Advanced Driver Assistance System (ADAS) for an Ultra-light Electric Vehicle

Frida Lindbom

Jakob Petersson

LUND UNIVERSITY

Department of Automatic Control

MSc Thesis
TFRT-6248
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Tryckeriet i E-huset
Lund 2024

Abstract

The aim of this thesis was to investigate, develop and implement an Advanced Driver Assistance System (ADAS) feature for an ultra-light electric vehicle, focusing specifically on an Automatic Emergency Brake (AEB). The fundamental requirements, for an ultra-light electric vehicle, were found and investigated through experiments. A person detection algorithm with camera and LiDAR was developed with the ability to measure distance to the detected person and the brake algorithm was designed and implemented with three alternative control strategies. The AEB was tested through experiments and the three control strategies were evaluated as well as the real-time properties and the overall performance. All three control strategies functioned, however the PI controller with a velocity strategy performed the best in regards to comfort. The AEB showed great promise in increasing the safety for ultra-light electric vehicles.

Acknowledgements

We would first like to thank our supervisor Hannes Bergkvist and the great people at LEVTEK SWEDEN AB for inviting us and giving us access to your prototypes and technology. A special thank you to Ivar Bergkvist for all of your technical help and guidance. We would also like to thank our supervisor Björn Olofsson from the Department of Automatic Control at LTH. We really appreciate always leaving our meetings with things being clearer than they were before and your support and time. We would also like to thank our examiner Yiannis Karayiannidis. Lastly we want to thank our friends, family and partners for supporting us throughout this thesis as well as the last five years.

Contents

1

Introduction

In this chapter a brief introduction to micromobility and advanced driver assistance systems will be presented, along with the objective of this thesis and its contributions to research.

1.1 Background

This thesis examines the benefits of driver assistance systems on ultra-light electric vehicles to improve driver safety. Based on research by [Abduljabbar et al., 2021], [Sexton et al., 2023] and [Ziebinski et al., 2017] micromobility and advanced driver assistance systems will be described.

Ultra-light Electric Vehicles

Micromobility is a form of transport optimized for short-distance travel, often in urban environments. The vehicles include bicycles, scooters and e-bikes as well as new innovative products such as ultra-light electric vehicles. They are often powered by electricity or human-power, light weight and operate at lower speeds, generally not exceeding 45 km/h. Micromobility vehicles have seen a drastic increase in popularity in recent years as the shared-service options using mobile apps have reached the market, but micromobility vehicles can be both privately owned and shared. The benefits of micromobility solutions are mainly that they are energy efficient, help in the reduction of emissions and pollution as well as relieve traffic congestion in urban settings [Abduljabbar et al., 2021].

As the popularity of micromobility vehicles has grown, research has been made about its safety. A literary review by Sexton et al. (2023) concluded that e-scooters are generally perceived as unsafe, especially for riding in car traffic. This can lead to riders using the side-walk instead, which both increase the risk for pedestrian injuries as well as leading to pedestrian discomfort.

Advanced Driver Assistance System

Advanced Driver Assistance System (ADAS) is a technical feature that assist the driver by using sensor data from the vehicle to alert the driver in certain situations [Ziebinski et al., 2017]. Such systems are highly valued features in cars because of their ability to reduce the chances of road traffic accidents, given the high number of incidents that occur every year [United Nations Economic Commision For Europe, 2021]. The response of the ADAS is based on input from the different sensors in the vehicle. The system should act against dangerous situations by analyzing data from the vehicle's behavior using proprioceptive sensors. The decision can also be based on the surroundings of the vehicle, e.g., ultrasonic, radar, LiDAR and other vision sensors [Ziebinski et al., 2017].

Furthermore ADAS can be categorized within three categories: Vehicle Dynamics Stabilization; Information, Warning & Comfort; and Automated & Cooperative Driving [Bengler et al., 2014]. Some examples of ADAS systems are Antilock Braking System (ABS), Lane Keeping Assistant (LKA) and Adaptive Cruise Control (ACC) [Bengler et al., 2014]. Automatic Emergency Braking (AEB) is an ADAS system that gathers data from a vehicle's sensors to prevent collisions by applying the brakes. Some of these systems have the ability to prevent collision while others simply reduce the speed before collision occurs [Ariyanto et al., 2018].

LEVTEK and LEVKART

LEVTEK Sweden AB is a start-up company currently developing prototypes for an ultra-light electric vehicle with smart functions, for example ADAS. For this thesis a prototype version of the vehicle LEVKART has been used as well as an iPhone 15 Pro Max for computing and vision.

1.2 Objective

As the number of Ultra-light Electric Vehicles (LEV) increases in urban environments, their safety becomes increasingly important. Therefore the objective of this thesis is to improve the safety of LEVs by developing and implementing ADAS features, particularly an Automatic Emergency Brake (AEB). To achieve this, a kinematic model will be used to develop the AEB. The requirements of AEB will be developed through tests to determine safe and comfortable deceleration values. The feature's real-time performance will be evaluated. The project, conducted together with LEVTEK, will utilize their LEVKART as an experimental platform for all tests and implementations. Furthermore, the project will go through these stages:

- Investigate fundamental requirements such as acceleration, braking, response time and actuation dynamics for ADAS for an ultra-light electric vehicle

- Develop a kinematic model of the ultra-light electric vehicle.

- Implement, test and evaluate the selected ADAS feature, AEB, on the ultra-light electric vehicle.

Based on these steps, three research questions have been formulated:

- What are the fundamental requirements of an AEB for a LEV?
- How can the kinematics of the vehicle be described?
- How can an AEB be developed on the experimental platform?

1.3 Delimitations

The development and implementation of the AEB was limited in the situation it should function in. The tests were conducted in a controlled environment, an empty space with only a few specific target people and the vehicle was only driven straight forward.

1.4 Contributions

The work for this thesis was divided equally between the authors and most parts were done together in close collaboration. Jakob was the main person responsible for the person detection algorithm and GUI whereas Frida was the main person responsible for the control strategies and brake algorithm.

2

Vehicle Platform

This chapter highlights the important features of the platform that will be used throughout all tests. It includes the system architecture, such as sensors and computing modules, along with the kinematic model that will be used in this thesis.

2.1 Vehicle

The experimental LEVKART platform has four wheels with non-geared electric hub motors in the two front wheels and it brakes using regenerative braking in the wheel motors. The driver stands between the two pairs of wheels and steers the front wheels according to the Ackermann principle [Veneri and Massaro, 2021]. Further specifications follow in Table 2.1.

Table 2.1 Test vehicle specification

Feature	**Value**
Weight	20 kg
L_1 (Figure 2.2)	0.6 m
L_2 (Figure 2.2)	0.47 m
Wheel Diameter	165 mm
Motor Type	BLDC

2.2 System Architecture

The system architecture of the experimental platform's computing units is described in Figure 2.1 and is structured in two main units, an iPhone and a VESC system. The VESC is an off the shelf motor controller. These two units communicate via Bluetooth to transmit information from one system to the other.

The VESC system is made up of several different computing units, the VESC express and VESC 1, 2 and 3. The different VESC units communicate via a CAN bus. The CAN bus is a communication system and standard commonly used in automotive applications. In a CAN network, short messages are broadcast to the network and all nodes can listen and pick up these messages. The format and identifiers for messages are standardized and it is appropriate for real-time implementations [Corrigan, 2016]. As can be seen in Figure 2.1, all nodes have two-way communication to the CAN bus.

The VESC Express has the ability to communicate via Bluetooth and processes all information going to and from the CAN bus and the iPhone. VESC 1 and 2 control each driving wheel and they are synchronized via the CAN bus. VESC 3 handles data from the IMU which can be sent on the CAN bus and logged on the VESC Express or transmitted via Bluetooth.

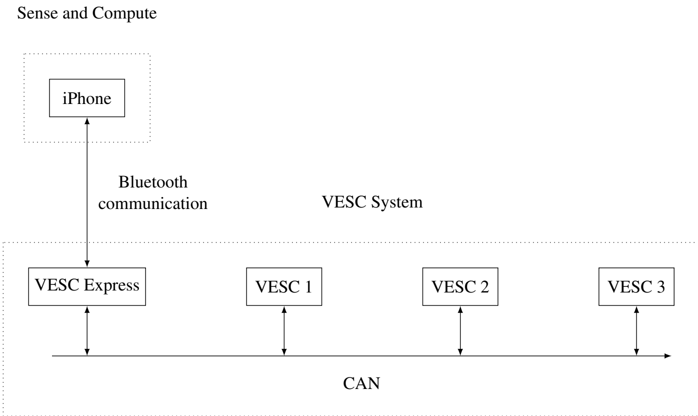


Figure 2.1 System architecture

2.3 Regenerative Braking

The prototype of the LEVKART has a regenerative braking mechanism. This mechanism allows the motors to brake the vehicle, by converting the kinetic energy into electrical energy. The recovered energy is transformed by inverting the current from the motors and simultaneously charging the battery. The braking torque of the motors is described in Equation (2.1) as an electromechanical torque from the BLDC motors. K_t is the motor constant, which is torque divided by the current of the motor [Rizzoni and Kearns, 2015].

$$T = K_t \cdot i \tag{2.1}$$

2.4 Kinematic Model

A kinematic model is a geometric description of the movement of a body without involving the forces associated with this movement[Nyberg, 2014].

The kinematics of the experimental platform can be described as the model of a car-like robot by Lynch and Park (2017) as

$$\dot{q} = \begin{bmatrix} \dot{\phi} \\ \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} (\tan\psi)/L_1 & 0 \\ \cos\phi & 0 \\ \sin\phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \tag{2.2}$$

ϕ is the heading direction, ψ is the steering angle and x and y describe the position of the midpoint between the rear wheels, these variables define the configuration of the vehicle. The controls v and w are the forward speed and the angular speed of the steering angle ψ. The length L_1 is the wheelbase in Figure 2.2. Like the LEVKART, it has two steered front wheels, two fixed-headed rear wheels and steered with Ackermann steering to reduce slipping, when turning.

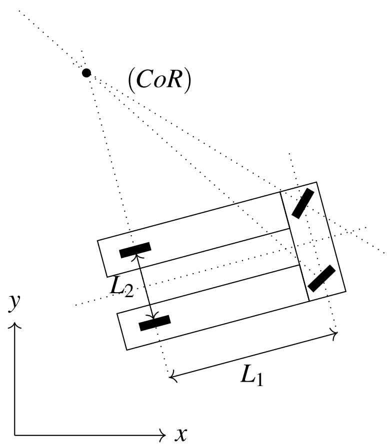


Figure 2.2 Kinematic model of the LEVKART

For emergency braking without turning, the motion is only in the forward longitudinal direction. If acceleration is assumed to be constant, the following equations describe the motion in continuous time

$$v = v_0 + at \tag{2.3}$$

$$s = v_0 t + \frac{1}{2}at^2 \tag{2.4}$$

where v, v_0, a, s and t are the current velocity, initial velocity, acceleration, distance and the time. The assumption of constant acceleration is likely to be inaccurate in most real-world applications and therefore the model should be treated as an approximation for the motion of the actual vehicle.

2.5 VESC Express Sensors

The VESC system contains an IMU unit with accelerometer, gyroscope and magnetometer. The accelerometer data is detected along three dimensions where the x-axis describes the acceleration in the forward direction of the vehicle. The VESC system can also be used to retrieve the speed of the vehicle as the revolutions per minute can be tracked and the system has knowledge of the wheel radius. All VESC system data can be both logged as well as used in real time.

2.6 Camera and LiDAR sensor

To obtain the video and depth data in the front of the vehicle an iPhone was used, where the information of the 48 MP camera and the LiDAR sensor were merged. LiDAR is an acronym for Light Detection And Ranging and emits pulses of laser to measure the distance to an object [NOAA, 2012]. The camera has a resolution of 1920 x 1440 pixels, while the depth sensor has a resolution of 320 x 240 pixels [Apple Inc., 2024b]. Fusing the sensors allow for a more comprehensice data collection, with accurate depth perception and high-resolution video.

3

Control Theory

In this chapter a brief introduction to a control strategy and a filtering method are given, along with a description of real-time requirements of the AEB.

3.1 PID Control

The most common type of controller is the PID controller. The PID controller input, u, is defined as

$$u = K\left(e + \frac{1}{T_i}\int e(t)dt + T_d\frac{de}{dt}\right) \tag{3.1}$$

where the error e is the difference between the reference signal r and the output signal y. K is the proportional gain, T_i is the integral time constant and T_d is the derivative time constant, these are the controller parameters.

