DEGREE PROJECT IN MECHANICAL ENGINEERING,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2017

Safety of Machine Learning Systems in Autonomous Driving

FADI AL-KHOURY

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF INDUSTRIAL ENGINEERING AND MANAGEMENT

Safety of Machine Learning Systems in Autonomous Driving

Fadi Al-Khoury

Master of Science Thesis MMK 2017:149 MES 015
KTH Industrial Engineering and Management
Machine Design
SE-100 44 STOCKHOLM

KTH Industriell teknik
och management

Examensarbete MMK 2017:149 MES 015

Safety of Machine Learning Systems in Autonomous Driving

Fadi Al-Khoury

Godkänt	Examinator Martin Törngren	Handledare De-Jiu Chen
	Uppdragsgivare	Kontaktperson

Sammanfattning

Maskininlärning, och i synnerhet deep learning, är extremt kapabla verktyg för att lösa problem som är svåra, eller omöjliga att hantera analytiskt. Applikationsområden inkluderar mönsterigenkänning, datorseende, tal- och språkförståelse. När utvecklingen inom bilindustrin går mot en ökad grad av automatisering, blir problemen som måste lösas alltmer komplexa, vilket har lett till ett ökat användande av metoder från maskininlärning och deep learning. Med detta tillvägagångssätt lär sig systemet lösningen till ett problem implicit från träningsdata och man kan inte direkt utvärdera lösningens korrekthet. Detta innebär problem när systemet i fråga är del av en säkerhetskritisk funktion, vilket är fallet för självkörande fordon. Detta examensarbete behandlar säkerhetsaspekter relaterade till maskininlärningssystem i autonoma fordon och applicerar en safety monitoring-metodik på en kollisionsundvikningsfunktion. Simuleringar utförs, med ett deep learning-system som del av systemet för perception, som ger underlag för styrningen av fordonet, samt en safety monitor för kollisionsundvikning. De relaterade operationella situationerna och säkerhetsvillkoren studeras för en autonom körnings-funktion, där potentiella fel i det lärande systemet introduceras och utvärderas. Vidare introduceras ett förslag på ett mått på trovärdighet hos det lärande systemet under drift.

KTH Industrial Engineering
and Management

Master of Science Thesis MMK 2017:149 MES 015

Safety of Machine Learning Systems in Autonomous Driving

Fadi Al-Khoury

Approved	Examiner Martin Törngren	Supervisor De-Jiu Chen
	Commissioner	Contact person

Abstract

Machine Learning, and in particular Deep Learning, are extremely capable tools for solving problems which are difficult, or intractable to tackle analytically. Application areas include pattern recognition, computer vision, speech and natural language processing. With the automotive industry aiming for increasing amount of automation in driving, the problems to solve become increasingly complex, which appeals to the use of supervised learning methods from Machine Learning and Deep Learning. With this approach, solutions to the problems are learned implicitly from training data, and inspecting their correctness is not possible directly. This presents concerns when the resulting systems are used to support safety-critical functions, as is the case with autonomous driving of automotive vehicles. This thesis studies the safety concerns related to learning systems within autonomous driving and applies a safety monitoring approach to a collision avoidance scenario. Experiments are performed using a simulated environment, with a deep learning system supporting perception for vehicle control, and a safety monitor for collision avoidance. The related operational situations and safety constraints are studied for an autonomous driving function, with potential faults in the learning system introduced and examined. Also, an example is considered for a measure that indicates trustworthiness of the learning system during operation.

Acknowledgements

I would like to thank all the people who contributed in some way to this thesis project. I appreciate the support of Prof. Martin Törngren and Naveen Mohan in hosting this project and our various discussions. I thank Prof. De-Jiu Chen, my supervisor, for his support and feedback, and also Lola Masson from the Laboratory for Analysis and Architecture of Systems (LAAS) in Toulouse for discussing SMOF with me. Finally, I thank Lars Svensson for help with the Swedish translation of my abstract.

Contents

1 Introduction

This section introduces the subject of safety for the context of autonomous driving. Automation systems are needed to deliver important functions that can be safety critical. Safety concerns become more difficult to tackle when the solutions to complex automation problems emerge from deep learning or machine learning, and involve elements that are non deterministic, and difficult to inspect for correctness. The research questions are formulated in this section, along with the corresponding delimitations. An outline of the thesis is then presented, followed by a list of contributions.

Automation is expected to continue to take on important functions in various products and services used by society. In safety-critical applications, faults in automation can have great consequences such as loss of life, damage to property, and financial losses. The safety of such systems needs to be assured with good confidence. The focus of this thesis is particularly on automation within autonomous driving. The SAE J3016 standard [1] describes five levels of driving automation, ranging from basic driver assistance in level 1, to full automation under all driving situations in level 5. The commercially available technology is currently at level 2, and higher levels are being developed. Audi has recently announced its A8 model for 2018 featuring level 3 automation [2]. With level 3, automated systems control the vehicle and monitor the environment under some driving modes, but a human driver is expected to intervene when requested as a fall-back measure.

Embedded systems are essential to many automation functionalities in autonomous vehicles. Several definitions exist for embedded systems, and more commonly relate to hardware and software aspects. For example, [3] offers a concise definition where "Embedded systems are systems which include a computer but are not used for general purpose computing." From a systems engineering perspective, a definition that can be suitable for the context of functional safety in autonomous driving is that an embedded system is "a system that is part of a larger system and performs some special purpose for that system (in contrast to a general-purpose part that is meant to be continuously configured to meet the demands of its users)" [4]. Common in applications with embedded systems is the requirement for real-time interaction with the world. Several subjects areas assist in the systematic design of embedded systems for safety: safety engineering, systems engineering, formal methods, and software testing.

Apart from safety, also several ethical and legal matters need to be considered when driving decisions are provided by algorithms. An interesting dilemma is discussed in [5] regarding how an algorithm should choose between two evils. This includes cases such as whether the algorithm should decide to sacrifice the vehicle's passengers in order to save a greater number of other people, or should it decide to protect the passengers at all cost. The answer to such a question depends on moral and philosophical viewpoints that vary across societies and individuals. A new General Data Protection Regulation, summarized in [6], is planned to be enforced across the EU in 2018. In this law, the user should be able to obtain an explanation regarding an algorithmic decision made on his/her behalf.

Several technologies are employed in the automotive industry and academia for monitoring the vehicle's environment, which include cameras, Radar, Lidar, and ultra-sonic sensors. The sensor data is fed to algorithms that support various functionalities, which may be of high relevance to safety. These algorithms may also be highly complex, making them difficult to analyze. The problem associated with using the sensor data for achieving the required functions is not possible to solve analytically for real world scenarios due to large input spaces (e.g. all combinations of pixel values). For such problems, solutions have emerged from the fields of Deep Learning, Machine Learning, and statistical signal processing. These approaches share a probabilistic paradigm. Depending on the method used, the probabilistic factors may or may not need to be explicitly treated when solving the problem. The answers are for some methods described in terms of probability distributions, while for other methods probabilistic factors affect the solution implicitly and are not indicated in the answers. This is especially the case with deep learning, where labeled data is used to train an artificial neural network to solve the problem. This thesis refers to systems utilizing probabilistic methods from these three fields as Learning Systems, and focuses explicitly on methods using supervised learning with labeled data used for training or statistical analysis.

When automation is used to drive the vehicle, there are several possible points of failure in the system. These include hardware and software malfunction, problems with network delays/synchronization, mechanical problems, and also the behavior of the learning systems used. Investigating the safety impacts of these elements in the real world for autonomous driving can be infeasible and also dangerous. This motivates alternative methods that instead involve the use of simulations. Of relevance also is the ISO 26262 [7] standard, which covers functional safety for electrical and/or electronic (E/E) systems in passenger vehicles, including software. This thesis focuses on the safety concerns with using learning systems in autonomous driving. The next section introduces the research questions that will be addressed.

1.1 Research Questions

The research questions investigated in this thesis are stated as follows:

1. What are the concerns when learning systems are used in safety critical applications, specifically in the pursuit of higher levels of autonomous driving (SAE level 3 and higher)?
   - Learning algorithms are affected by probabilistic factors inherent in their design. What are the concerns if we need to assure their safety in high-criticality automotive applications?
   - What are the implications when we need to comply with ISO 26262, specifically with regard to clauses 7 and 8 of part 3, which address hazard analysis, risk assessment, and the functional safety concept?
2. Given the challenges with regards to safety-assurance of learning systems, what solutions can be considered?
   - What are the prominent formal methods approaches, and how can we assess their suitability for different applications? For example: is it helpful to use methods that verify mathematical properties of inputs and outputs for the learning system, or is it more appropriate to introduce other architectural elements, such as safety monitors, that address behavioral/functional properties?
   - Using an example deep learning system in a speed control application, how could we design a simple safety monitoring solution that is feasible for this scenario. Would the monitor design complexity present challenges?
3. Due to cost and safety considerations for prototyping autonomous speed control systems in a physical environment, how can simulations assist in testing safety monitors and learning systems? What are the needed elements in the simulator to support safety studies of vision based learning systems? What commercial tools can we identify and test for this setup?

1.2 Delimitations

Some delimitations were made in order to better address the main focus of this work, while others had to be imposed due to limited project time and resources. Below is a list of the delimitations.

- The aim in this project is to study safety concerns with (supervised) learning systems and how safety can be assured for an example automotive application. Designing a high performing learning system is not a primary objective, and could even defeat the purpose of our investigation. Beyond achieving basic performance for conducting experiments, optimizing performance can also use up excess project time, considering the limited computational resources available.

- A discussion is presented regarding clauses 7 and 8 of the ISO 26262 part 3, which relate to hazard analysis and functional safety at the concept phase. Other parts of the standard are not discussed unless there is a need to reference them. The focus is on studying safety for learning systems at a conceptual level rather than undertaking product development activities.

- Although a safety monitoring strategy is developed for the general scenario of combined longitudinal and lateral motion, only the simpler frontal collision avoidance monitoring is tested in simulations.

- The approach presented in this thesis uses a model of the leading vehicle's expected future trajectory in collision avoidance. However, this is not addressed in any depth, and only the case of a fixed leading vehicle is considered.

- Fault tolerance within the control system by incorporating sensor fusion using different types of sensors is not considered. Although the simplicity of the design presented in Section 4.2 could be criticized, the focus in this thesis is not on sensor solutions yielding high robustness and performance, but rather is on the fundamental safety challenges with learning systems, and handling faults at an architectural level.

