acceleration results in

$$a_{rec,lat} = \frac{2 \cdot 2}{0.858053^2} = 5.43m/s^2 \tag{A.6}$$

Analyzing the result: $a_{req,lat} < 0.7g$, which implies the required acceleration is feasible to handle by a normal car.

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

CHALMERS
UNIVERSITY OF TECHNOLOGY