The PID controller consists of three parts, P, I and D. The P part stands for "proportional", the I part for "integral" and the D part for "derivative". A controller can be designed with one, two or all three of these parts depending on the needs and characteristics of the system where it is applied. If the controller only has a proportional part it can lead to steady-state errors. This can be resolved with an integral part, which will increase if an error exists and therefore corrects this. A PI controller is an appropriate controller in most applications. A controller with a derivative part is used if there is a need to be able to predict the error, such as in slower systems [Hägglund, 2000].

The integral part is important for handling load disturbances. A large integral gain can attenuate disturbances effectively, but too large can lead to oscillatory behavior, instability and poor robustness. A larger proportional part leads to a smaller steady-state error, but can also lead to oscillatory behavior. A block diagram showing the PID controller in Figure 3.1 [Åström and Murray, 2008].

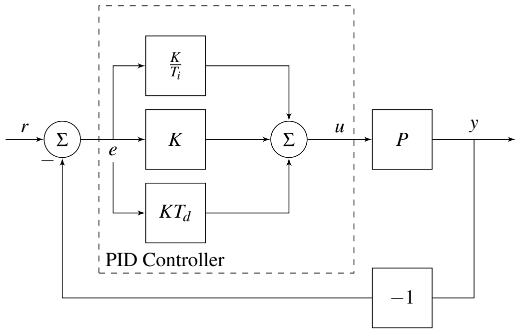


Figure 3.1 A block diagram of a simple PID controller, where P denotes the process

3.2 Kalman Filter

The performance of an ADAS system largely depends on the quality of sensor data and how it is used and a key for this is sensor fusion. The concept of sensor fusion can be described as combining sensory data from different sources such that the combination is better than information from a single source. An algorithm important for sensor fusion is the Kalman filter [Gustafsson, 2010].

A Kalman filter is an estimation filter that uses two steps to estimate the state of a system, prediction and correction. The basic version of the Kalman filter can be used when a system can be described with a linear model and the noise of the process and measurements have a Gaussian probability distribution. The Kalman filter fuses noisy sensor data and a model to compute an optimal state estimation [Åström and Murray, 2008]. If the system is non-linear it is possible to use an extended Kalman filter to propagate a Gaussian approximate state distribution [Gustafsson, 2010].

State Model

To model the vehicle's motion along a straight line, the constant acceleration translational kinematics model in discrete time is described by Gustafsson (2010) as

$$x(t+T) = \begin{bmatrix} I_n & TI_n & \frac{T^2}{2}I_n \\ 0_n & I_n & TI_n \\ 0_n & 0_n & I_n \end{bmatrix} x(t) \tag{3.2}$$

where I_n is the identity matrix, T is the sampling time and $x(t) = [p(t),\ v(t),\ a(t)]^T$. The functions $p(t)$, $v(t)$, and $a(t)$ is the position, velocity and acceleration over time. The 0_n is the zero n-dimensional vector. $x(t+T)$ can also be denoted as x_{k+1}, where $kT = t$.

Kalman Filter Equations

Consider the discrete time linear system described in Gustafsson (2010), with state transition matrix, F, control input matrix, B and observation matrix, H

$$x_{k+1} = F_k x_k + B_k u_k + v_k \tag{3.3}$$
$$y_k = H_k x_k + w_k \tag{3.4}$$

where x_{k+1} represents the state at the next time step, x_k is the current state, u_k is the controller input, y_k is the measurement, and v_k and w_k are the process and measurement noise, respectively. The Kalman filter uses the covariance Cov(v_k) = Q_k and Cov(w_k) = R_k to predict the output.

The predicted state estimate and the predicted error covariance is given by

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k \tag{3.5}$$
$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \tag{3.6}$$

The difference between the measurement data, z_k, and the predicted state, $\hat{x}_{k|k-1}$ is called the innovation residual and is given by the following relation

$$\tilde{y}_k = z_k - H_k \hat{x}_{k|k-1} \tag{3.7}$$

To simplify the equation, the innovation covariance is introduced as the sum of the prediction covariance, P_k, and the measurement covariance, R_k [Gustafsson, 2010]. The innovation covariance is given by

$$S_k = H_k P_{k|k-1} H_k^T + R_k \tag{3.8}$$

The optimal Kalman gain, K, represents the proportion of prediction covariance to innovation covariance, as the following equation states

$$K_k = P_{k|k-1} H_k^T (S_k)^{-1} \tag{3.9}$$

The corrected state estimate will then be the predicted state, which will be amended by the Kalman gain multiplied with the innovation residual, $\tilde{y}_k$,

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k \tag{3.10}$$

At last the estimated covariance matrix will be corrected,

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \tag{3.11}$$

3.3 State Machine

A finite state machine, later called state machine (SM), is a mathematical model to represent steps in a process or in a computation. The SM consists of states and transitions, also known as nodes and edges, and a SM can only be in one of the states at a time. There is always an initial state and to change state a specific input has to be triggered, see Figure 3.2 [Årzén, 2024]. The initial state will always be S_0 and instantly when condition a is triggered the state will change to S_1. The state machine will be in this state until b is true, see Figure 3.2.

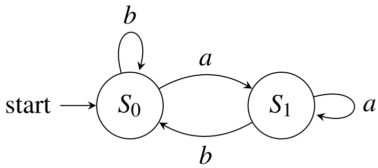


Figure 3.2 Example of a state machine, where a and b are conditions that trigger a transition

3.4 Real-Time Requirements of AEB

AEB relies on precise timing constraints to prevent collision. Hence, it is crucial to determine the real-time requirements of the LEVKART to facilitate the implementation of an AEB system. Therefore the worst case response times needs to be decided [Årzén, 2024]. In Figure 3.3 the process is described and each of these steps will be considered in the real-time analysis.

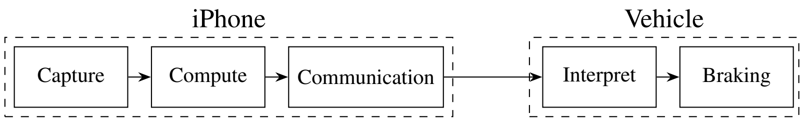


Figure 3.3 Block diagram for AEB feature

Capture and Compute

The real-time requirements are influenced not only by physical factors but also by computing and messaging times, i.e., formulating the correct information to the motor controllers. In the capture phase the object that is to be avoided by the AEB is detected by a camera. The camera is set to 30 frames per second and have the ability to detect objects at the same rate. In this project, the LEVKART is using the camera and A17 Pro chip from the iPhone 15 Pro Max to compute and detect objects using the CoreML framework. There are four modes to compute with the CoreML and they are [Apple Inc., 2024c]:

- `all` - specifying the model's execution on the Apple Neural Engine, CPU, or GPU, based on their respective capabilities. This is the default mode.

- `cpuOnly` - denoting the model running exclusively on the CPU of the device.
- `cpuAndGPU` - indicating the model's operation primarily on the GPU with occasional CPU usage.
- `cpuAndNeuralEngine` - using the CPU and the ANE.

The average inference times of the iPhone according to Denis (2023) are presented in Table 3.1. The object recognition model is creating requests to Apple's CoreML

Table 3.1 Average inference time [ms]

Configuration	A16 Bionic	A17 Pro
CPU	331	302
GPU + CPU	208	146
ANE + GPU + CPU	44	37

framework to perform image analysis [Wallner and Bérenger, 2023].

4

Fundamental Requirements

The fundamental requirements for an ADAS feature are important to understand in the initial phase of the development. What is a fundamental requirement depends on the specific feature, which in this case is an automatic emergency brake. The requirements that will be investigated for the AEB are how velocity and acceleration are related to the current input to the electric brakes as well as comfort for the driver in different braking situations. This was done through braking tests.

4.1 Braking Tests

To investigate the braking patterns of the LEVKART, tests were set up where a driver stands on the vehicle the is kept at a certain speed. When the test velocity is reached, the brake is initialized by an observer via the Bluetooth-connected iPhone. The reason for this is both for standardization reasons as well as to keep the driver as unprepared as possible, simulating a real emergency brake scenario. The braking current is set to a certain ratio of the maximum and should brake at a constant deceleration. The tests were conducted with all combinations of the settings described in Table 4.1, resulting in 42 tests per driver.

Table 4.1 Test parameters, velocities and deceleration ratios together with their corresponding current value

Speed [km/h]	Deceleration ratio	Corresponding current [A]
5	0.1	4.5
7.5	0.2	9
10	0.4	18
12.5	0.6	27
15	0.8	36
17.5	1.0	45
20		

Figure 4.1 describes the test scenario. The vehicle is kept at a constant velocity, then at the braking point a constant deceleration value is set and the vehicle slows down until it comes to a full stop. The distance from the braking point to the stop point is measured using a camera, and to calibrate the camera a measuring tape marking 1 m is used.

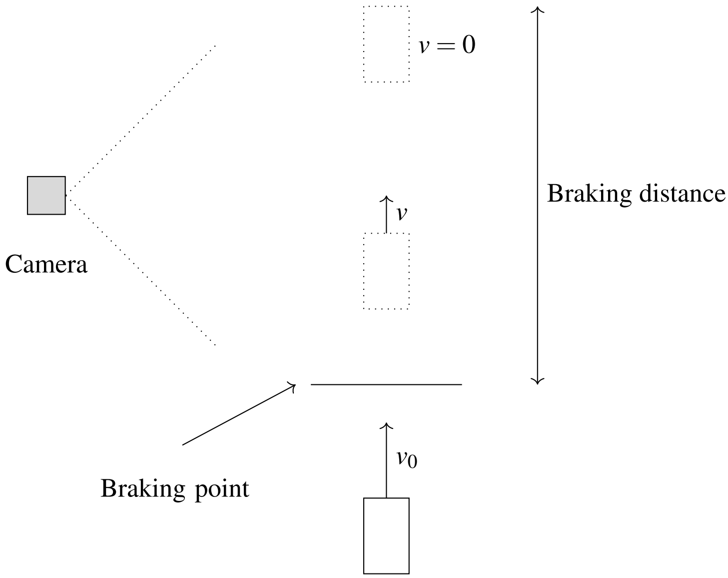


Figure 4.1 Braking test

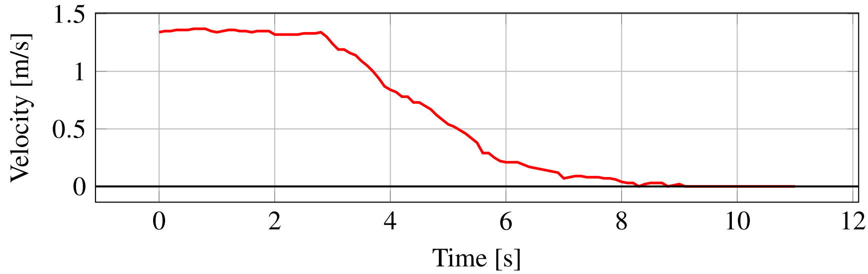
The acceleration, velocity and braking distance are measured for all tests as well as the current from the electric motors. Acceleration, velocity and current is measured from the vehicle with the VESC control unit.

4.2 Resulting Fundamental Requirements

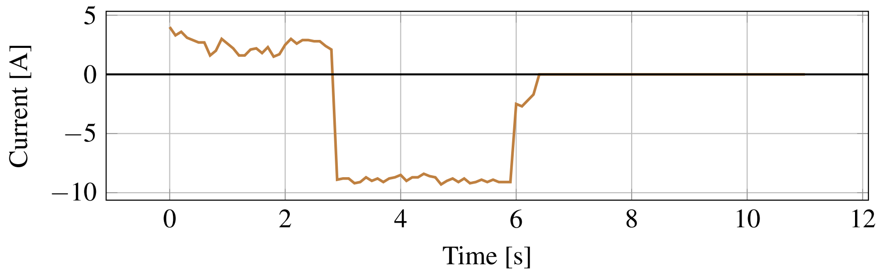
The main goal for the braking tests was to understand the relationship between acceleration and brake current as well as understand how comfort level is related to the braking process.