- A prototype vision-based speed control system will be developed in later sections. Imperfections in the image data due to noise, distortions, motion blur, are not incorporated, as this introduces unneeded variables to the study, and also increases computational demands in preparing experiments. The main interest is in the fundamental safety concerns when deploying learning systems in safety critical applications. Noise or imperfect data is a problem that is not specific to learning systems. Also for safety monitoring, the focus is on the functionality assuming perfect input information.

- An example measure is discussed and tested as an indicator of the learning system's trustworthiness during operation. The development of better indicators, or the investigation of desired properties of such indicators has not been attempted in this work.

1.3 Research Method

To address the goals for this thesis, literature is examined on relevant topics and a demo is built in a simulation environment. Relevant literature is identified on learning systems, especially with regard to safety and how the problem can be defined in the context of safety. Also, the types of learning systems that are most relevant in autonomous driving are identified, along with particular safety concerns with their use. The ISO 26262 is consulted for a recommendation on how to meet safety goals given the challenges with assuring the safety of learning systems. Literature is also examined on formal methods, and how they can fit into safety assurance for autonomous driving, beginning with the SMOF approach [8, 9] which is of interest at the Embedded Control Systems devision at KTH. The knowledge and experience gained is then demoed in a virtual vehicle simulation setup, and experiments are performed to further examine the safety concerns and seek insights.

1.4 Thesis Outline

This thesis studies the safety concerns with learning systems delivering critical functionalities in autonomous driving, and investigates an architectural approach for meeting safety requirements. This section introduces the topic of safety in the context of autonomous driving, and the use of learning systems in safety-critical applications. Section 2 discusses the safety concerns, as well as the use of learning systems for autonomous driving. This is followed by a discussion of the ISO 26262 standard and the architectural approach to addressing safety is considered. Section 2.3 reviews the literature on safety monitoring and identifies a promising approach. Section 3 applies the safety monitoring approach to the context of collision avoidance. A case study is presented in Section 4, in which simulation is used to support safety studies of safety monitoring and control. The control system considered uses camera images as input, and utilizes deep learning. An indicator is presented for the trustworthiness of the learning system during operation. Experiments are then performed to study safety and test collision avoidance. Finally, Section 5 discusses and summarizes the work in this thesis, and offers suggestions for further research.

1.5 Contributions

The contributions of this thesis are the following:

- A discussion is made of the safety concerns with deploying learning systems in safety critical automotive applications in Sections 2.1 and 2.2. Safety monitoring approaches are also discussed along with their complexity considerations in Section 2.3.

- Models for longitudinal as well as combined longitudinal and lateral collision avoidance scenarios are developed and used within a safety monitoring approach in Section 3.1.

- An approach to safety distance calculation based on time varying relative acceleration models is presented in Section 3.2. Constant acceleration models are shown to offer relatively optimistic safety distances in comparison, which is not desirable.

- The use of simulations for supporting safety studies of camera-based learning systems is discussed in Section 4.1, and three simulation tools, IPG CarMaker, TESIS DYNAware, and TASS PreScan are considered for this application.

- A simulation-based approach is presented in Section 4.3 for performing safety studies with camera-based vehicle control, where a full control and safety toolchain can be tested. Experiments with malfunctioning behaviors in a vision-based speed control system are also presented. An architectural safety monitoring approach is applied and demonstrated for collision avoidance in case of hazardous speed control behavior.

- An example runtime indicator is presented in Section 4.3.4, for the trustworthiness of a vision-based speed control system. Such an indicator can help warn in advance before severe safety interventions are needed.

2 Background

This section presents the background relevant to the discussions of this thesis. A function approximation view of learning systems is introduced and used to discuss safety concerns. Applications for learning systems in autonomous driving are then discussed, with particular mention of the end-to-end approach to solving automation problems, and the safety implications that arise. The ISO 26262 is discussed for this context, and an architectural approach to safety is considered. Safety monitoring is then introduced, and a promising approach in the literature is identified.

Although developments in automation may yield good performance, and even eliminate sources of human error, there are several sources of failures that introduce hazards. For the case where the automation systems employ learning algorithms, these sources can include unreliable sensors, problems with the data used for training, as well as limitations in the algorithms and faults in their design.

With regard to the learning systems addressed in this thesis, Vapnik [10] provides a definition that is helpful in this context. In Vapnik's paper, learning is posed as a problem of function approximation involving three components:

1. A generator of random vectors x, drawn independently from a fixed but unknown distribution $P(x)$.
2. A supervisor which returns an output vector y to every input vector x, according to a conditional distribution function $P(y|x)$, also fixed but unknown.
3. A learning machine capable of implementing a set of functions $f(x, w)$, where $w \in W$ is a set of parameters to be learned.

To illustrate these components in an example application, consider the case where a deep neural network classifies input images. The generator would be the training images used to train the network, with each image being a vector x in Vapnik's model. The supervisor would be the ground truth labeling process for the training images, where the class label is encoded in the output vector y referred to above. A commonly used distinction in machine learning literature is between supervised and unsupervised learning. In the former, ground truth labels are available, as is the case in our image classification example. With unsupervised learning on the other hand, ground truth labels are not available and other mechanisms are used. A key point is that $P(y|x)$ is unknown, and in effect sampled for available x vectors. Finally, the implementation of the particular deep neural network architecture would constitute the learning machine. The term w refers to the network parameters such as the weights and biases. The resulting output of the neural network, depends on the input image x, and the network parameters w used, which are determined by the training process. The concept of learning systems discussed in this thesis uses this function approximation view of supervised learning.

A discussion is presented in [11, 12] on the critical role of training data and loss functions, which represent training objectives, for the safety of machine learning systems. For the loss functions, the authors note that with quantities related to performance or prediction error, the human cost that is relevant for safety may not be accounted for. For example, consider the trivial scenario where the steering angle for a vehicle is learned by a neural network from labeled data. The quantities relevant for safety may include the risk of collision, and the risk of injury or death, which depend on the operating environment. If the backpropagation method is used with the angle error alone, these safety considerations are not incorporated in the loss function. Also, arguments based on laws of large numbers (e.g. average error) may not be suitable when considering safety, since safety-critical cases may be rare and underrepresented in an aggregate quantity. Two concerns are noted by the authors for the training data

- The training samples may not be drawn from the underlying distribution, $P(y|x)$. For the image classification example, consider the case when the training images are produced by ideal sensors, and contain no noise, distortions, or motion blur, but the actual $P(y|x)$ needed for the application is for non-ideal realistic images. By using ideal training images, the training samples were drawn from a different distribution than $P(y|x)$. In addition, the difference between the target distribution and the distribution from which training samples are drawn may not be easily understood or possible to account for.
- The training samples are absent in parts of the x vector space. This can especially be the case for rare, low probability, x regions. Safety relevant rare conditions may be insufficiently represented, or not represented at all.

There may be other concerns the authors do not explicitly include, such as the formulation of the learning problem with adequate and suitable information to infer outputs. This can be viewed as sampling from $P(y|z)$ when $P(y|x)$ is the needed distribution for the problem. If the mapping between z and x is many-to-one, then simply extra training data would be needed. If the mapping is one-to-many, or many-to-many then nondeterminism is introduced, affecting the potential for inferring the output. Overfitting is another problem in supervised learning, which is related to the complexity of the learned system (e.g. the number of weights) relative to the amount of training data. A complex model can fit exactly a set of points that draw a straight line with an added noise, however, it will generalize very poorly on unseen data compared to a simple model for a straight line. This can be related to the discussion on the loss functions. When using only the error in the loss function, the complex model would be favored, however, an additional term can be incorporated to avoid overfitting by penalizing model complexity. This technique is known as regularization. Overfitting can also be related to causality and interpretability. The higher model complexity can allow for fitting noise and vagaries in the training data that are not part of the underlying physics, and also make the model more difficult to understand.

The authors suggest four strategies for improving safety:

- **Inherently Safe Design** Causality and interpretability can be insisted on in the design of systems, which allows for the behavior of the models in response to inputs to be understood and verified. Irrelevant input and output variables need to be eliminated, to expose the main ”physics” of the system governing its functional and safety properties. Interpretability refers to the possibility of understanding the models and their operation, for e.g. whether a model is black box, gray box, white box, etc. Interpretable models can much more easily be examined with respect to safety or other properties. The authors, do not include causality as a safety concern beside the training data and loss functions above, but it can contribute an offset between the target and training data distributions. For example, consider the case discussed in the previous paragraph regarding sampling from $P(y|z)$ when $P(y|x)$ is the needed instead.
- **Safety Reserves** In [11, 12], this refers to optimizing not only with respect to the model parameters, but also with respect to uncertainties in training data. A practical example is not suggested, but the key is to consider the worst case outcome while varying uncertainties, whether due to training and test distribution mismatch or instantiation of the test set. It is also suggested to consider fairness and equitability, so that certain groups are not underrepresented for safety.

- **Safe Fail** With this strategy, the system may elect a reject option, in which it does not attempt to predict the output for the given input sample. For example, in regions that are too close to decision boundaries, or regions that correspond to rare input conditions, a safe fail would be to ask the human operator to intervene.
- **Procedural Safeguards** User experience design can help guide practitioners on how to correctly setup the machine learning system. This includes the training data set, evaluation procedures, and other elements. If automated design processes are performed by users who are not deeply knowledgeable of the systems and associated safety concerns, incorrect use can be a concern. Another procedural safeguard is to have the possibility of public audit of source code, so that potential problems can be discovered. Having the data source publicly available is also suggested by the authors.

The next section discusses some autonomous driving applications from the literature with regard to safety. Following this, the relevant safety standard for such applications is considered, and then the topic of safety monitoring is introduced.

2.1 Safety Issues When Using Learning Systems for Autonomous Driving

A pioneering work in this area is [13] from 1989, where a neural network computes the steering angle for a vehicle using camera and laser rangefinder inputs. This approach later came to be called end-to-end learning, since the algorithms are not hand designed. After the introduction of convolutional neural networks (CNN), end-to-end learning was utilized in a famous project by LeCun et al. [14], to drive a model truck on unknown open terrain while avoiding obstacles. More recently, advancements have been made in CNNs for vision tasks, which were driven by improvements in computational resources and network designs with more hidden layers. Networks with large amount of layers are referred to as "deep" networks in the literature, with the corresponding phrase "deep learning" also used. [15] used Deep CNN for end-to-end steering in lane and road following.