Data Analysis

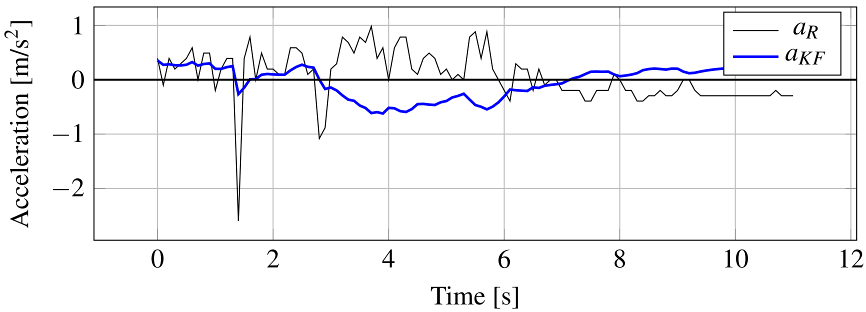
The velocity and current data corresponded to the test scenario, but the acceleration data was very noisy. Alfian et al. (2021) describe this as a common problem for accelerometers and conclude that a Kalman filter could be used to filter out noise. A Kalman filter for motion is described in Section 3.2 and an example of data for a brake test scenario is seen in Figure 4.2. In Figure 4.2, the current is negative as the velocity decreases, which is to be expected. The raw acceleration data a_R in Figure 4.2c is very noisy and hard to interpret, but the Kalman filtered acceleration data a_{KF} corresponds well with the velocity data. The Kalman filter does not only consider the raw acceleration data but also the model of the one-dimensional motion in Equation (3.2). The filtered data corresponds better with the velocity and current data and is therefore considered more reliable.



(a) The speed during the braking process



(b) The current from the motors



(c) The measured acceleration, a_R and the Kalman filtered acceleration, a_{KF}

Figure 4.2 Data from braking tests

To understand the relation between acceleration and brake current while braking, the average filtered acceleration of each brake cycle was extracted, when the current to the electric motors is negative. This value can be compared to the average brake current value at each test scenario to find the relation.

Results

The relationship between brake acceleration and current into one of the motors was measured and the result can be seen in Figures 4.3 and 4.4 and this relation follows a linear pattern. This corresponds to Equation (2.1), in which the electric motor current is proportional to the torque. Figures 4.3 and 4.4 represent this relation for two different drivers where the weight is the main difference. As this relation will be dependant of the weight of the driver, vehicle and load, an approximation of the relation was made based on the average weight of a person. The average weight of an adult in Sweden is 76 kg according to Statistiska Centralbyrån (2018), based on this the linear relations of the two drivers were interpolated and the average linear relation is given by Equation (4.1).

$$a = 0.0576i + 0.1559 \tag{4.1}$$

where a is the acceleration and i is the braking current.

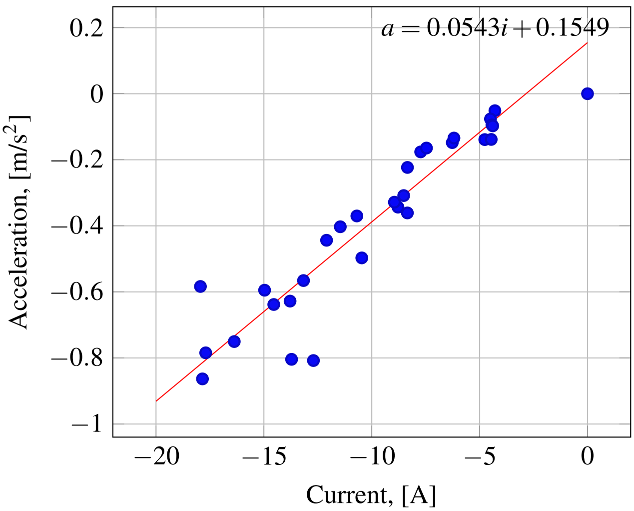


Figure 4.3 Acceleration and current relation Driver 1

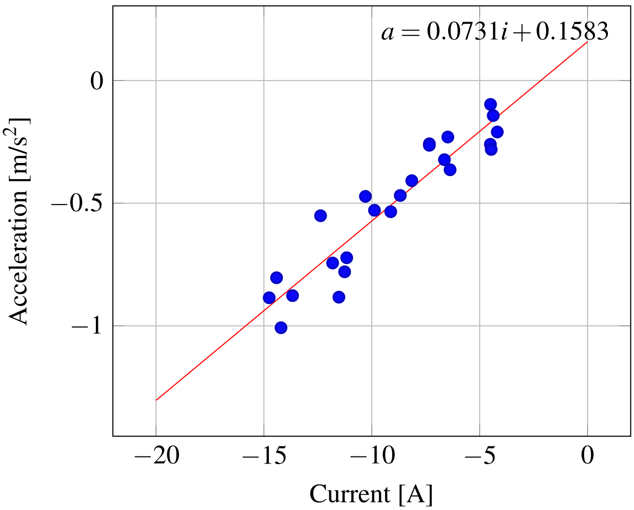


Figure 4.4 Acceleration Current relation Driver 2

During the test, a subjective measure of comfort was taken as well and the relation between the maximum acceleration value and comfort level during different brake currents and speeds can be seen in Figure 4.5. The comfort levels are denoted in a scale from 1 to 5, where 1 is very uncomfortable and 5 is very comfortable. As all accelerations rated as comfortable (4 or 5) had a maximum negative acceleration under $-1\ \mathrm{m/s^2}$, this was considered to be the maximum deceleration value for a comfortable brake.

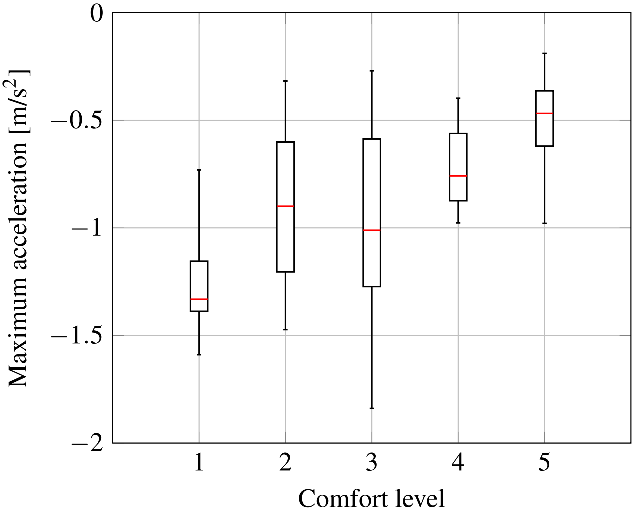


Figure 4.5 Comfort results from all tests

5

Vision and LiDAR

A fundamental part of the AEB functionality is sensor input and how it is processed to recognize a situation where emergency braking is necessary. This includes processing camera and LiDAR input and merging the sensor data, which enables both object detection and distance measurements.

5.1 Vision and Person Detection

The Vision framework provided by Apple uses machine learning to detect objects and humans. It has the capability to detect 19 unique body points according to Apple Inc. (2024a), see Table 5.1. The frame from the stream is analyzed and processed by the Vision framework's class VNDetectHumanBodyPoseRequest and if there is a human in sight it will return an observation with all the joints.

Table 5.1 The joints that VNDetectHumanBodyPoseRequest detects. **Both left and right*

Part	Joint
Head	Nose
	Ear*
	Eye*
Upper body	Neck
	Shoulder*
	Elbow*
	Wrist*
	Hand*
Lower Body	Hip*
	Knee*
	Ankle*
	Foot*

The framework can recognize several humans in each frame and all of them will be appended to an array consisting of VNHumanBodyPoseObservation objects. This array is later processed and analyzed to retrieve the bounding box and the mean x value of all the joints in the observation. If there are multiple persons in sight, they are compared against each other to determine which is along the vehicle's path and at the closest distance. These two conditions will be described in Sections 5.2 and 5.3.

5.2 Distance Measurements

To ensure an accurate distance measurement to the detected object, the depth data from the LiDAR sensor is fused together with the pixel buffer from the camera, so that the timestamps are equivalent. A depth map is a representation of a scene captured by a depth sensor, in this case the LiDAR sensor. The depth map consists of a pixel buffer where each pixel contains a depth value that specifies the distance from the sensor to the corresponding point in the scene. Essentially, the depth map provides a spatial representation about the objects in the scene, allowing the system to retrieve an estimation of distances and therefore determine the space between itself and the person in view. In Figure 5.1 the same scene is captured with both the camera and the LiDAR sensor. To the right the depth map is converted to a regular picture with a gradient showing the distances to different objects.

Figure 5.1 Camera and LiDAR representation of a scene.

When a person is detected with the Vision framework, see Section 5.1, a bounding box around the person is returned. This box is then passed to an algorithm that computes the distance to the person of interest. First the algorithm extracts the center coordinates of the bounding box, which is translated to the corresponding coordinates of the depth map. As a safety precaution, and to ensure an accurate value, the distance is not solely measured to the center, instead a 20×20 pixels square surrounding the center of the bounding box is taken into consideration. The distance to the person is then established by determining the closest distance from each pixel in the 20 by 20 square, see Figure 5.2.

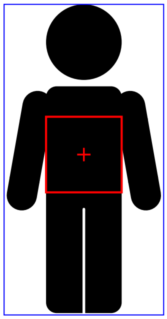


Figure 5.2 Bounding box of the person detection algorithm, blue, and its center point, red

5.3 Path

One crucial condition to acknowledge when designing the AEB for the LEVKART is to confirm if there is a person standing in the vehicle's path. This check will avoid unwanted behavior such as braking for a person in the peripheral view of the camera. The procedure is based on geometry, utilizing the vehicle's width, the distance to the person, the mean x-value of the joints and the camera's field of view, see Figure 5.3.

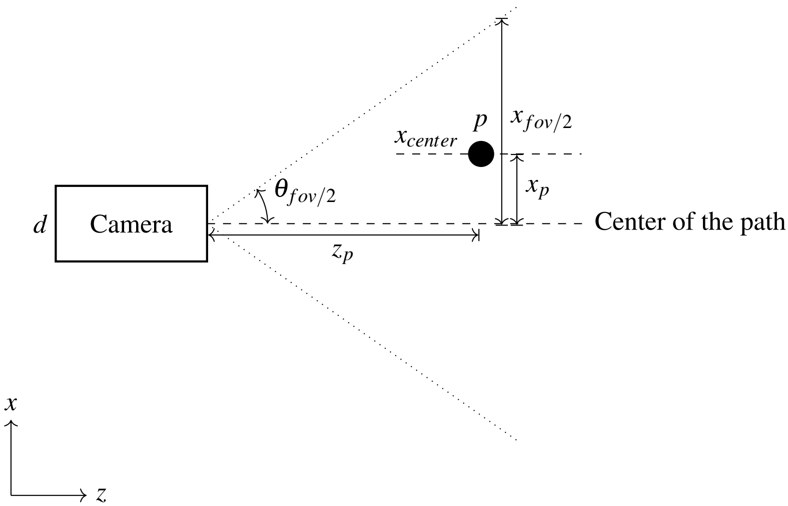


Figure 5.3 An illustration of the geometry behind if the person is obstructing the vehicle's path

When the person detection algorithm identifies a person within the field of view, it provides two key pieces of information: the bounding box outlining the person's location and the mean x-value of the joints, which represents the center of the person, see Figure 5.4. This x-value enables the system to determine whether the person is obstructing the vehicle's path.

Initially, the distance was measured from the centerline to the edge of the field of view using the equation:

$$x_{fov/2} = z_p \cdot \tan\ \theta_{fov/2} \tag{5.1}$$

Here z_p represents the distance to the person in meters, $x_{fov/2}$ signifies the distance between the center of the image and the edge of the field of view, and $\theta_{fov/2}$ denotes half of the field of view (FOV) angle of the camera, which, in this context, is 36.0°, see Figure 5.3.

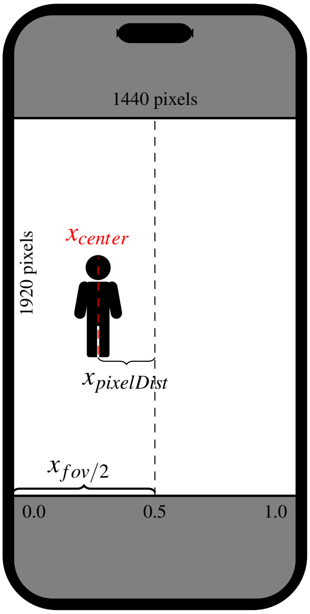


Figure 5.4 This illustration showcases the AEB view, featuring key components for the person detection algorithm and path analysis.

Furthermore, the mean x-value from the joints was utilized to detect the person's position on the screen using:

$$x_{pixelDist} = |0.5 - x_{center}| \cdot 1440 \tag{5.2}$$

In this equation, $x_{pixelDist}$ and x_{center} represent the normalized distance from the center measured in pixels and the mean x-value for the joints, see Figure 5.4. The absolute value is then multiplied by the camera's width resolution, which is 1440 pixels in this case.

To determine the meter-per-pixel value necessary for computing the person's position, the result from Equation (5.1) was divided by half of the resolution along the x-direction:

$$x_{mpp} = \frac{x_{fov/2}}{720} \tag{5.3}$$

Here, x_{mpp} denotes the distance in meters per pixel. Finally the physical distance of the person can be derived by combining Equation (5.2) and (5.3),

$$x_p = x_{pixelDist} \cdot x_{mpp} \tag{5.4}$$

If the x_p from Equation (5.4) is less than the width of the path, which is the vehicle width, d, with a safety margin of 0.1 m on each side, the person is obstructing the path, otherwise they are not, see Figure 5.3.

6

Braking Strategy

This chapter will present the implemented solutions and the overall strategy when designing an AEB algorithm.

6.1 Braking Algorithm

To implement the AEB, an algorithm was developed to identify the situation, activate the braking function and execute it. The logic of this is described in the state machine in Figure 6.1.

State Machine

There are four nodes in the state machine in Figure 6.1 where the AEB function can be in.

- S_0: Initial state, vehicle should function normally.
- S_{PD}: Person detection state, person detected but emergency braking conditions are not met. Vehicle should function normally.
- S_B: Brake state, active emergency braking.
- S_{NA}: Non acceleration state, vehicle should not accelerate in final stage braking. Handles the case where the vehicle is close enough to a person that the person detection algorithm can no longer identify them.

The conditions for the state machine involve four main components, three of which are given by the algorithms described in Chapter 5. *Person* is a boolean variable which is true if a person has been detected by the person detection algorithm. *Path* is a boolean variable that signifies whether the detected person is within the designated path area. d_{actual} is the value of the distance from vehicle to person. The last component, v, is the speed of the vehicle. The notation in Figure 6.1 is based on mathematical logical notation where $\neg$, $\wedge$ and $\vee$ represents *not*, *and* and *or*.

The S_B state is not entered until the distance to the person is less than 5 m. This limit was set as it is not desirable to brake for persons that are too far away to be important for the driving scenario, but it should be low enough so that the velocity can be decreased at a safe rate. This value could be altered depending on the environment where the vehicle would be used.

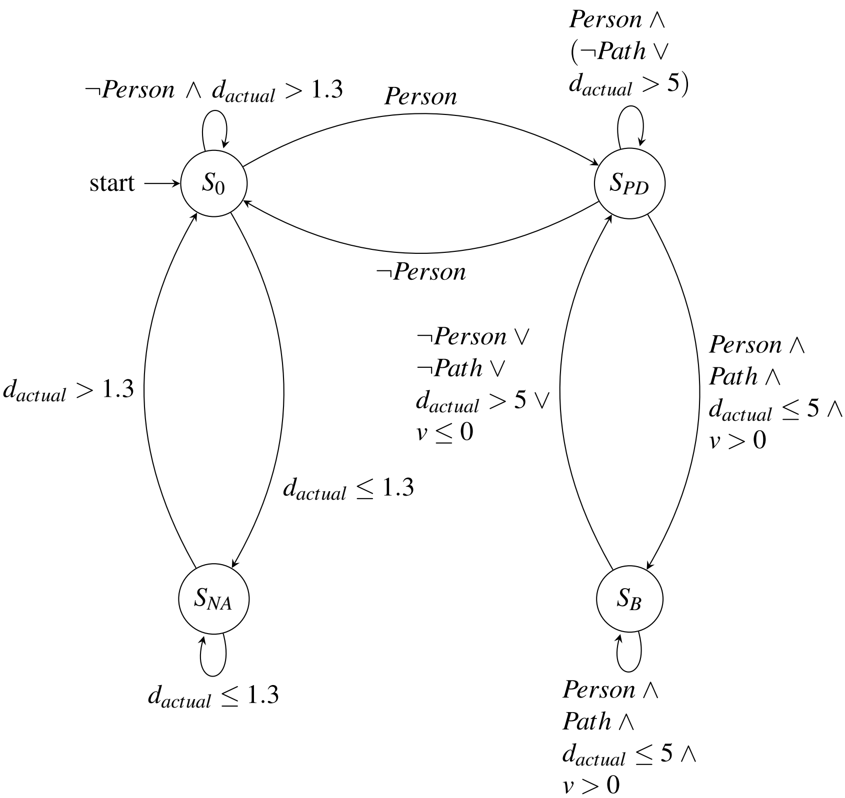


Figure 6.1 State machine of the AEB algorithm

Active Braking

As the vehicle enters the state S_B, it will activate the AEB to bring the vehicle to a stop before a crash with the detected person. To ensure this, a safety distance, d_{safety}, is set of 1.5 m where the vehicle will aim to have come to a complete stop. The distance in all following computations in the control strategies will therefore be based on the resulting distance, d, computed as

$$d = d_{actual} - d_{safety} \tag{6.1}$$

where d_{actual} is the measured distance to the person. The control strategies discussed in Section 6.2 determines a current to set for the electric motors. This current is sent continuously over Bluetooth to the CAN bus on the vehicle and is updated at a frequency of 30 Hz.

VESC

The actuation of the brake command is executed on the VESC controller where the last decision making takes place. It is important that the right input and command is prioritized, so that the driver feels safe. Input to the VESC can come from the manual two-way lever, where the driver controls acceleration and braking, and via Bluetooth from the iPhone. When the AEB is active, the VESC control unit constantly receives commands via Bluetooth, but commands can come in parallel from the lever .It is important that a command hierarchy is established to ensure the driver's safety. The hierarchy is shown in Table 6.1 where brake commands are prioritized over speed input, but manual braking commands from the driver are prioritized over the AEB commands.

Table 6.1 Hierarchy on the VESC controller with priority from 1 to 4, where 1 is the highest priority

Command	**Priority**
Brake from lever	1
Brake from iPhone	2
Speed input from lever	3
Other inputs from iPhone	4

Close Up Obstacle Detection

The S_{NA} state will activate a close-up obstacle detection algorithm in the case that a person is no longer detected by the person detection algorithm, but the LiDAR has a distance to something in front of the vehicle at a distance of less than 1.3 m. The distance of 1.3 m was chosen since it was at this distance the person detection algorithm started to fail recognizing a person. The LiDAR depth map is analyzed for a square of 30×30 pixels in the center of the camera and if any distance is less than 1.3 m, maximum braking current is applied to the motors regardless of the object. This ensures that the vehicle will not crash into a person if the person is too close for the person detection algorithm to detect them and the manual gas is applied. This state will not be left until the LiDAR distance is over 1.3 m again, indicating that the person has stepped aside.

6.2 Control Strategies

The AEB function is safety critical, which means it should both be able to come to a full stop before collision, as well as be comfortable and non-surprising. To be able to achieve this, the control strategy is very important. For this reason, three control strategies are implemented and will be evaluated through tests, strategy A, B and C. A new control loop is started each time the S_B state is entered in Figure 6.1.

Strategy A: Simple Distance Feedback

The first control strategy, which will be referred to as strategy A, is a simple model based on calculating the constant acceleration a needed to reach a velocity of zero at a given distance d and starting velocity v.

$$a = \frac{-v^2}{2 \cdot d} \tag{6.2}$$

The acceleration, a is updated at 30 Hz, at each new frame from the camera. As it corrects itself during the process as the distance and velocity changes it can be considered to be closed-loop control.

Strategy B: PI Control with Velocity

A PI control strategy was developed based on keeping the velocity at the right value, given by a certain profile and the strategy will be known as strategy B. As the decrease in velocity should be gradual and adjusted to the situation, a velocity decrease profile is set from the initial speed and the distance, and should be linear according to,

$$a_{\text{target}} = \frac{-v_{\text{start}}^2}{2 \cdot d_{\text{start}}} \tag{6.3}$$

$$v_{\text{ideal}} = a_{\text{target}} \cdot t + v_{\text{start}} \tag{6.4}$$

The velocity decrease has a constant acceleration value given by Equation (6.3). When a control signal is to be computed, the a_{target} from Equation (6.3) is used to find the ideal speed v_{ideal} at the current time, according to Equation (6.4). The error for the PI controller results in the difference between the current speed and this ideal speed. To enable the controller to handle moving people, the target acceleration is updated every tenth calculation (3 Hz) of the control signal with the current velocity and distance to person as v_{start} and d_{start}.

The velocity inputs are taken from the speed measurements from the VESC controller on the vehicle.

Strategy C: PI Control with Acceleration

The PI controller with acceleration will be referred to as strategy C and it is very similar to the PI controller with velocity. It uses Equation (6.3) to determine a constant ideal acceleration and the control error is set as the difference between the ideal acceleration and the current acceleration. The ideal acceleration value is updated every tenth calculation (3 Hz), in the same way as the PI controller with velocity.

The acceleration inputs come from the IMU on the iPhone attached to the vehicle. Initially the IMU data was Kalman filtered but the filtered data was not an improvement and therefore it was decided to only use the raw IMU acceleration data.

6.3 Control Tuning

For the PI control strategies to function and be as comfortable as possible, tuning of the parameters was needed. As there are two PI control strategies based on different inputs, strategy B and C, they each need different parameters. The parameters are, as can be seen in Equation (3.1) K, the proportional parameter, and K/T_i, the integral parameter. This was done with a trial-and-error method by first finding a proportional parameter that was powerful and fast enough while not leading to oscillatory behavior. After that, an appropriate integral parameter was found to make the control more accurate and smooth, while not creating oscillating behavior. The final parameters are described in Table 6.2 for the PI controllers. Other methods could be applied to ensure better tuning, however trial-and-error was sufficient within the scope of the thesis.

Table 6.2 Parameters for the different PI controllers

Parameter	Strategy B	Strategy C
K	15.0	1.5
K/T_i	2.0	0.75

7

Experiments

In this chapter the results of the different braking strategies from Chapter 6 are presented and compared. Test results from the real-time testing and close-up object detection testing are also presented.

7.1 Braking Test Results

The purpose of the braking tests were to evaluate the performance of the three different control strategies for the braking. The test methods involved reaching a predetermined speed, driving towards a person, and allowing the AEB to analyze the situation and engage the brakes when it was necessary. Each brake strategy was tested starting from 5 km/h, continuing until the algorithm was no longer considered safe.

The results from the tests are presented in Tables 7.1 and 7.2. The three different brake strategies are:

- Strategy A: Simple distance feedback
- Strategy B: PI control with velocity
- Strategy C: PI control with acceleration

The "distance" column in the tables indicates the remaining distance after the vehicle came to a complete stop. In certain tests, where the final distance is 0.0 m, the driver intervened by placing one foot on the ground to prevent a collision with the person, who acted as an obstacle. This was annotated with keywords like "Safe" and "Unsafe", with "Safe" denoting a velocity low enough to enable easy vehicle stoppage by placing a foot down, while "Unsafe" indicated the need for a much more forceful manual deceleration of the vehicle.

Both of the results from Driver 1 and Driver 2 clearly demonstrate that strategy B yields the highest comfort level performance. The results indicates that the performance of the braking algorithm performs better at low speeds than it does at higher speeds.

Table 7.1 Braking test results for Driver 1

Brake type	Velocity [km/h]	Distance [m]	Comfort	Comment
Strategy A	5	1.5	5	
	6	1.4	5	
	7	1.3	4	
	8	0.9	4	Figure 7.1
	9	0.0	4	Safe
	10	0.0	4	Safe, Figure 7.2
	11	0.0	3	Safe
Strategy B	5	0.8	5	
	6	0.7	5	
	7	0.2	5	
	8	0.0	5	Safe, Figure 7.5
	9	0.0	5	Safe
	10	0.0	5	Unsafe
Strategy C	5	1.5	5	
	6	1.5	5	
	7	1.1	4	
	8	0.5	4	Figure 7.7
	9	0.0	4	Unsafe
	10	0.0	4	Figure 7.8

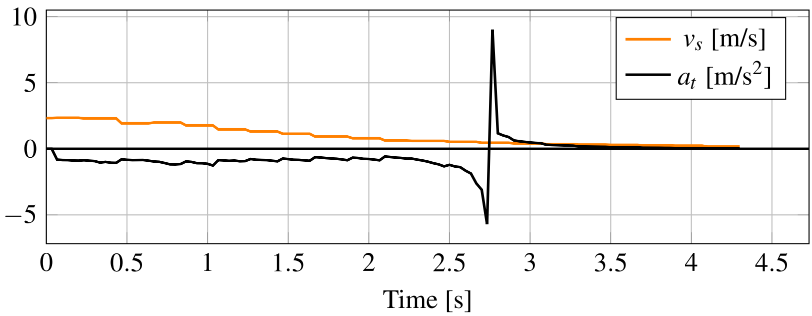
There are four tests, two for Driver 1 and two for Driver 2, that are considered "Unsafe" and the remaining are "Safe".