The main attraction with the end-to-end approach is that no domain expertise is required on how the system should solve the problem. Although some expertise would be required for selecting a suitable training data set and loss functions for the problem, the process for solving the problem is inferred implicitly from data.

Relating to the interpretability concept mentioned earlier, end-to-end systems can be viewed as uninterpretable, black box, computations. Consider that solving the main problem controlling a vehicle using vision information consists of several logical subproblems, such as: lane tracking, scene understanding, vehicle state awareness, coordinating with other vehicles, action policy, and others. The logic of how an end-to-end system solves the main problem is not understood, neither is the correctness of the solutions to each logical subproblem. This presents a safety concern in its own right, and compounds the previously mentioned challenges regarding adequate training data.

For safety, training data needs to cover rare cases that exercise all logical subsystems of an end-to-end system. To illustrate the challenge, consider a problem composed of two subproblems: P_A and P_B. Since with end-to-end systems the solutions to the subproblems are not inspected separately, faults in one subproblem may be masked in the aggregate system. Consider fault F_A which affects the solution to P_A. Due to a masking affect, this particular fault may not always be detected in the aggregate system. Rather than requiring simply that the training data accounts for F_A, in an end-to-end system the data needs to exercise F_A such that it is detectable in the aggregate system. The rarity of needed data could be much higher than if subproblems could be inspected separately. As the complexity of problems increases for end-to-end systems, it becomes increasingly harder to alleviate safety concerns associated with adequate training data.

Analytical methods for verification of properties of deep networks have been proposed in the literature [16, 17]. However, these methods use a mathematical view of the system's inputs and outputs, where one needs to specify safety properties concerning the values of input pixels and output quantities. Even if this could be achieved, the likely complex methods required for deriving such specifications would need verification efforts of their own.

Next, a brief discussion is presented of some relevant portions of the ISO 26262 safety standard, in relation to safety critical autonomous driving applications.

2.2 The ISO 26262 Standard

As the automotive industry aims at higher levels of automation, there is a trend of increasing complexity in the various software, hardware, and mechatronic systems deployed in vehicles. Due to the challenges in scaling the safety assurance with system complexity, as well as the critical application areas involved, safety becomes increasingly a major issue. The ISO 26262 [7] standard addresses the functional safety of electrical and/or electronic (E/E) systems within road vehicles, which for example include systems for: driver assistance, propulsion, and vehicle dynamics control. Safety requirements can be at the technical or functional level. Technical safety requirements pertain to safety of systems at the technical implementation level, and can for example include: voltage limits, memory requirements, speed ranges, safety distances, reaction times, etc. Functional safety requirements, on the other hand is implementation-independent, and relates to a higher level of abstraction which addresses behaviors of systems in response to inputs. This can involve, for example: actuator actions, computational flow within software, and transitions across system states. For functional safety, the system as a whole needs to be considered, along with the environment in which it operates.

The aim in the ISO 26262 is to assure the absence of unreasonable risk due to hazards caused by malfunctioning behavior of E/E systems, including software. The standard provides an automotive safety lifecycle, with a framework for the elimination of hazards and minimizing residual risk. The Safety Integrity Level is an important concept in the standard, pertaining to the amount of rigor in safety requirements that would be suitable for the amount of risk involved. Also, requirements for validation of safety and for relations with suppliers are addressed in the ISO 26262.

An overview of the ISO 26262 is shown in Figure 1. For the safety studies of learning systems presented in this thesis, the clauses in the standard which are most relevant are clauses 7 and 8 of part 3, addressing hazard analysis and functional safety at the concept phase of product development. The focus of this work is on the fundamental concerns with the use of learning systems. Safety within management, product development, and production activities is not of direct relevance, and is hence not discussed.

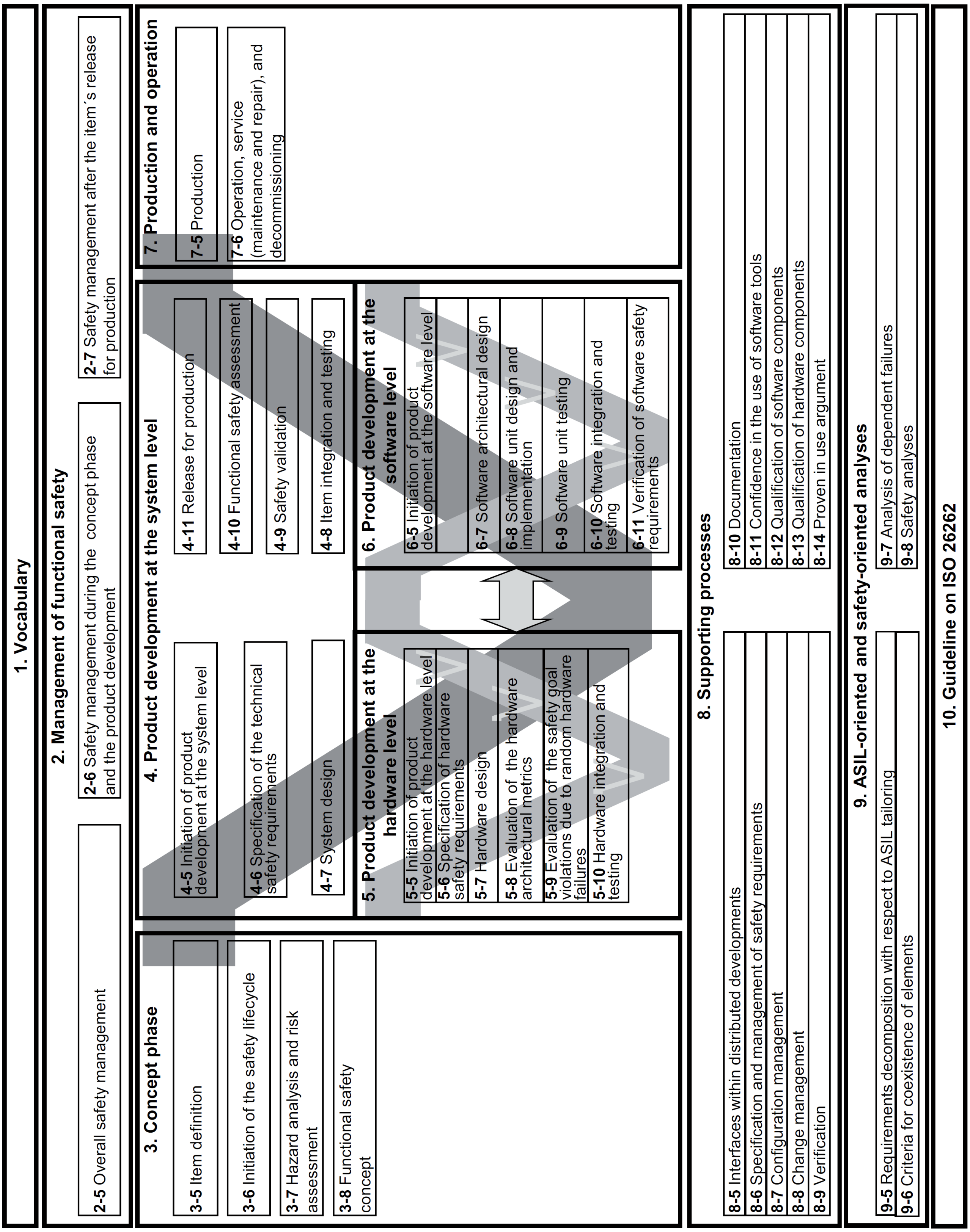

1. Vocabulary
2. Management of functional safety
2-5 Overall safety management
2-6 Safety management during the concept phase and the product development
2-7 Safety management after the item´s release for production
3. Concept phase
3-5 Item definition
3-6 Initiation of the safety lifecycle
3-7 Hazard analysis and risk assessment
3-8 Functional safety concept
4. Product development at the system level
4-5 Initiation of product development at the system level
4-6 Specification of the technical safety requirements
4-7 System design
4-8 Item integration and testing
4-9 Safety validation
4-10 Functional safety assessment
4-11 Release for production
5. Product development at the hardware level
5-5 Initiation of product development at the hardware level
5-6 Specification of hardware safety requirements
5-7 Hardware design
5-8 Evaluation of the hardware architectural metrics
5-9 Evaluation of the safety goal violations due to random hardware failures
5-10 Hardware integration and testing
6. Product development at the software level
6-5 Initiation of product development at the software level
6-6 Specification of software safety requirements
6-7 Software architectural design
6-8 Software unit design and implementation
6-9 Software unit testing
6-10 Software integration and testing
6-11 Verification of software safety requirements
7. Production and operation
7-5 Production
7-6 Operation, service (maintenance and repair), and decommissioning
8. Supporting processes
8-5 Interfaces within distributed developments
8-6 Specification and management of safety requirements
8-7 Configuration management
8-8 Change management
8-9 Verification
8-10 Documentation
8-11 Confidence in the use of software tools
8-12 Qualification of software components
8-13 Qualification of hardware components
8-14 Proven in use argument
9. ASIL-oriented and safety-oriented analyses
9-5 Requirements decomposition with respect to ASIL tailoring
9-6 Criteria for coexistence of elements
9-7 Analysis of dependent failures
9-8 Safety analyses
10. Guideline on ISO 26262

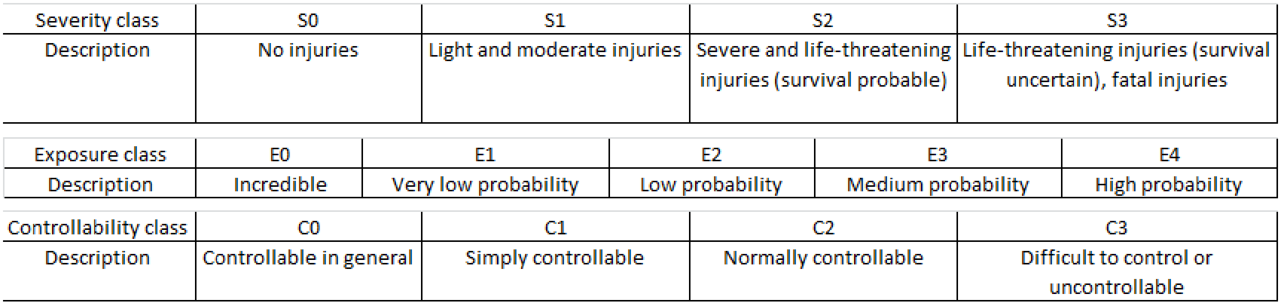
Figure 1: ISO 26262 overview [7]. Copyright remains with ISO (used with permission).