Table 7.2 Braking test results for Driver 2

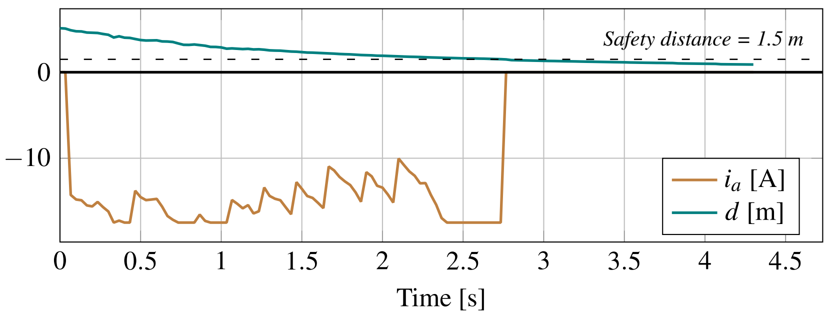
PI type	Velocity [km/h]	Distance [m]	Comfort	Comment
Strategy A	5	1.9	5	Figure 7.3
	6	1.8	5	
	7	1.8	5	
	8	1.7	4	
	9	0.7	4	
	10	0.0	3	Safe, Figure 7.4
	11	0.0	3	Safe
Strategy B	5	1.6	5	Figure 7.6
	6	0.9	5	
	7	0.7	5	
	8	0.0	5	Safe
	9	0.0	5	Safe
	10	0.0	5	Safe
	11	0.0	4	Safe
Strategy C	5	0	4	Safe, Figure 7.9
	6	1.5	3	
	7	1.3	3	
	8	0.0	2	Unsafe
	9	0.0	2	Unsafe
	10	0.5	4	Figure 7.10

Strategy A

Data from selected braking tests of strategy A is presented in Figures 7.1, 7.2, 7.3 and 7.4. For all tests, the distance and velocity decrease gradually as intended. An aspect to note is that in Figure 7.1 as well as Figures 7.5 and 7.9 the target acceleration decreases rapidly until it changes sign and becomes a large positive integer. This happens because of the fact that the controllers calculate the target acceleration, based on the distance to the obstructing person. The safety distance is subtracted to the actual distance and as a result of this the target acceleration, Equation (6.3), will be negative and therefore result in a positive target acceleration.

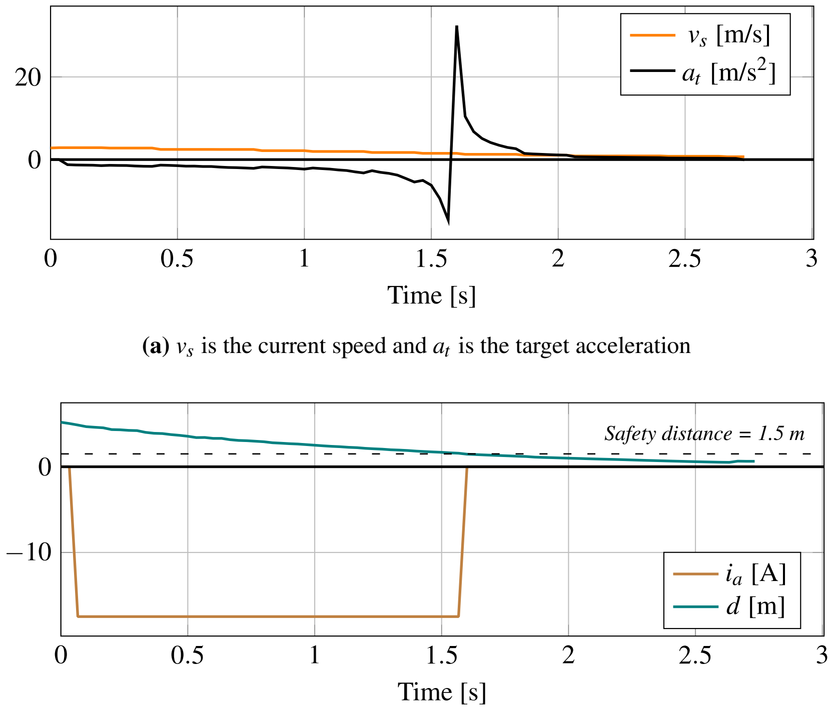


(a) v_s is the current speed and a_t is the target acceleration



(b) i_a is the braking current sent to the motor and d represents the distance to the object. The dashed line at 1.5 m depicts the safety distance to the person

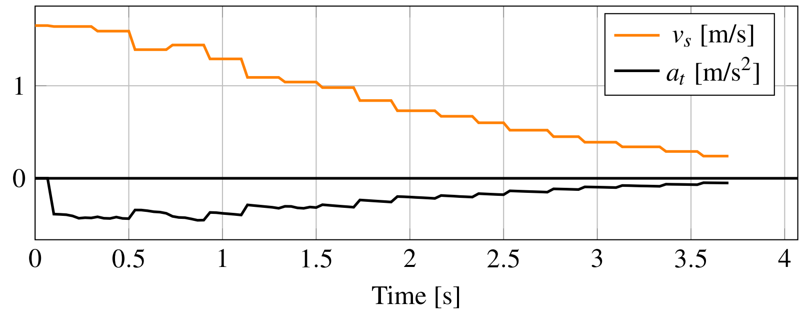
Figure 7.1 Braking test conducted with Driver 1 driving at 8 km/h



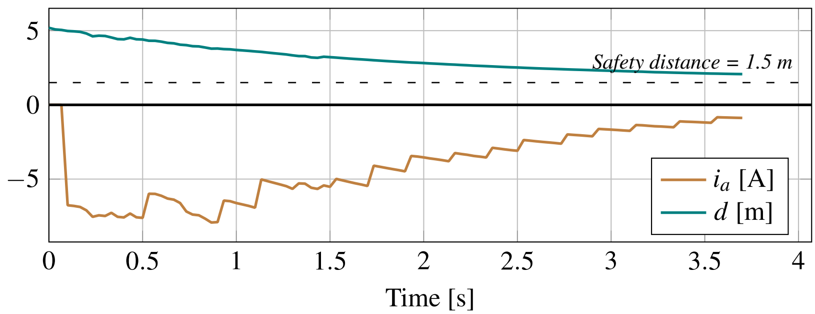
(a) v_s is the current speed and a_t is the target acceleration

(b) i_a is the braking current sent to the motor and d represents the distance to the object. The dashed line at 1.5 m depicts the safety distance to the person

Figure 7.2 Braking test conducted with Driver 1 driving at 10 km/h

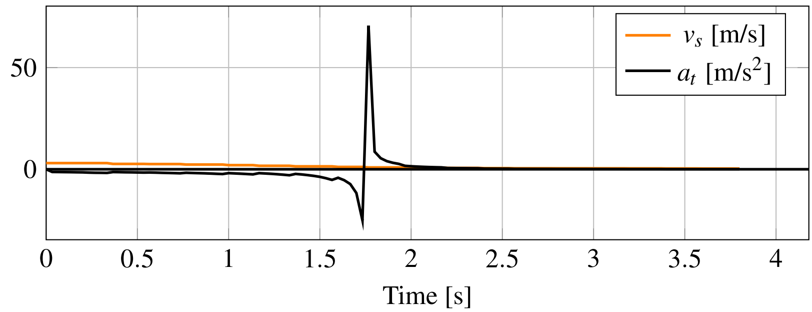


(a) v_s is the current speed and a_t is the target acceleration

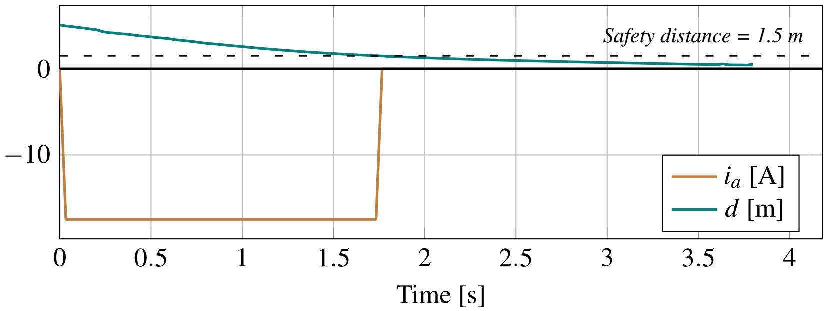


(b) i_a is the braking current sent to the motor and d represents the distance to the object. The dashed line at 1.5 m depicts the safety distance to the person

Figure 7.3 Braking test conducted with Driver 2 driving at 5 km/h



(a) v_s is the current speed and a_t is the target acceleration

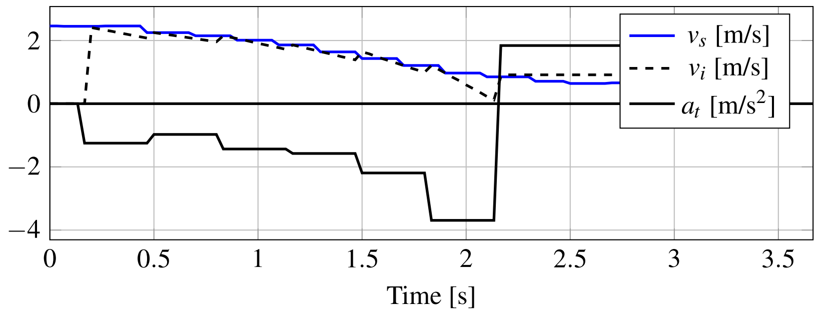


(b) i_a is the braking current sent to the motor and d represents the distance to the object. The dashed line at 1.5 m depicts the safety distance to the person

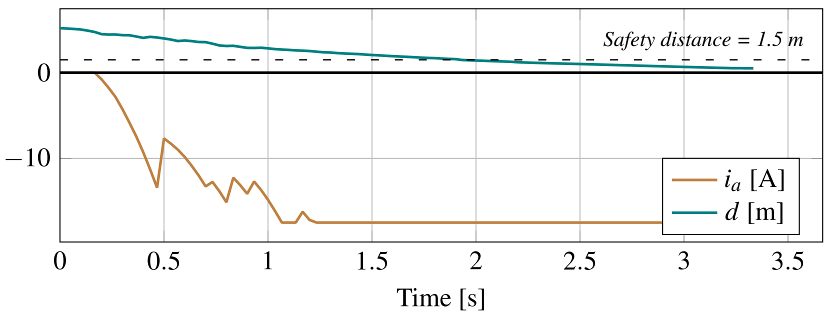
Figure 7.4 Braking test conducted with Driver 2 driving at 10 km/h

Strategy B

Data from selected braking tests for strategy B is presented in Figure 7.5 and 7.6. The decrease in velocity follows the ideal speed profile and the distance decreases gradually.

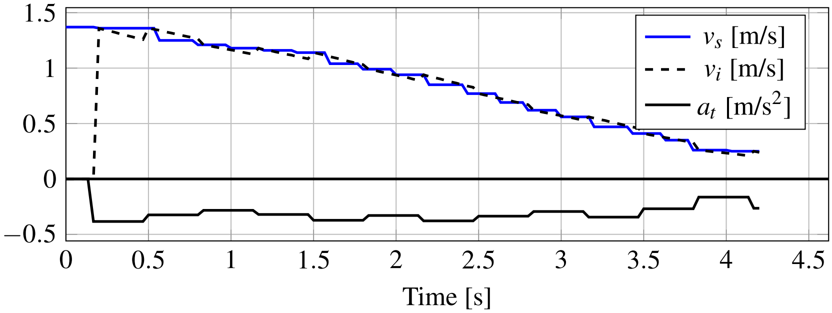


(a) v_s is the current speed, v_i is the ideal speed and a_t is the target acceleration.

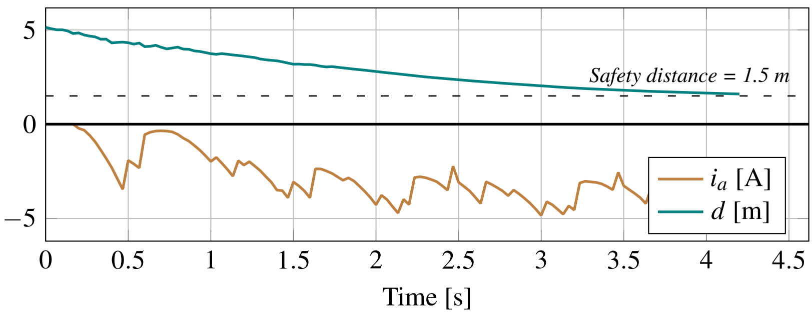


(b) i_a is the braking current sent to the motor and d represents the distance to the object. The dashed line at 1.5 m depicts the safety distance to the person

Figure 7.5 Braking test conducted with Driver 1 driving at 8 km/h



(a) v_s is the current speed, v_i is the ideal speed and a_t is the target acceleration.

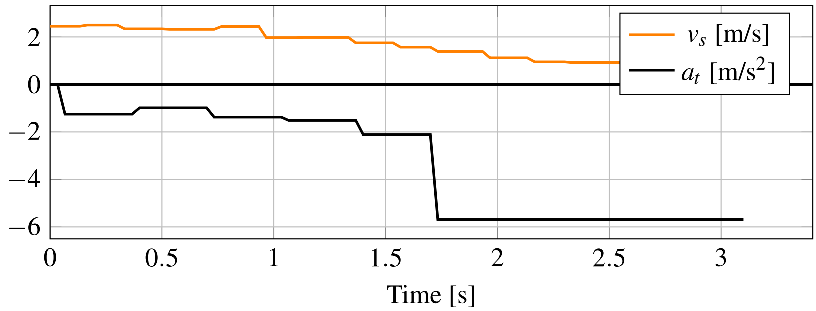


(b) i_a is the braking current sent to the motor and d represents the distance to the object. The dashed line at 1.5 m depicts the safety distance to the person

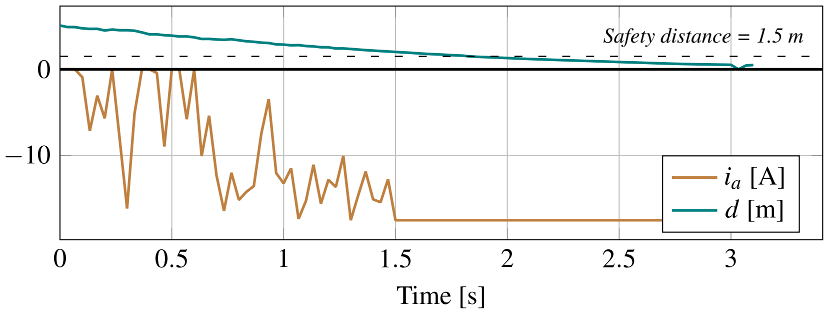
Figure 7.6 Braking test conducted with Driver 2 driving at 5 km/h

Strategy C

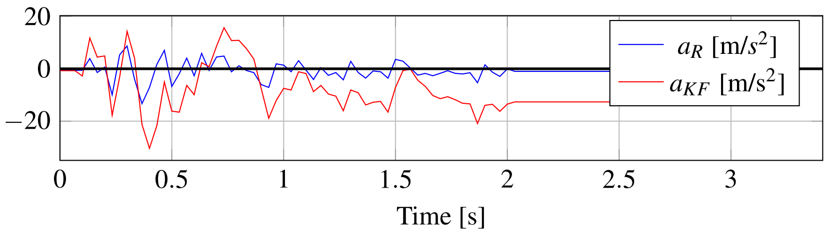
Data from selected braking tests for strategy C is presented in Figures 7.7, 7.8, 7.9 and 7.10. In Figures 7.7c and 7.9c the data from the accelerometer is shown together with the Kalman filtered acceleration data. It can be seen that the Kalman filtered data does not limit the accelerometer noise as the filter increases the error.



(a) v_s is the current speed and a_t is the target acceleration

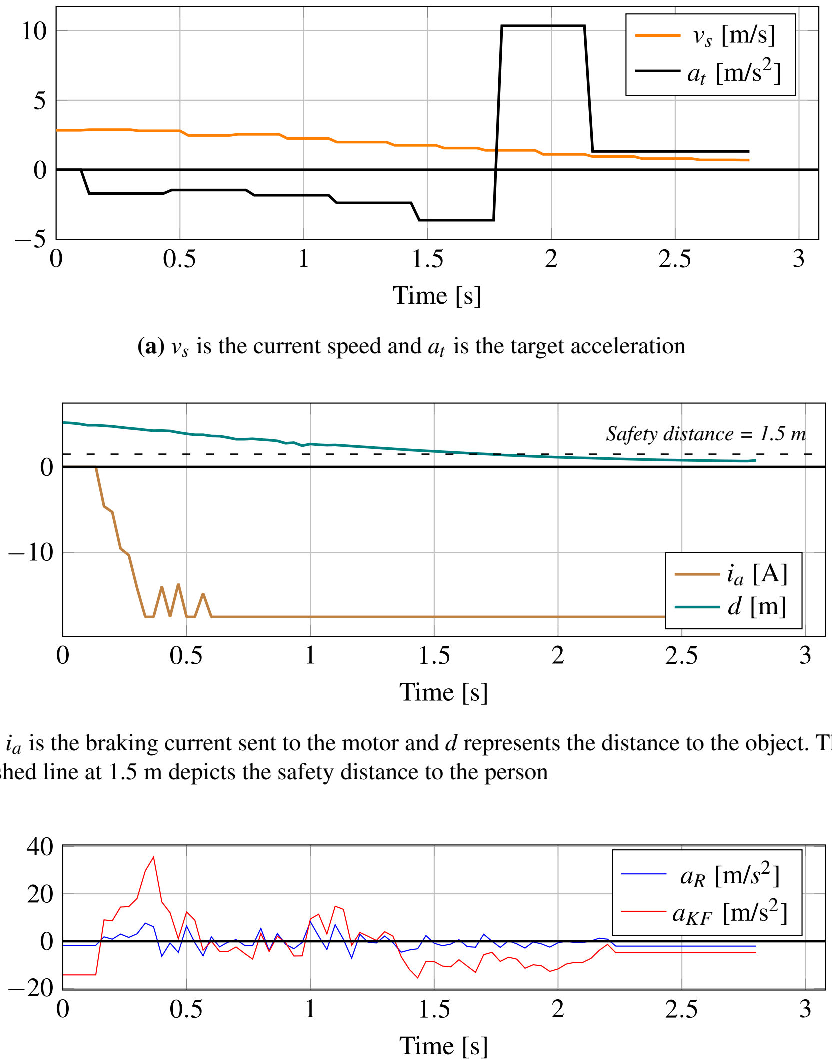


(b) i_a is the braking current sent to the motor and d represents the distance to the object. The dashed line at 1.5 m depicts the safety distance to the person



(c) a_R is the measured acceleration data and a_{KF} is the Kalman filtered acceleration data

Figure 7.7 Braking test conducted with Driver 1 driving at 8 km/h

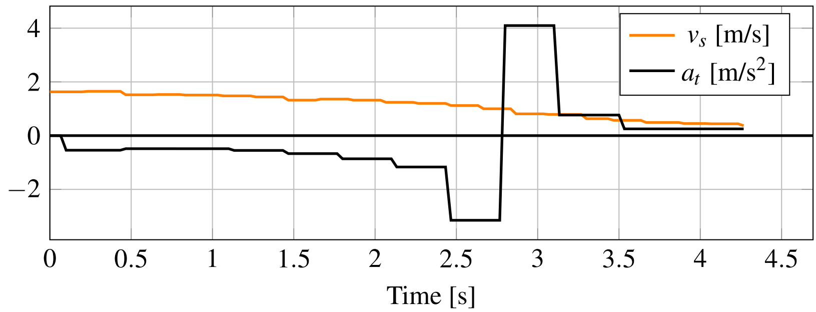


(a) v_s is the current speed and a_t is the target acceleration

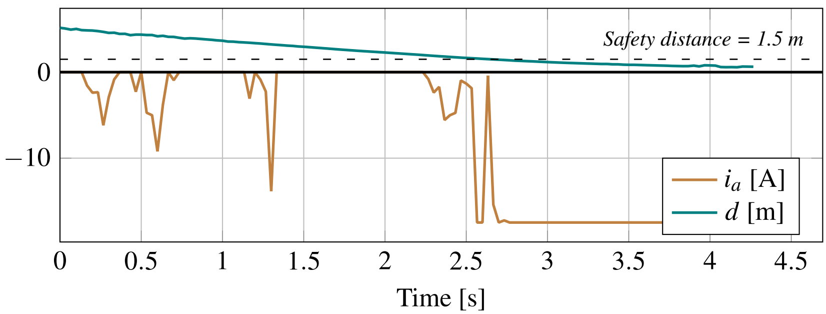
(b) i_a is the braking current sent to the motor and d represents the distance to the object. The dashed line at 1.5 m depicts the safety distance to the person

(c) a_R is the measured acceleration data and a_{KF} is the Kalman filtered acceleration data

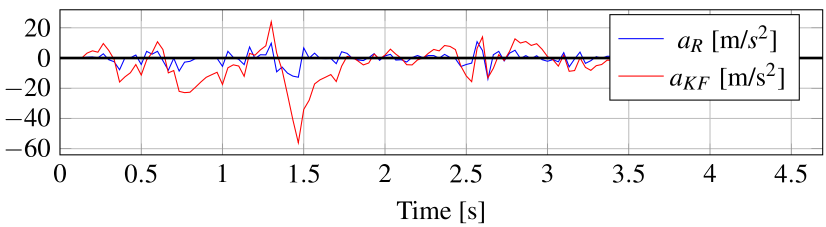
Figure 7.8 Braking test conducted with Driver 1 driving at 10 km/h



(a) v_s is the current speed and a_t is the target acceleration

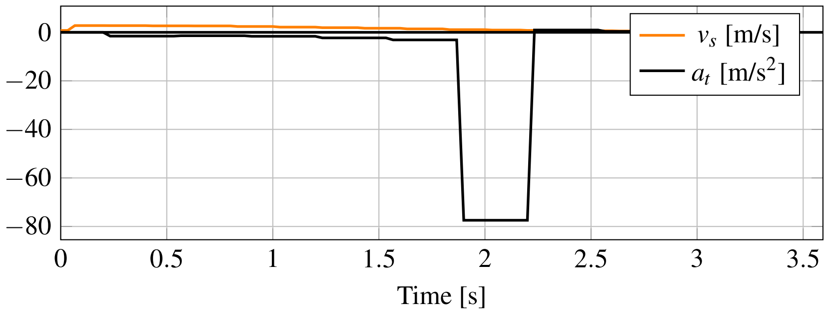


(b) i_a is the braking current sent to the motor and d represents the distance to the object. The dashed line at 1.5 m depicts the safety distance to the person

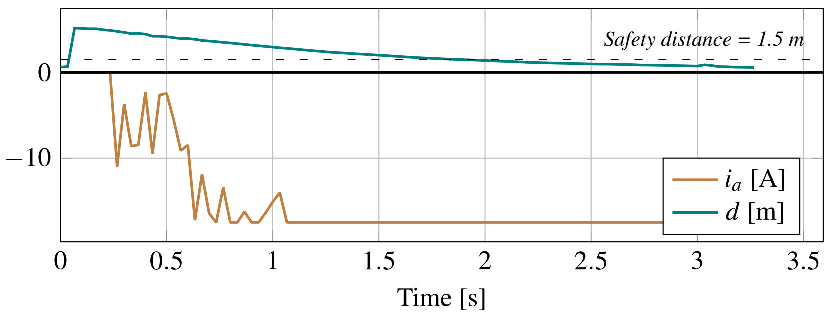


(c) a_R is the measured acceleration data and a_{KF} is the Kalman filtered acceleration data

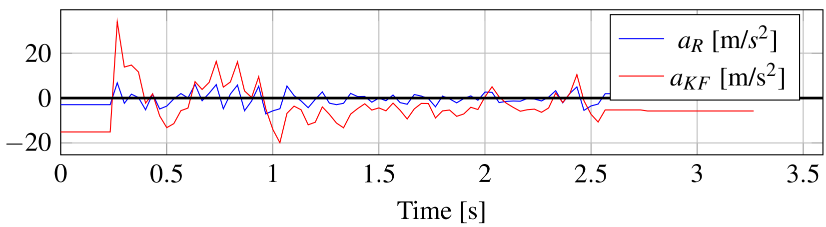
Figure 7.9 Braking test conducted with Driver 2 driving at 5 km/h



(a) v_s is the current speed and a_t is the target acceleration



(b) i_a is the braking current sent to the motor and d represents the distance to the object. The dashed line at 1.5 m depicts the safety distance to the person



(c) a_R is the measured acceleration data and a_{KF} is the Kalman filtered acceleration data

Figure 7.10 Braking test conducted with Driver 2 driving at 10 km/h

Integral

The integral of the current, sent to the motors, during the braking maneuver describes when in time the braking force is accumulated. Analyzing this shows that strategy A accumulates more brake current in the initial stages, while B and C accumulates more brake current in the later stages. Strategy A had harsh braking initially which led to discomfort for the driver, see Figures 7.11 and 7.12.