2.2.1 Hazard analysis, risk assessment and ASIL determination

In the ISO 26262, the term ”item” is used to refer to a system or array of systems that implements a function at the vehicle level, and to which the standard is applied. The item of interest in the investigations presented in this thesis can be defined as the autonomous driving function achieved via learning systems, without specifying any particular choice of learning systems, or the type of input and interfaces they involve. An example of such an item discussed in Section 4 is a speed control system using a deep neural network vision system. That item takes camera images of the road as input, and outputs brake and acceleration levels.

During hazard analysis and risk assessment the item is evaluated without internal safety mechanisms, with regard to its potential hazardous events (i.e. combinations of hazards and operational situations). Hazards that can result also from foreseeable misuse shall be analyzed. The hazards that can be triggered by malfunctions in the item are identified, and safety goals for the prevention or mitigation of the hazardous events are formulated.

The standard requires for hazards to be determined systematically. Some hazard analysis techniques mentioned in the standard are Failure Modes and Effects Analysis (FMEA), and Fault Tree Analysis (FTA). Hazardous events and their associated safety goals are assigned Automotive Safety Integrity Levels (ASILs), which are determined according to an assessment of the severity, probability of exposure, and controllability of the hazards. Figure 2 shows the classification classes for these factors.



Severity class	S0	S1	S2	S3
Description	No injuries	Light and moderate injuries	Severe and life-threatening injuries (survival probable)	Life-threatening injuries (survival uncertain), fatal injuries

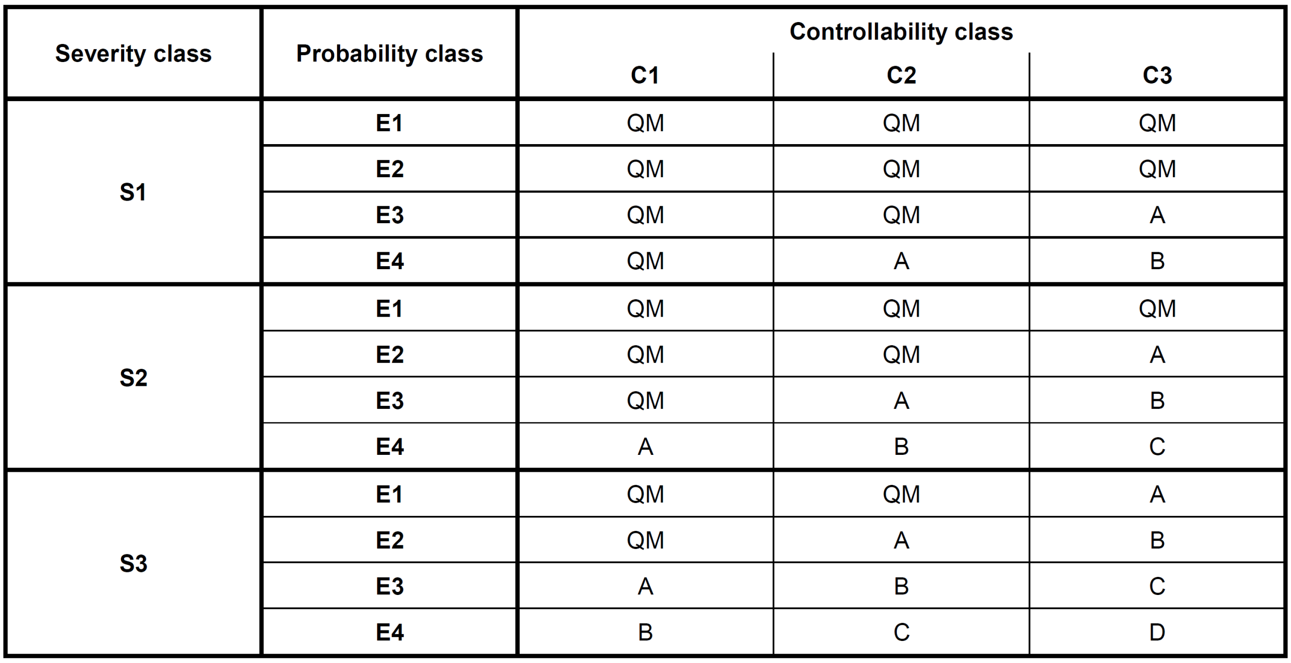
Exposure class	E0	E1	E2	E3	E4
Description	Incredible	Very low probability	Low probability	Medium probability	High probability

Controllability class	C0	C1	C2	C3
Description	Controllable in general	Simply controllable	Normally controllable	Difficult to control or uncontrollable

Figure 2: ISO 26262 classification of hazardous events [7].

Exposure and controllability assessments use estimates for the probability of exposure to hazardous situations, and the probability of avoiding harm, respectively. For classes E1-E4 and C1-C3, the difference in probability from one class to the next is an order of magnitude. It has been suggested in [18] to avoid probabilistic assessments as they bring in subjectivity for assumptions and analyses about the system, and that ASIL assessment entails a tension between safety and business competitiveness considerations for manufacturers. The assignment of ASIL in the standard using the severity, exposure, and controllability classes is shown in Figure 3.

In the collision avoidance scenarios discussed in this thesis, the hazard represents a situation where the vehicle may collide with obstacles due to failures in control functionalities achieved via learning systems. The corresponding safety goal is to avoid colliding with obstacles due to malfunctioning vehicle control from learning systems. Collisions can lead to fatal injuries, which are not avoidable without safety measures. This requires severity class S3, and a controllability class C3 classifications. Exposure refers to the relative frequency of exposure to the possibility of the hazard occurring. For example, hazardous events related to airbags have low exposure since airbags rarely deploy, while hazardous events related to braking systems have high exposure as braking is relevant in many driving situations. For example, an omission failure or braking, is relevant and dangerous in many driving situations - thus having high exposure. In the context of safety critical learning systems used in autonomous driving, the exposure is high when essential driving functions are automated. This would require an exposure classification of E4. The ASIL that results from these severity, exposure, and controllability classifications is an ASIL D. Furthermore, a challenge when using use a data-driven design process rather than analytic methods, is the presence of not only known unknown factors, but also unknown unknowns that are difficult to account for in probability assessment.



Severity class	Probability class	Controllability class		
		C1	C2	C3
S1	E1	QM	QM	QM
	E2	QM	QM	QM
	E3	QM	QM	A
	E4	QM	A	B
S2	E1	QM	QM	QM
	E2	QM	QM	A
	E3	QM	A	B
	E4	A	B	C
S3	E1	QM	QM	A
	E2	QM	A	B
	E3	A	B	C
	E4	B	C	D

Figure 3: ASIL assignment in ISO 26262 [7]

The ISO 26262-9:2011, Clause 5, offers an opportunity to implement safety requirements by independent architectural elements, and assign a potentially lower ASIL to the decomposed safety requirements. This is referred to as ASIL decomposition. For learning systems whose safety is difficult to analyze and verify, it can be desirable to divert away safety compliance efforts to simpler architectural elements that achieve the safety goals. For the ASIL D safety goal of avoiding collisions due to learning systems' functions, Clause 5.4.10, allows a decomposition into one ASIL D(D) requirement and one QM(D) requirement. The class QM (quality management) denotes that complying with the standard is not required. The ASIL QM(D) can be assigned to the learning system responsible for the autonomous driving functions, while an ASIL D(D) requirement can be assigned to a dedicated safety monitor for which it is simpler to assure safety compliance.

2.2.2 Functional safety concept

Clause 8 of ISO 26262-3 addresses the functional safety concept, which involves the allocation of functional safety requirements to architectural elements of the item, or to external measures. With the safety goal of avoiding collisions stated earlier decomposed into one ASIL D(D) requirement and one QM(D) requirement, an architectural element can be introduced to satisfy the ASIL D(D) requirement, while the original learning system can be exempt from needing to comply with the standard. The ASIL D(D) safety requirement can simply be to avoid colliding with obstacles. The added architectural element for this requirement can be considered as a safety monitor or a collision avoidance system.

The exact definition for this system would vary with implementation and technology choices, including the type of sensors used and the available system outputs. ASIL decomposition helps to divert safety assurance efforts from the complex systems that employ learning systems. However, complexity in the safety monitoring device can also increase verification efforts and should be minimized.

In this thesis, safety is addressed not by attempting to verify the safety properties of learning systems, and not by employing strategies that foster safety within learning systems. Instead, the idea is for safety concerns to be diverted to a separate architectural component for achieving the needed safety goals, whose verification is much simpler. Safety monitoring is considered for applications where the learning system drives the vehicle. Only accidents associated with colliding with obstacles are addressed, for which case collision avoidance is of relevance, and safety monitoring should achieve this function.

2.3 The Safety Monitoring Approach to Safety

Safety monitoring is not a new paradigm, although other terms have also been used to describe the approach. An early work in this area is [19] from 1987, which shows a formal methods approach that generates a supervisor from a system model and a constraint model both expressed as automata. The supervisor issues inhibitions to prevent hazardous actions. An example offered is when two users should not simultaneously access a shared resource. The objective would be to inhibit transitions in order to satisfy synchronization requirements. The authors provide a formal approach to proving that the needed supervisor exists, and that the corresponding synthesis problem is solvable.

Automated tools have been developed for verifying properties of automata models, known as model checkers. Model checkers exhaustively check the reachable states of the model. If the property holds, the model checker confirms it with full certainty, and provides some diagnostic information otherwise. Model checking has been used in the design of safety monitors. Siminiceanu and Ciardo [20] use their SMART model checker to verify the design and expose potential problems with the NASA airport runway safety monitor, which detects incursions and alert pilots.

With increasing complexity in system behavior and its environment, more descriptive models become needed, and their state space size increases. This presents a scalability challenge when using model checkers, since all the reachable states need to be checked. This is referred to as the ”state space explosion” problem. In [20], the popular NuSMV and SPIN model checkers were found not feasible for the runway safety monitoring application with a large state space.

One other consideration is that embedded systems exhibit both event-driven and time-driven phenomena. To represent such systems, hybrid models are used which include both discrete and continuous variables. Model checkers for hybrid systems exist, notably [21], however the computational complexity in model checking hybrid models is larger than for discrete models, which present further scalability difficulties. In applications where an implicitly discrete view of time and other continuous variables is used (e.g. due to sampling and quantization), it can be suitable to use discrete models.