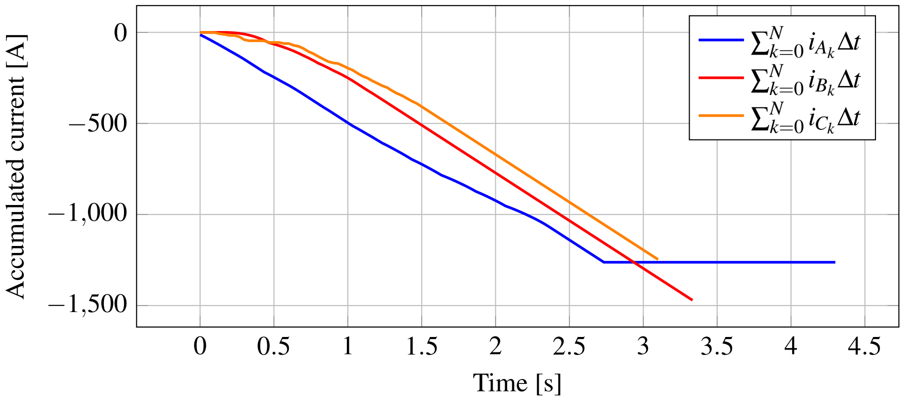


Figure 7.11 The integral of the braking current during braking test of Driver 1 at 8 km/h

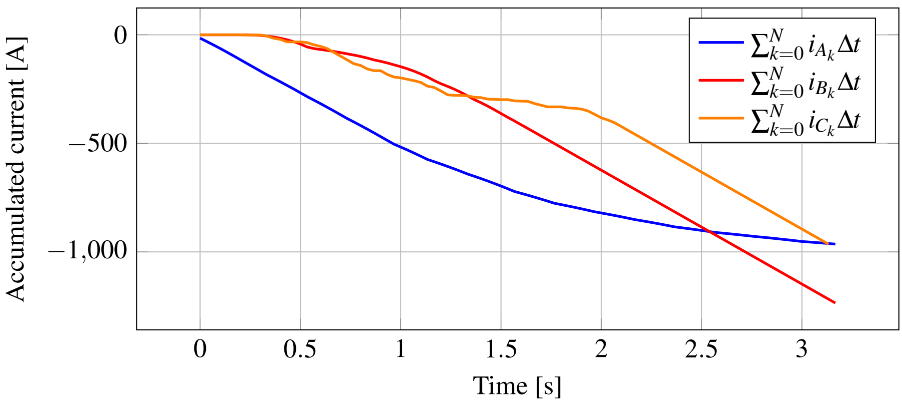


Figure 7.12 The integral of the braking current during braking test of Driver 2 at 8 km/h

7.2 Real-Time Results

There are two key aspects that will be measured through experiments: The capture and compute phase on the iPhone and the wireless communication between the iPhone and the VESC motor controller.

Capture and Compute

To evaluate if the real-time requirements for the system are fulfilled, the time was measured between selected key process steps for the AEB in the iPhone software. The time was measured 90 times for each step. The total time for these steps represent all time required from detection of a person to sending a brake value to the VESC control unit. The steps chosen are the following:

- CP: Choose Person, the time for the system to recognize a person and choose one if two are seen in the field of view based on the distance and if they are in the path.
- D: Distance, the time to measure the distance to the person using LiDAR
- FC: Fulfills Condition, the time to determine if the person fulfills all conditions to enable the AEB
- BA: Brake A, the time to set a brake value for strategy A
- BB: Brake B, the time to set a brake value for strategy B
- BC: Brake C, the time to set a brake value for strategy C

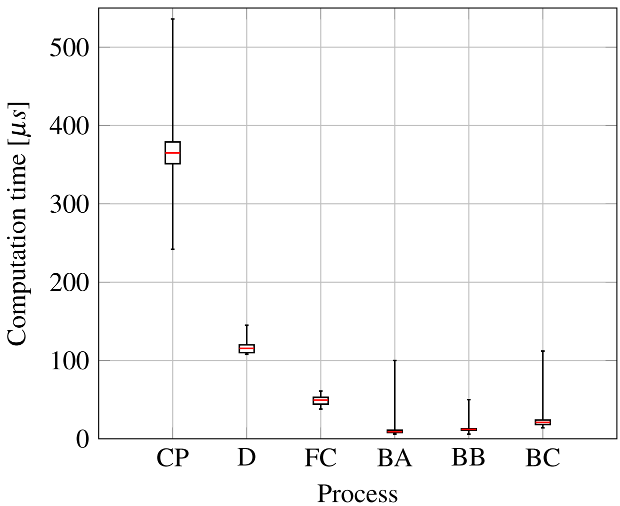


Figure 7.13 Capture and compute real-time results

The results can be seen in the box plot in Figure 7.13. A more detailed view of the computation times can be seen in Table 7.3 where the shortest, median and longest time are shown. The worst-case execution times are the summation of the longest execution times, which resulted in 842 μs for strategy A, 792 μs for strategy B and 854 μs for strategy C.

Table 7.3 The minimum, maximum and median computation time in [μs]

Process	**Shortest time**	**Median**	**Longest time**
Choose Person	242	365	536
Distance	108	115.5	145
Fulfills Condition	38	49.5	61
Brake A	6	8	100
Brake B	6	12	50
Brake C	14	21	112
Worst-case execution time for A			**842**
Worst-case execution time for B			**792**
Worst-case execution time for C			**854**

Communication

When a brake value has been decided, the information was sent to the VESC control unit via Bluetooth, and the time for this was also measured in a test. The communication was assessed by sending one hundred messages back and forth and measuring the time between each one being sent and then returning to the iPhone. Considering the short duration of time needed for the controller to engage the brakes, it is assumed that the time taken by the VESC controller to process a Bluetooth message and activate the brakes matches the time required for interpreting the message and forwarding it back to the iPhone.

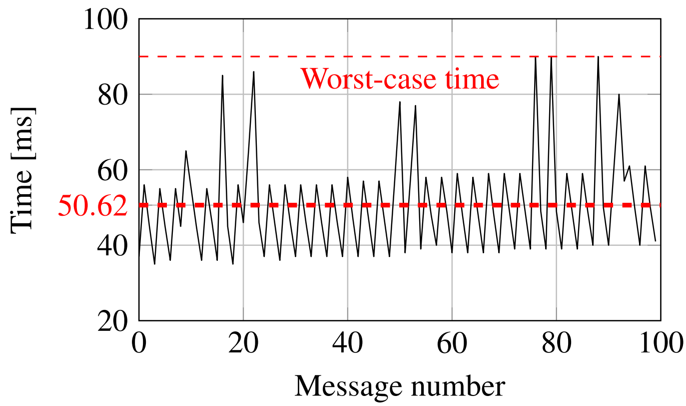


Figure 7.14 The round-trip communication time between the iPhone and the motor controller.

The result from the communication test shows that the average communication time is 50.62 ms, see Figure 7.14. The worst-case time was 90 ms. This result represents the communication to and from the motor controller. Together with the worst case execution time from Table 7.3, the total worst-case execution times is around 91 ms.

7.3 Close-Up Obstacle Detection

A function that was hypothesized to not work exactly as intended was the close-up obstacle detection, corresponding to the S_{NA} state in Chapter 6. The main concern for the function was that errors could occur if the person detection function found only one joint, often the hands or arms, and tried to measure the distance to it, see Figure 7.15. In this case, it may measure the area behind since the distance measurement was not designed to handle the distance to only one joint. To test the close-up obstacle detection function, tests were held that focused on the hypothesis that the detection of single joints are the cause of this problem.

The vehicle was driven from a point where the person detection algorithm clearly sees the person, and then the vehicle is moved closer to the person with the acceleration lever kept at the maximum level. For the close-up obstacle detection to work well, the vehicle should not be able to accelerate at any point in the test, as a person is present closely at all times. The result is set as "Yes" if it worked as intended, and "No" if the vehicle started accelerating.

The tests were held with two different test persons, first with the person standing with their hands behind their back, *Arms in*, and then with their hands held slightly out, *Arms out*, from their body. The results are presented in Table 7.4.

Table 7.4 Close-up object detection test

	Driver 1		Driver 2	
Test nr.	Arms in	Arms out	Arms in	Arms out
1	Yes	No	Yes	No
2	Yes	Yes	Yes	No
3	Yes	No	Yes	No
4	Yes	No	Yes	No
5	Yes	No	Yes	No

(a) Driver 1, distance = 8.17

(b) Driver 2, distance = 5.19

Figure 7.15 An example of distance measurement errors, where the number inside the red box indicates the perceived distance to the detected joint

8

Discussion

The stages of the project referred in the objective of the thesis have been carried out. The fundamental requirements were investigated and were used to model the kinematics of the vehicle. The ADAS feature chosen was an AEB and it was implemented and tested on the LEVKART platform and is evaluated in the following chapter.

8.1 Control Strategies

In the thesis, three different control strategies for an AEB system were evaluated and they will be discussed and compared based on comfort level and performance. The three different control strategies all performed differently but managed to fulfill the basic functionality of the AEB.

Strategy A: Simple Distance Feedback

Controller A succeeded in braking in front of the detected person in a safe manner. However, its harsh braking action, particularly at higher speeds, led to discomfort for the driver. At low speeds, 5 to 7 km/h, the braking was very comfortable as can be seen in Tables 7.1 and 7.2. At the velocities around 10 km/h and above, the comfort level was neutral. In these cases the controller sends a high brake current to the motors at the beginning resulting in a strong braking action, see Figures 7.2 and 7.4. The braking current directed to the motors is capped at -17.48 A, corresponding to an acceleration of -1 m/s^2.

Strategy B: PI Controller based on Velocity

Controller B performed very well at all speeds that were tested. The controller computes an ideal speed to be able to stop at the desired distance from the person ahead. The PI controller regulates the speed in order to keep this ideal speed profile, as can be seen in Figures 7.5b and 7.6b. All the tests in Tables 7.1 and 7.2, except one, resulted in very comfortable braking. The comfort value for 10 km/h in Table 7.2 is deviating from the others because of the feeling of driving at a high speed against an object, in particularly a person. The comfort level experienced by the driver can be related to the controller's slow but methodical increase in braking force, achieved through gradual adjustments in the braking current until it reaches the maximum braking current. This approach enables the driver to anticipate and adapt to the braking maneuvers. This can be seen in Figures 7.5b and 7.6b. It can also be seen when comparing the integral of the braking current to that of strategies B and C that the braking rates are different, which can be seen in Figures 7.11 and 7.12. Strategies B and C have a lower rate of decrease than strategy A, indicating a smoother and more comfortable AEB.

Strategy C: PI Controller based on Acceleration

In comparison to the other control strategies, strategy C was very uneven and in several cases the brake current switched rapidly from negative to naught and back to a negative value, see Figures 7.7b and 7.9b. This behavior only occurred when the accelerometer data deviated significantly, leading to a substantial error in the PI controller. When the brake current is set to zero, the motor controller disengages, allowing the wheels to rotate freely without any current supply. Upon receiving a negative current value, the regenerative brakes are engaged, exerting a braking force. This behavior led to the unpleasant feeling experienced when controller C was activated. But as can be seen in Tables 7.1 and 7.2, the unwanted behavior was not always prominent. The controller performed as intended for both drivers at 10 km/h, i.e., gradually decreasing the braking current until it reaches the limit, see Figures 7.8 and 7.10.

Siciliano et al. (2008) argues that designing a controller based on raw acceleration may be problematic. The presented solution to this is either filtering the acceleration or creating a feed-forward action based on the current instead. In any further development, this should be considered.

Summary Control Strategies

Controller A effectively braked in front of detected persons but exhibited discomfort due to harsh braking actions at higher speeds. The PI controllers demonstrated smoother braking actions, with controller C occasionally exhibiting rapid and uneven braking, though performing as intended at 10 km/h. The best performing controller was strategy B.

8.2 Final Stage Braking

Several factors contribute to problems in the final stages of the braking algorithm, close to the detected person. One of these factors is the computation of the target acceleration if the vehicle passes the safety distance of 1.5 m.