Runtime verification is another paradigm related to model checking. Rather than using a model checker to verify the safety monitor offline, the safety monitor checks the execution of the system online. In effect, not all possible system executions are exhaustively checked, but only those that are encountered during system operation. This provides a fault detection mechanism. A key difference with model checking is that no model for the system is needed. A runtime verification architecture is proposed in [3] for safety-critical embedded systems. In this approach, execution traces are analyzed during system operation and checked against formal specifications. A separate controller is then triggered if a violation occurs. The focus of this work is more on the technical level, relating to the technical operation of systems. An example offered by the author is the following: ”If the brake pedal has been pressed, then within 200ms cruise control should be disengaged for 100ms.”

For collision avoidance, the top level safety goal is at the functional level, relating to the behavior of the vehicle within its environment. For complex scenarios, model checking can help ensure that this goal is achieved by the safety monitor, also if using runtime verification.

The work presented in this thesis will use the SMOF safety monitoring framework where the supervisor issues interventions, following [8, 9]. In this SMOF approach, the state space is partitioned into safe, warning, and critical regions. The safety monitor triggers interventions upon the system entering the warning regions to insure that transitions into critical regions are prevented. The authors also present an automated tool for the identification of warning states, and the synthesis of safety strategies given the available interventions. The main promise in the SMOF tool is the capability to produce a safety monitor given a system and a set of available interventions.

2.3.1 Application to Collision Avoidance

Collision avoidance (CA) requires not only the abstract safety rules that can be obtained with SMOF, but also the implementation of these rules. This includes detection and tracking of obstacles, and the needed planning and action steps for avoiding a collision. An overview of collision avoidance systems is found in [22]. It is noted that collision avoidance per se is not always attainable in automotive applications, and collision mitigation is a more realistic focus, i.e. reducing the severity of accidents. One main challenge, is defining the conditions in which the CA system should intervene. The author suggests examples where vehicles meet in a narrow road allowing little separation, and overtaking maneuvers that can be confused with imminent dangers. These challenges present a trade-off between effectiveness and unneeded interventions. Also, the work compares steering and braking maneuvers in terms of efficiency. For avoiding a head-on collision, with constant acceleration models the author shows that the needed separation distance is much higher with braking than steering as the initial speed increases. In fact, it was shown that with constant acceleration models, the separation distance is proportional to the square of velocity for braking, but linearly proportional to velocity for steering.

The concept of ”decision functions” is also presented in [22], in which dynamic models of the ego and other vehicles, as well as assumptions of future actions are basis for CA intervention decisions. Constant velocity and constant acceleration dynamic models are presented for estimating future trajectories. Several measures have been discussed for assessing the corresponding collision threat, including: distance, time to collision, closest point of approach, required acceleration, and others.

This thesis will present a different approach for integrating the two components, which is in alignment with the SMOF approach. Rather than using constant acceleration models, an approach will be shown involving time varying acceleration models for calculating the needed safety distance dynamically at run time, rather than only indicating a threat level or warning. This framework also allows for flexibility in defining the expectation of future actions of obstacles, which can be of help with the previously mentioned confusing scenarios (meeting at narrow roads and overtaking). However, this is not within the scope of this thesis and is not presented. In the following section, the main methodology is presented along with an example.

3 Safety Monitoring for Collision Avoidance

This section applies the SMOF safety monitoring approach to the context collision avoidance, both for the longitudinal (front and back) and lateral (sideways) cases. Next, the problem is formulated and the SMOF rule synthesis tool is tested with models for the two cases. The latter case will involve a much larger state space. After obtaining safety rules, the technical implementation of these rules is addressed.

3.1 SMOF for Collision Avoidance

In the SMOF framework, the safety system is responsible for monitoring safety and triggering interventions when the system transitions into a warning state. A formal description of the problem is presented next, but the reader may skip to Section 3.1.1 for an implementation of this approach. In the SMOF framework, the tasks may be viewed as follows:

1. Let the vehicle's surrounding regions be represented by a set of nondeterministic finite state machines (FSMs) $R_i, i = 1, 2, 3, ...n$, each possessing its own set of states S_i, generators Σ_i, and transition relations $\rightarrow_i$ with which jointly the FSMs satisfy constraints related to interdependency of different regions. Let the joint system be denoted by $\mathbf{T} = \{R_1, R_2, R_3, ..., R_n\}$ with the space of possible states $\mathbf{S}$, transition relations $\rightarrow$, and generators $\mathbf{\Sigma}$. Given a catastrophic combination of the sets $S_c \in \mathbf{S}$, obtain the set of predecessor warning states S_w

$$S_w = Pre(S_c) := \left\{ s_w \in \mathbf{S} : \exists \sigma \in \mathbf{\Sigma}, \exists s_c \in S_c, s_w \xrightarrow{\sigma} s_c \right\}.$$

2. Specify the available monitor interventions that if associated to warning states $s_w \in S_w$ transitions into catastrophic states can be possibly canceled. This can also be viewed as introducing an additional FSM (the monitor), so that the joint system can have no catastrophic states. Let the monitor be given by $M(S_m, \Sigma_m, \rightarrow_m)$, $\mathbf{T}' = \{M, \mathbf{T}\}$ denotes the joint system with the monitor. The monitor's states are derived from the states of R_i through a function F. In other words, $S_m = F(S_1, S_2, S_3, ..., S_n)$. To cancel catastrophic transitions some dependency has to exist also between the generators of R_i and the monitor's generators. The task at this step is to suggest the monitor's state derivations and generators, such that a cancellation of catastrophic transitions can be possible.

3. Find the possible mappings between the available monitor interventions and the warning states $\mathbf{s_w} \in \mathbf{S_w}$ such that the catastrophic states are no longer reachable. In the FSM view of the monitor, we need to now find the transition relation such that $\nexists \sigma_m \in \Sigma_m, s_c \in S_c$ that satisfy $F^{-1}(s_m) \xrightarrow{\sigma_m}_m s_c$. In the Joint system, the critical states are unreachable.

Referring back to the collision avoidance problem, the vehicle's autonomous driving systems contribute an additional actor P to $\mathbf{T}'$. Its states that are relevant to safety are contained in the space of possible states for the system with no monitor $S_p \subset \mathbf{S}$, but the generators $\mathbf{\Sigma}$ and transition relations $\rightarrow$ of $\mathbf{T}$ are pessimistically unconstrained by those of P.

In a comprehensive collision avoidance scenario, the vehicle's surrounding regions need to be considered longitudinally to avoid collisions from the front and back, laterally to avoid collisions from the sides, as well as possibly under the vehicle to avoid tires running on hazardous objects below. The boundaries for the zones depend on the possibilities for hazards and monitor interventions, and must be chosen such that intervention specifications are implementable. This thesis will next discuss a longitudinal collision avoidance model, and build up to a model for combined longitudinal and lateral collision avoidance.

3.1.1 Longitudinal Collision Avoidance

For the longitudinal case, an occupancy grid is considered in the form shown in Figure 4, consisting of safe, warning and critical regions. The boundaries of the regions, in terms of both shape and distance, are beyond the scope of SMOF, and are treated separately as technical specifications.

In the figure, the critical region is R_c, and the two warning regions are R_f and R_b, which respectively represent the front and back of the vehicle. With this simple model, the possibilities for obstacles to emerge into the critical region from below, or descend from above are not supported.

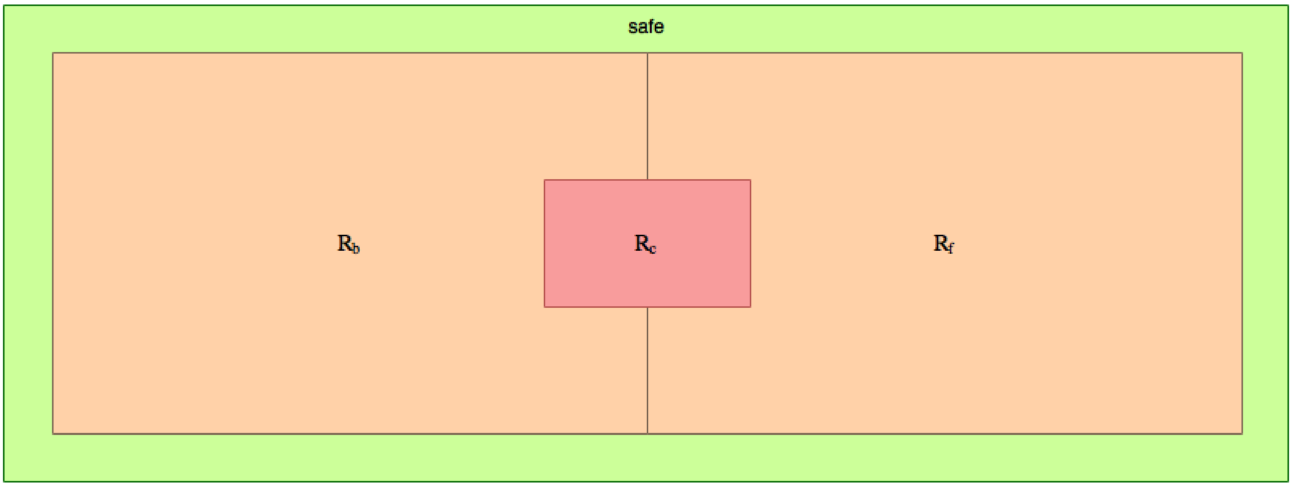


Figure 4: Occupancy grid for longitudinal model

The occupancy state of each warning region is modeled using a finite state machine, shown in Figure 5. The state begins with the region empty, and can become occupied if an object approaches. After which, the object may either continue to approach, or become relatively stationary. At this point, the status cannot change to empty before the object departs first.