The target acceleration is computed in Equation (6.3) with the distance being the result of Equation (6.1). As a result of this, the distance can be negative, which leads to a positive target acceleration. This effect can be seen in Figures 7.1, 7.2, 7.4, 7.1, 7.7, 7.9 and 7.10 as the target acceleration spikes. This spike does not occur in the brake current, the current is set to maximum deceleration or zero, depending on the braking strategy. The phenomenon does not lead to any problems in the presented test scenarios, but would need to be considered in any further development to improve the system's robustness.

The close-up obstacle detection algorithm did not function as intended in the experiments described in Section 7.3. The hypothesis that motivated the experiments was that if a person holds their arms and hands slightly out from their body, the person detection algorithm will react incorrectly, which can be seen in Figure 7.15. The results presented in Table 7.4 shows that this hypothesis is correct as the algorithm functions as intended if the hands are kept behind the back, but not with the arms out. This did not cause any problems in the braking tests as the acceleration lever was released when the AEB started. In any further development, this would be an important problem to address.

8.3 Real-Time Analysis

The real-time experiments in Section 7.2 resulted in values for the time the AEB takes to react to a person and engage the brakes. The worst-case execution times for the capture and compute stages is around 800-850 μs and for the communication it is 90 ms, resulting in an average worst-case execution time of approximately 91 ms. This means that the main time delay from the first to the last stage of the AEB algorithms is the communication over Bluetooth. To improve the total real-time performance, the main focus should be to find an alternative for the communication. The real-time tests were performed with one person, which means that the "Choose Person" part of Figure 7.13 may vary if several people are present in the view of the camera and therefore adding execution time to the overall process.

8.4 Usability

The result from the braking tests in Section 7.1 show that the AEB functionality works for most of the cases. This will be discussed in relation to possible real applications.

Speed and Brake Distance

The well functioning and predictable control strategies A and B brake the vehicle in a safe way up to approximately 10 km/h according to Tables 7.1 and 7.2. This speed limit is mostly a result of the control strategies being limited to an acceleration value of $-1\ \mathrm{m/s^2}$, as that was considered the maximum comfortable and safe acceleration. Figure 8.1 shows the braking distances and corresponding times for braking with a constant acceleration of $-1\ \mathrm{m/s^2}$ at different speeds. The minimum braking distance for 10 km/h is 4 m. This means that to increase the velocities that the AEB functionality works for, either the maximum deceleration value or braking distance could be altered. However, this would need to be balanced against the safety of the feature.

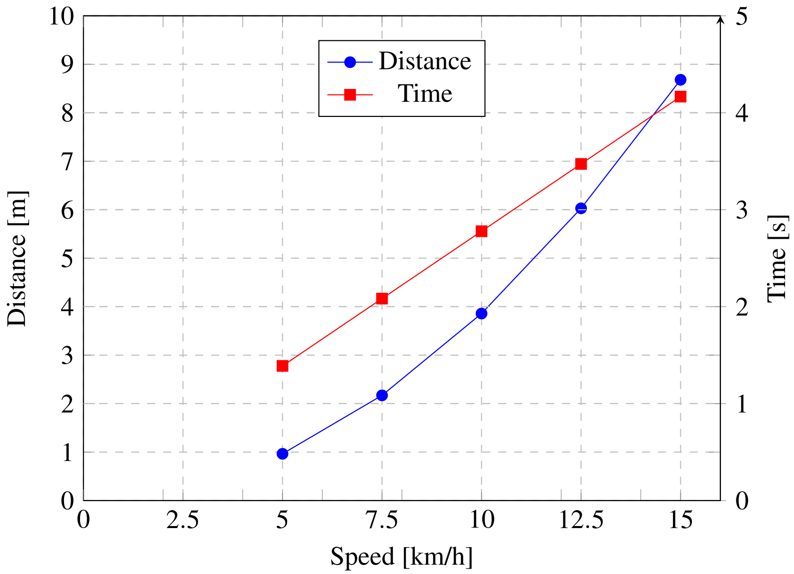


Figure 8.1 Distance vs speed at $-1\ \mathrm{m/s^2}$

Environments

An important aspect to consider if the AEB would be developed further is the environment it should be developed for. In the scope of this thesis, the focus was to successfully brake in an empty space if a person stands still in the path of the vehicle. This dictated the choice for braking distance as well as maximum deceleration value and enabled the person detection algorithm to be less complicated. To ensure the AEB has the desired functionality in, for example, crowded city streets, these aspects would likely need to be updated. The braking distance is also affected by the friction between the wheels and the road surface. All tests were performed in a warehouse environment on concrete floors, which will have an impact on the result. For further development other road surfaces should be considered.

9

Conclusion

The test results from the different control strategies are promising for future development, as all three performed well at all test velocities. Each controller slowed down the test vehicle when approaching a person, demonstrating their capability to respond to an emergency situation. The primary focus of the experiments was to assess the comfort levels associated with each strategy, prioritizing the driver's sense of control. While the simple distance feedback controller A effectively stopped the vehicle, it did so less comfortably. The PI controller with acceleration, strategy C, showed mixed performance, because of unreliable and shifting accelerometer data, resulting in occasional discomfort. Finally, the PI controller with velocity, strategy B, consistently delivered comfortable braking experiences across all speeds. Strategy B is therefore the recommended control method for the application.

9.1 Future Work

Despite the AEB functioning well, it has a lot of potential to be improved. Primarily, limited time constraints prevented comprehensive testing and fine-tuning of the implemented Kalman filter, resulting in its inadequate performance. With more time and testing capabilities strategy C could be improved.

The algorithm for detecting obstacles does not consider curved paths, limiting its effectiveness to scenarios where the vehicle travels along a straight line. An improvement would be to incorporate the steering angle and the vehicle's turning radius so it does not start braking when avoiding a person.

The person detection algorithm and the close-up object detection require improvement to prevent scenarios where drivers can accelerate when a person is present in front of the vehicle. This could be achieved by incorporating additional constraints to enhance the accuracy of determining whether the detected joint is indeed a person or not.

The tests were conducted within a warehouse environment with a restricted amount of people. Conducting more extensive tests in diverse settings would provide a more comprehensive evaluation of the algorithm's performance.

The prototype vehicle that was used during the tests was solely equipped with electric brakes, that used regenerative braking to stop. These are less effective at lower speeds, leading to extended braking distances compared to conventional mechanical brakes. An improvement for future iterations is to add mechanical braking force at the end of the braking action.

Bibliography

Abduljabbar, R. L., S. Liyanage, and H. Dia (2021). "The role of micro-mobility in shaping sustainable cities: A systematic literature review". *Transportation Research Part D: Transport and Environment* **92**. ISSN: 1361-9209. DOI: https://doi.org/10.1016/j.trd.2021.102734.

Alfian, R. I., A. Ma'arif, and S. Sunardi (2021). "Noise Reduction in the Accelerometer and Gyroscope Sensor with the Kalman Filter Algorithm". *Journal of Robotics and Control (JRC)* **2**:3, pp. 180–189. ISSN: 2715-5072. DOI: 10.18196/jrc.2375.

Apple Inc. (2024a). *Detecting human body poses in images*. URL: https://developer.apple.com/documentation/vision/detecting_human_body_poses_in_images (visited on 2024-04-08).

Apple Inc. (2024b). *Iphone 15 pro technical specifications*. URL: https://www.apple.com/iphone-15-pro/specs/ (visited on 2024-04-08).

Apple Inc. (2024c). *Mlcomputeunits*. URL: https://developer.apple.com/documentation/coreml/mlcomputeunits/ (visited on 2024-04-08).

Ariyanto, M., G. D. Haryadi, M. Munadi, R. Ismail, and Z. Hendra (2018). "Development of Low-Cost Autonomous Emergency Braking System (AEBS) for an Electric Car". In: *2018 5th International Conference on Electric Vehicular Technology (ICEVT)*, pp. 167–171. DOI: 10.1109/ICEVT.2018.8628442.

Årzén, K.-E. (2024). *Real-time control systems.* Department of Automatic Control, Lund University.

Åström, K. and R. Murray (2008). *Feedback Systems: An Introduction for Scientists and Engineers*. English. Princeton University Press. ISBN: 9781400828739.

Bengler, K., K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner (2014). "Three Decades of Driver Assistance Systems: Review and Future Perspectives". *IEEE Intelligent Transportation Systems Magazine* **6**:4, pp. 6–22. DOI: 10.1109/MITS.2014.2336271.

Corrigan, S. (2016). *Introduction to the Controller Area Network (CAN)*. Tech. rep. Texas Instruments.

Denis, F. (2023). *Core ML performance benchmark, 2023 edition*. URL: https://www.photoroom.com/inside-photoroom/core-ml-performance-benchmark-2023-edition (visited on 2024-03-18).

Gustafsson, F. (2010). *Statistical Sensor Fusion*. Studentlitteratur. ISBN: 9789144054896.

Hägglund, T. (2000). *Reglerteknik AK, Föreläsningar*. Institutionen för reglerteknik, Lunds universitet.

Lynch, K. M. and F. C. Park (2017). *Modern Robotics: Mechanics, Planning, and Control*. 1st. Cambridge University Press, USA. ISBN: 1107156300.

NOAA (2012). *What is LiDAR*. URL: https://oceanservice.noaa.gov/facts/lidar.html#:~:text=Lidar%20%E2%80%94%20Light%20Detection%20and%20Ranging,Lighthouse%2C%20Dry%20Tortugas%2C%20Florida. (visited on 2024-05-08).

Nyberg, C. (2014). *Mekanik, Stelkroppsdynamik*. Liber. ISBN: 9789147114443.

Rizzoni, G. and J. A. Kearns (2015). *Principles and Applications of Electrical Engineering*. McGraw-Hill Education, p. 1152. ISBN: 9780073529592.

Sexton, E. G. P., K. J. Harmon, R. L. Sanders, N. R. Shah, M. Bryson, and C. T. Brown (2023). "Shared e-scooter rider safety behaviour and injury outcomes: a review of studies in the United States". *Transport Reviews* **43**:6, pp. 1263–1285. DOI: 10.1080/01441647.2023.2219838.

Siciliano, B., L. Sciavicco, L. Villani, and G. Oriolo (2008). *Robotics*. en. 1st ed. Advanced Textbooks in Control and Signal Processing. Springer, London, England.

Statistiska Centralbyrån (2018). "Varannan svensk har övervikt eller fetma". URL: https://www.scb.se/hitta-statistik/artiklar/2018/varannan-svensk-har-overvikt-eller-fetma/ (visited on 2024-05-12).

United Nations Economic Commision For Europe (2021). *Statistics Of Road Traffic Accidents In Europe And North America*. United Nations Publications.

Veneri, M. and M. Massaro (2021). "The effect of ackermann steering on the performance of race cars". *Vehicle System Dynamics* **59**:6, pp. 907–927. DOI: 10.1080/00423114.2020.1730917.

Wallner, J. and H. Bérenger (2023). *Apple's vision framework: Exploring advanced image similarity techniques*. URL: https://medium.com/@MWM.io/apples-vision-framework-exploring-advanced-image-similarity-techniques-f7bb7d008763.

Ziebinski, A., R. Cupek, D. Grzechca, and L. Chruszczyk (2017). "Review of advanced driver assistance systems (ADAS)". *AIP Conference Proceedings* **1906**:1, p. 120002. ISSN: 0094-243X. DOI: https://doi.org/10.1063/1.5012394.

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER'S THESIS

Date of issue
June 2024

Document Number
TFRT-6248

Author(s)
Frida Lindbom
Jakob Petersson

Supervisor
Hannes Bergkvist, LEVTEK SWEDEN AB, Sweden
Björn Olofsson, Dept. of Automatic Control, Lund University
Yiannis Karayiannidis, Dept. of Automatic Control, Lund University, Sweden (examiner)

Title and subtitle
Advanced Driver Assistance System (ADAS) for an Ultra-light Electric Vehicle

Abstract

The aim of this thesis was to investigate, develop and implement an Advanced Driver Assistance System (ADAS) feature for an ultra-light electric vehicle, focusing specifically on an Automatic Emergency Brake (AEB). The fundamental requirements, for an ultra-light electric vehicle, were found and investigated through experiments. A person detection algorithm with camera and LiDAR was developed with the ability to measure distance to the detected person and the brake algorithm was designed and implemented with three alternative control strategies. The AEB was tested through experiments and the three control strategies were evaluated as well as the real-time properties and the overall performance. All three control strategies functioned, however the PI controller with a velocity strategy performed the best in regards to comfort. The AEB showed great promise in increasing the safety for ultra-light electric vehicles.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
1-62

Recipient's notes

Security classification