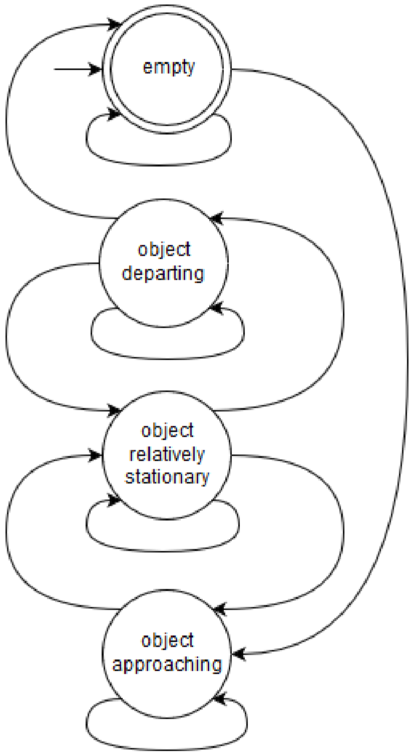


Figure 5: Finite state machine representation of region status

The SMOF tool uses a template NuSMV model that includes a model of the system, and a model of accident causality. The occupancy FSM was implemented as a NuSMV module and instantiated in two variables representing the R_f and R_b regions. The accident model used defines an accident as an approaching state for two consecutive time steps in any region.

The tool identified the seven expected warning states:

- an approach in R_f, with empty, stationary, or departing R_b
- an approach in R_b, with empty, stationary, or departing R_f
- an approach in both R_f and R_b

The next step is to suggest interventions that can be used to synthesize safety strategies. The two obvious interventions for longitudinal collision avoidance would be to brake and to accelerate. The technical specification of interventions, as for physical boundaries of regions and time step, are beyond the scope of the SMOF methodology, and are to be addressed separately. However, the interventions suggested must be implementable for any resulting safety strategies to be viable. Braking is defined, at this stage, as simply braking sufficiently such that if a frontal approach occurs it does not persist in the next time step. The accelerate intervention is defined also in a similar way, assuming suitable technical implementation.

The brake and accelerate interventions are not adequate in the case of approaching states simultaneously in the two regions. As would be expected, the SMOF tool fails to find a safety strategy. To demonstrate the tool's synthesis capability, a third intervention has been added to allow for the possibility of avoiding accidents by transferring control to other systems or the driver. This could be criticized due to the various implementation challenges with such an intervention. Notwithstanding, the transfer control intervention is assumed to be viable, and used simply to demonstrate safety strategy synthesis with SMOF.

The tool identifies four strategies. The expected strategy has been found, as shown below, while the other three strategies involve transferring control also when not approached simultaneously from the front and back.

- brake following a frontal approach
- accelerate following a backward approach
- transfer control following simultaneously both a forward and backward approach

This example demonstrates the SMOF approach and tool, where the safety monitoring rules are produced automatically according to user supplied models. The next section develops a model for combined longitudinal and lateral collision avoidance.

3.1.2 Longitudinal and Lateral Collision Avoidance

In this combined scenario, the lateral motion is accounted for in addition to the longitudinal motion modeled in the previous section, as well as the interaction between the two. This allows the following possibilities not available in the previous case:

- With awareness of the lateral dimension, collisions from the sides may also be avoided, rather than only the front and back.
- The safety distance needed can be lower if collisions can be avoided with steering, or both steering and longitudinal control.

As was previously noted, specification of region boundaries, the time step, as well as the technical implementation of interventions is beyond the scope of the SMOF approach. The occupancy grid considered is shown in Figure 6, with in addition to the R_f and R_b regions, the left and right regions R_l and R_r are included. The possibility for lower safety distance due to steering is not aimed at in this model, as it would require being able to differentiate between the left and right regions within both R_f and R_b (for example, steering left does not avoid obstacles at the front-left direction). This would require additional variables, and result in a larger state space. As was noted previously, state space explosion is a major concern with model checking.

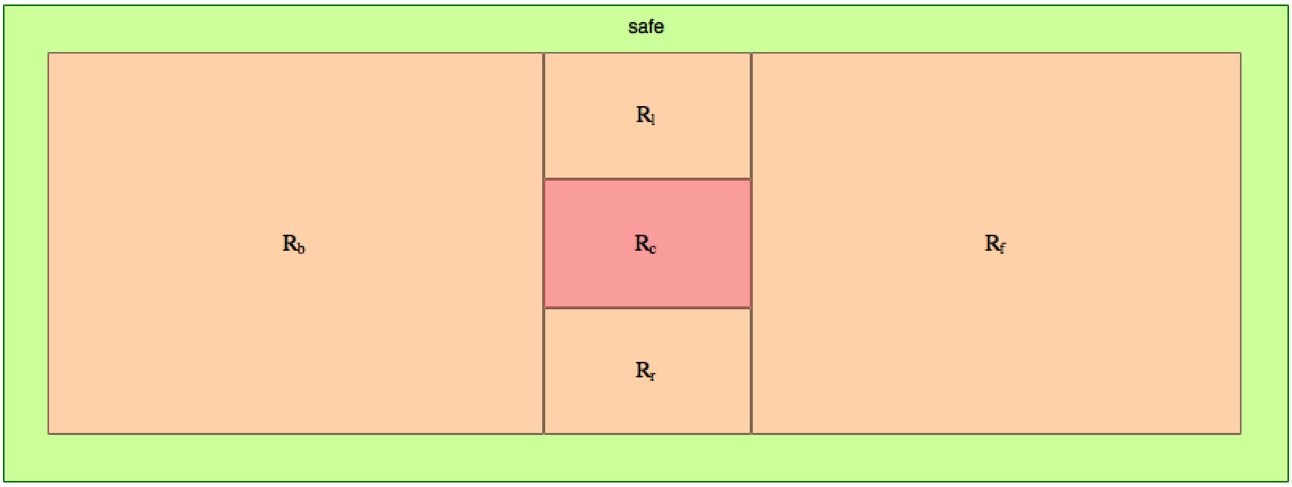


Figure 6: Occupancy grid for combined longitudinal and lateral model

Using the same set of states as before, one way to model the combined longitudinal and lateral motion is to simply enumerate all possible combinations. For the longitudinal model, one region was described using four states. Lateral motion can be described similarly using the same four states. The number of combinations is 16. The number of transitions to set would then be 16 × 16 = 256. This approach was not pursued, although it is the obvious way. Instead, the previous model in Figure 5 was adapted with one instance used for each of the longitudinal and lateral motions within a region. Then, a few constraints were imposed to model the dependencies between lateral and longitudinal states within a region.

Each warning region is described by two FSMs of the form shown in Figure 7, one for the longitudinal and another for the lateral states.

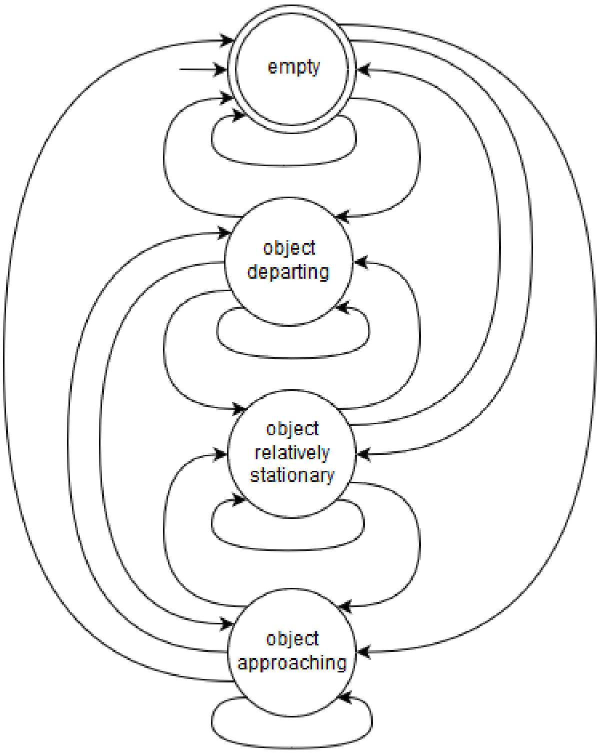


Figure 7: Finite state machine representation of longitudinal/lateral components

Compared to the previous FSM, now all transitions are possible, due to motion from the other direction. To account for the interdependencies between the two FSMs, the following constraints are added ($\iff$ and $\implies$ denote logical equivalence and implication, respectively)

- empty laterally $\iff$ empty longitudinally
- longitudinally stationary or departing without previous approach $\implies$ lateral approach
- longitudinally approaching and previously departing $\implies$ lateral approach (previous object exited and new one entered)
- longitudinally empty and previously stationary or approaching $\implies$ previous lateral depart
- laterally stationary or departing without previous approach $\implies$ longitudinal approach
- laterally approaching and previously departing $\implies$ longitudinal approach

- laterally empty and previously stationary or approaching $\Longrightarrow$ previous longitudinal depart

First the SMOF approach was attempted only with two regions, R_f and R_r. Accidents can be caused either by continuous longitudinal approach from the front, or continuous lateral approach from the right. The tool identified 51 warning states, which comprise of cases with either a frontal longitudinal approach, or a right lateral approach.

Two interventions were then used: braking and left steering. As explained in the previous section, the technical implementation of the interventions is not covered in the SMOF approach. Braking is defined as previously: decelerating sufficiently such that if a frontal approach occurs it does not persist in the next time step. Left steering is defined in a similar way with no reference to implementation. The tool finds the expected strategy:

- brake following a frontal longitudinal approach
- steer left following a right lateral approach
- brake and steer left following a simultaneous frontal longitudinal approach and right lateral approach

This successfully demonstrates the SMOF tool for a more complex application than previously seen in Section 3.1.1. Next, the two missing regions were added, namely the back and left regions, and the accident model was correspondingly extended. The SMOF tool was found unfortunately not feasible even in computing the warning states for the full model.

Although the tool has not been of assistance, this does not preclude using the SMOF architecture. Two additional interventions were added for acceleration and right steering, and a further transfer control intervention was added, following Section 3.1.1. A strategy was proposed manually and checked with NuSMV:

- brake if not longitudinally approached from the back and either approached longitudinally from the front or trapped laterally
- accelerate if not braking, and not longitudinally approached from the front and either approached longitudinally from the back or trapped laterally
- steer left if not laterally approached from the left and either approached laterally from the right or trapped longitudinally
- steer right if not steering left, and not laterally approached from the right and either approached laterally from the left or trapped longitudinally
- transfer control if trapped longitudinally, and trapped laterally

where ”trapped” refers to when having obstacles in two opposite regions, i.e. front and back for longitudinal, while left and right for lateral trapping. In order to avoid conflicts when two opposing interventions are possible, preference is given to braking and left steering.

In the preceding sections, the SMOF tool has been demonstrated for the simple scenario of longitudinal collision avoidance, and for a more complex scenario with frontal and rightwards collision avoidance incorporating two dimensional motion. Although the tool suffered from the state space problem when applied to an even more complex model, a solution could be model checked successfully, allowing for the use of SMOF safety strategies. The main components in the SMOF approach is the partitioning of the state space to identify warning states, and the use of safety strategies, that when needed enact interventions to prevent transitions to critical states. These were achieved in all the cases considered in the previous sections. The following section examines the technical implementation aspects that need to be addressed.

3.2 Safety Distance Using Time-Varying Acceleration Models

Given predicted relative acceleration $\hat{a}(t)$ between the ego vehicle and the object (e.g. a leading/trailing vehicle), and initial relative speed v_0, the task is to obtain the initial separation x_0 that avoids a trajectory reaching 0 distance, if this is possible for $\hat{a}(t)$ and v_0. The main result is derived in Appendix A, and stated as:

$$\int_{x_0}^{0} \hat{a}(t)dx = -\frac{1}{2}v_0^2 \tag{1}$$

Equation 1 gives a relation between initial velocity and required separation for time-varying acceleration. An analytic solution requires an expression for the relative acceleration $a(t)$ in terms of x, which is not available. Alternatively, a solution can be found numerically as will be shown in the next section. For the constant acceleration case the equation reduces to

$$x_0 = \frac{v_0^2}{2a_c} \tag{2}$$

where a_c is the constant acceleration.

To demonstrate a numerical approach to finding the needed separation distance in terms of initial relative speed, consider that the object moves towards the ego vehicle according to predicted acceleration curve $a_i(t)$, which is initially positive but decreases as the other object reacts to the situation, and consider that the ego vehicle is capable of acceleration profile $a_s(t)$ in collision avoidance. For this example, assume that the two functions are of the form

$$a_i(t) = R\left(\frac{1}{2} - tanh(t)\right)$$
$$a_s(t) = -k_s\, R\, tanh(\alpha_s\, t).$$

R is the expected possible acceleration range in m/s^2. The hyperbolic tangent has been used simply due to its shape containing transient and steady-state regions. A value of 4.3403 corresponds to 0-100km/hour in 6.4 seconds. k_s is a scale factor representing the amount by which the magnitude of the ego vehicle's reaction is larger than the object's reaction, and α_s represents the relative quickness in responding to the situation.

Let us use: $R = 0.5, k_s = 1.5, \alpha_s = 0.5$ as example. The resulting functions are shown in Figure 8, with also the relative acceleration $\hat{a}(t) = a_i(t) + a_s(t)$ plotted.

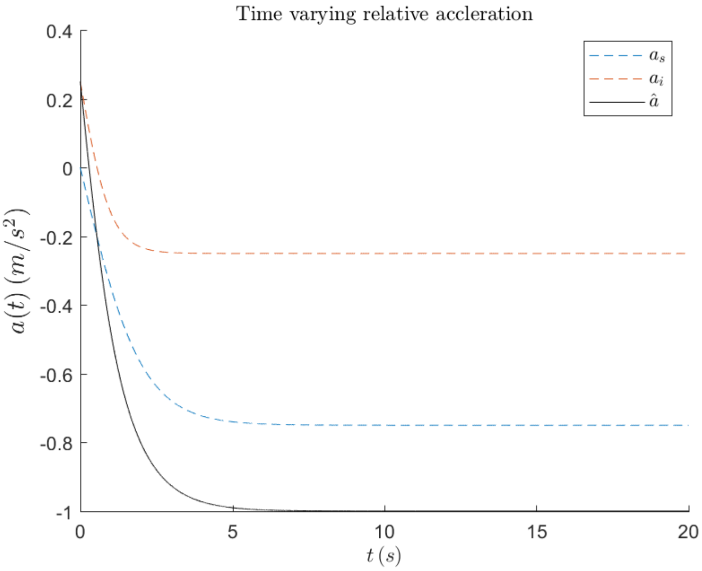


Figure 8: Time varying accelerations of the two objects

These simple models contain two phases: a transient response and a steady-state response. This assumes that the steady-state (of acceleration) continues indefinitely, implying that no reduction in deceleration occurs, and that collision avoidance efforts can persist as long as needed. However, in practice the relative acceleration may eventually go to zero, as would be the case with braking leading to a stopped vehicle.

The relative distance trajectory $x(t)$ is found by integrating the acceleration twice:

$$\begin{aligned}\frac{d^2x(t)}{dt^2} = \frac{dv(t)}{dt} &= \hat{a}(t)\\ v(t) &= \int_0^t \hat{a}(\tau)d\tau + v_0\\ x(t) &= \int_0^t \int_0^\gamma \hat{a}(\tau)d\tau d\gamma + v_0 t + x_0.\end{aligned}$$

where v_0 is the initial approach (relative) speed. For any v_0, we can find the least separation (least negative x_0) that ensures $x(t) > 0, \forall t > 0$. By iterating over different approach speeds we can obtain a relation between v_0 and x_0.

A speed of 40km/hour corresponds to 11.12m/s. Let us consider approach speeds in the range [0.1,11.12]. Figure 9 shows the least separation solutions for the different initial speeds v_0. Notice that trajectories are always non positive for all the curves. Arriving at zero separation indicates that the initial separation is minimal. Due to the prolonged steady-state acceleration seen in Figure 8, the trajectory reverses after reaching zero separation (objects just touch then separate). The trajectory after zero separation is not of relevance for collision avoidance, and can be ignored.

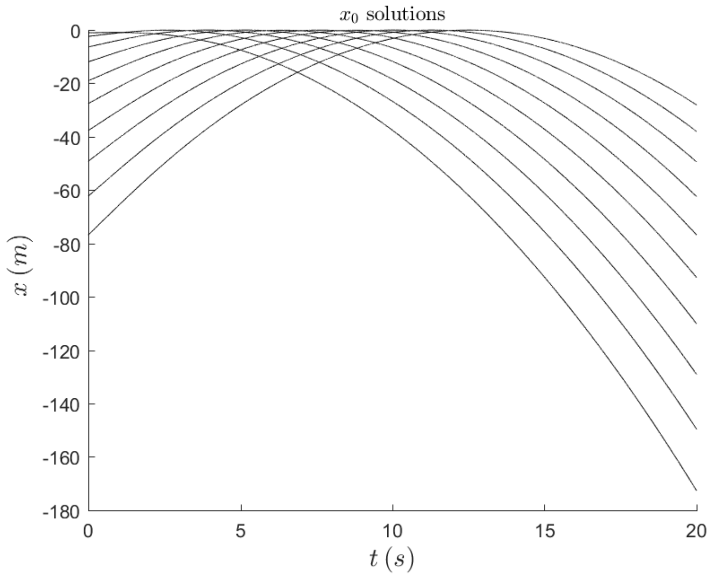


Figure 9: Solution trajectories for different initial speeds v_0

Finally a relation between v_0 and x_0 can be obtained, as shown in Figure 10.

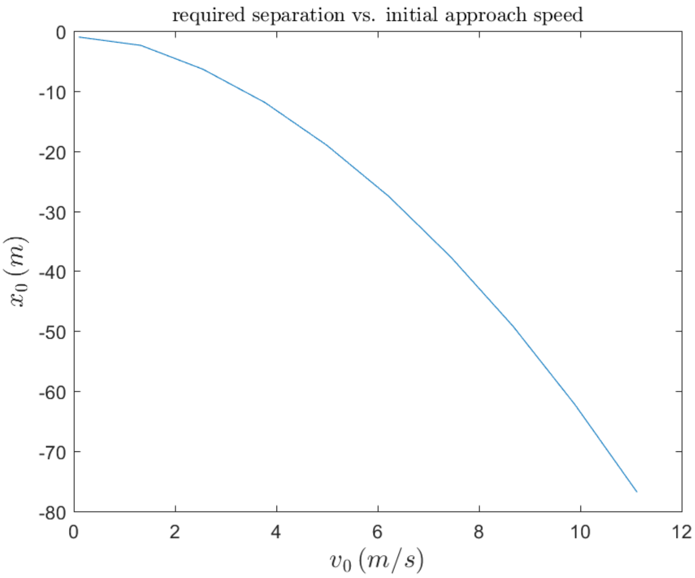


Figure 10: v_0 to x_0 relation

Let us now consider the constant acceleration case to find how the required initial separation distance compares to the result in Figure 10. It is not apparent how one should select a value of a_c in Equation 2. As example, consider the steady-state, average, median, and 25th percentile accelerations during the initial 10 seconds. Figure 11 shows the results for these constant estimates obtained by invoking Equation 2, together with the time-varying acceleration result.

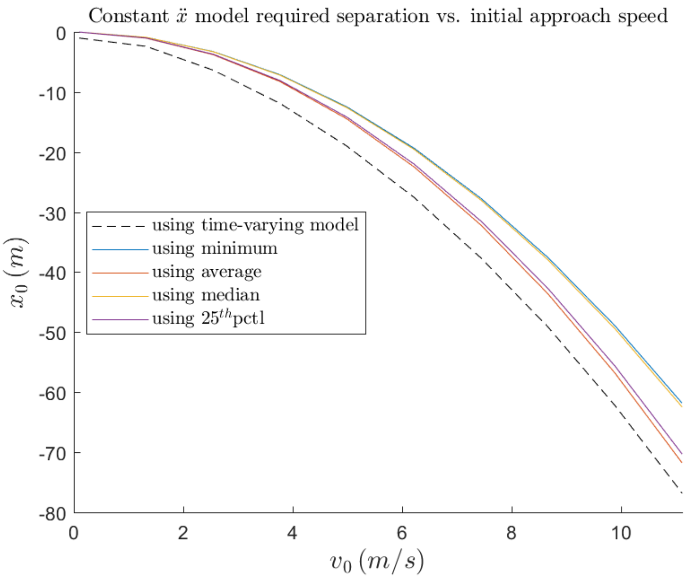


Figure 11: v_0 to x_0 relation for constant acceleration

Optimistic separation distances were found for all the constant acceleration cases. The steady-state (minimum) acceleration yielded the most optimistic separation, as would be expected. The median is the acceleration at 5s in Figure 8, which is very close to the steady state. The 25^{th} percentile is 2.5s in the figure. The average is the most conservative but still gives optimistic separation distance. These results indicate the significance of the initial period where the approach speed is high but little deceleration is taking place.

In the approach presented in this section, collisions can be avoided if the separation is greater than the obtained safety distance value. This gives the distance boundary needed in the SMOF approach, for interventions to be viable. The calculation uses relative motion in one direction. The boundary can be constructed along any direction of interest provided the following are available along that direction:

- Current speed of approach between ego vehicle and object
- Knowledge of ego vehicle's acceleration capability
- Expectation for future acceleration of the object

Once the object is within this physical boundary, interventions can be triggered according to the safety strategies used.

Next in this thesis, a case study is presented which employs simulation to test and demonstrate SMOF safety monitoring for collision avoidance. The main motivation in this work is safety assurance for learning systems. The study will deploy a vision-based autonomous driving function, along with safety monitoring using the approach discussed thus far.

4 Case Study: Safety Monitoring for Vision-Based Vehicle Control

This section presents an implementation of safety monitoring and vehicle control toolchains within a simulation environment. A deep learning system will be used to drive a simulated vehicle, and experiments will be performed to investigate safety with malfunctioning vehicle control.

Previous sections have discussed the safety concerns with learning systems in autonomous driving, and the difficulty of verifying their safety properties. Instead, an architectural safety monitoring approach is pursued in this thesis. Section 3.1 applied the SMOF safety monitoring approach to collision avoidance, and Section 3.2 discussed an approach for obtaining the distance boundary needed for interventions to be viable. This section studies a case where a learning system controls the vehicle, and develops a framework for supporting safety investigations. The scenario consists of a vision-based speed control system for longitudinal motion, along with a collision avoidance safety system to mitigate risks due to faults in the learning system.

Driving simulators are of help in this study, both for generating synthetic data, and for testing the effectiveness of safety systems. The experimental approach depends greatly on the capabilities offered by the driving simulator, and computational resources available. This topic is addressed next, after which the control toolchain and the experiments are presented.

4.1 Driving Simulation

The motivation for using driving simulators in this study is that they can offer the possibility to experiment with driving scenarios that are difficult, and costly, to create in a real environment. For example, factors that make real-world experiments prohibitive are: access to vehicles and needed hardware, ensuring tests are safe, needing to perform system identification or measure relevant parameters, setting up prototype autonomous driving systems, the time to set up and conduct experiments, and also the difficulty of setting up controlled experiments when relevant variables cannot be controlled.

With driving simulators, it is possible to work at a higher level of abstraction focusing more closely on the learning and safety systems at the functional level. The driving environment can be interfaced with system models as part of a software-in-the-loop or model-in-the-loop setup, which helps in evaluating functional safety.

Specifically for this vision-based speed control study, the elements needed in a driving simulator are the following.

- **off-line data export** The quantities that are relevant to training need to be available for off-line use. These include: image frames, distance traveled by ego vehicle, distances to other vehicles of interest, acceleration and brake pedal positions, simulation time, as well as any other interesting information for training.
- **closed loop control** To pursue control in a dynamic virtual world environment, there needs to be a possibility for using the above quantities in a closed loop setup where data is processed and control is sent to the vehicle during simulation. This can require synchronization between several simulation software components and user models.
- **adequate virtual world** A basic level of scientific accuracy is needed in the simulation environment. This includes the physics of the vehicles within the environment, and also the visual appearance of virtual world.

Three virtual driving simulators have been tried to assess their potential to support vision-based control studies: IPG CarMaker, TESIS DYNA4, and TASS PreScan. These simulators are discussed in the following subsections.

4.1.1 IPG CarMaker

CarMaker is a virtual testing platform for passenger cars and light-duty vehicles. The simulator can be interfaced with the MATLAB/Simulink environment using CarMaker's provided templates that include S-function blocks. The user's control strategies can be implemented either by manipulating Simulink signals or by creating custom modules. CarMaker can be run either stand-alone, or in cosimulation with Simulink.

CarMaker offers possibilities for offline data export. After simulations, image frames can be obtained from the animation tool either by exporting video frames, or by exporting a video. However, timestamps are not available if the image frames are exported directly. The data plotting tool, IPGControl, can be used for exporting simulation variables into a csv file. To capture data, IPGControl needs to be configured before simulations.

Except for a camera feed, all our needed quantities can be used directly in a closed loop setup within Simulink. IPG offers the Video Data Stream (VDS) extension package which allows sending video out of the animation tool via TCP, and includes some additional features for adding camera noise, motion blur, etc. This feature was not included in my initial trail, and had to be requested specifically. Using the additional visual features greatly reduces simulation and movie export speeds.

Once a camera is configured within CarMaker, the animation tool, which is a separate program, sends out the video stream via TCP. After making several attempts to receive the video in Simulink using different approaches, at best the received images contained some distortions/artifacts, and the animation tool did not synchronize successfully with Simulink.

As for other factors, CarMaker meets the needs adequately with regards to scientific accuracy of the simulations and visual appearance of the virtual world.

4.1.2 TESIS DYNAware DYNA4

DYNA4 is simulation framework aimed at supporting vehicle development processes, and runs within Simulink, allowing for signals to be accessed and manipulated. Users can also configure and add new Simulink modules from within DYNA4.

To export data for offline use, one can select the needed quantities before the simulation, and then a mat file with needed variables is produced after its completion. DYNA4 stores the data needed to recreate the animation after simulations. Video can then be played back or exported, for any completed simulation.

Data signals are available for closed loop control, however, in the commercially available DYNA4 product camera pixels are not accessible in Simulink. TESIS kindly allowed me the opportunity to test a development version which includes this functionality. To obtain the camera feed for closed loop control, the camera is configured within DYNA4, and a receiver block is added to the corresponding Simulink model of the simulation project. A blocking communication is setup between the animation tool and Simulink, allowing for pixel data be accessed simply.

DYNA4 offers a visually appealing animation which can support our vision-based control investigation, as well as an adequately accurate virtual world.

4.1.3 TASS PreScan

Prescan offers a simulation framework for Advanced Driver Assistance Systems (ADAS) and Intelligent Vehicle (IV) systems, targeted more for the simulation of sensors and the virtual environment, rather that vehicle dynamics. Simulink is also used as a simulation engine. After the scenario is configured in a GUI tool, a Simulink model is generated. ADAS functions can be implemented simply by modifying the generated models. Also, a wide variety of demos are provided featuring different ADAS functions, allowing for users to substitute their own algorithms. Sensor technologies such as radar, laser, camera, ultrasonic, GPS and C2C/C2I communications, are supported. A physics-based camera model is also available, which accounts for a complete pipeline from light sources to optics and final image.

One notable aspect with PreScan is its strong integration with MATLAB and Simulink, allowing for a lot of possibilities for accessing signals and controlling various elements of the virtual world, e.g. all simulation vehicles, pedestrians, sensors, and even lightposts. RGB pixels are also available conveniently in Simulink, facilitating camera-based ADAS function development. Furthermore, simulation runs can be automated with MATLAB scripts, which allows for performing tests with changed parameters and collecting results for processing, or exporting the data. Video can also be exported in different formates using a GUI visualization tool.

Simple vehicle dynamics models are available in PreScan, which can be freely modified in Simulink. For more complex dynamics, PreScan allows for integration with other tools, such as CarSim and veDYNA.

4.2 Vision to Speed Control

Since the focus of this work is on learning systems in safety critical automotive applications, an example system is considered that controls the speed of the ego vehicle using camera images of the road. This study is delimited to longitudinal speed control, although a more comprehensive model was developed in Section 3.1.2.

The scenario consists of one leading vehicle fixed at some distance ahead of the ego vehicle. The task of the learning system is to detect the leading vehicle, and feed this information to a controller which then controls the gas and brake pedal positions. In such a setup, the output of the learning system can be readily compared to ground-truth for analysis.

A deep neural network is used to detect the leading vehicle and provide bounding box information to be used for speed control. Since the distance to the leading vehicle is directly related to the bounding box size, a simple power-law function can be used to obtain a control signal, which then determines the gas and brake pedal positions.

Using such a setup where the learning system outputs intermediate bounding box information has some benefits:

- This corresponds to a logical subproblem to the main speed control problem. If an end-to-end approach were used, the learning system would instead map pixel data to gas/brake pedal control. As discussed in Section 2.1, logical subsystems of end-to-end systems cannot be inspected, which allows a possibility for faults to be masked.
- The ground truth is simple to determine, and depends only on images. In fact, the detector's performance can be assessed intuitively by viewing the resulting image sequence. For an end-to-end system, in contrast, the ground truth may depend on other factors such as the current speed and acceleration.
- Since the scenario involves only one leading vehicle, the bounding box is sufficient information for a control strategy.

- Many pretrained object detectors are available, allowing for the possibility of experimenting with existing state of the art networks.

The function of the learning system in this study is to identify the leading vehicle in camera images containing the road, environment, scenery and other traffic objects. For such a problem, object localization as well as classification are needed. One interesting approach is to use a region-based convolutional neural networks (R-CNN), which includes a region proposal network and an object classification network. A recent advancement in using this architecture is the Faster R-CNN [23, 24], where the classification network shares full-image convolutional features with the region proposal network, which enables nearly cost-free region proposals. The experiments presented in the following section employ a Faster R-CNN trained for localizing and classifying the leading vehicle.

4.3 Experiments

The experimental setup considered consists of an empty stretch of road with one fixed vehicle located some distance ahead of the ego vehicle. A camera is placed behind the front windshield of the ego vehicle, and a Faster R-CNN system processes the images in order to control the speed. This simple experimental setup allows for testing only frontal collision avoidance. Extensions could be made in order to study more comprehensive scenarios, especially where the safety strategies are complex. However, this was not attempted in this experiment. More complex monitoring scenarios would help to test safety monitors, but contribute little to a safety investigation of the fundamental concerns with learning systems.

The strategy computed for the SMOF safety monitor in the longitudinal case (see Section 3.1.1) was to trigger a braking intervention when the warning region is in an approach state. This occurs as soon as when the distance to the leading vehicle becomes less than the safe distance. A key point is that the safety monitor can trigger interventions, rather than passively monitor, or only block actions, and technical specifications of interventions and region boundaries allow for guaranteeing collision avoidance if the specifications hold. The monitor calculates the safe distance dynamically based on the ego vehicle's current speed and the relative acceleration, using the approach presented in Section 3.2. Since in this simplified setup the leading vehicle is stationary, only the ego vehicle's deceleration curve is needed in the calculation.

Figure 12 shows a block diagram of the control setup used for experiments. Camera images are used to control the speed of the ego vehicle, with a safety monitor installed to intervene when a warning state is reached. Ground truth information is used for safety monitoring, as this avoids introducing additional points of failure extraneous to the learning system under study.

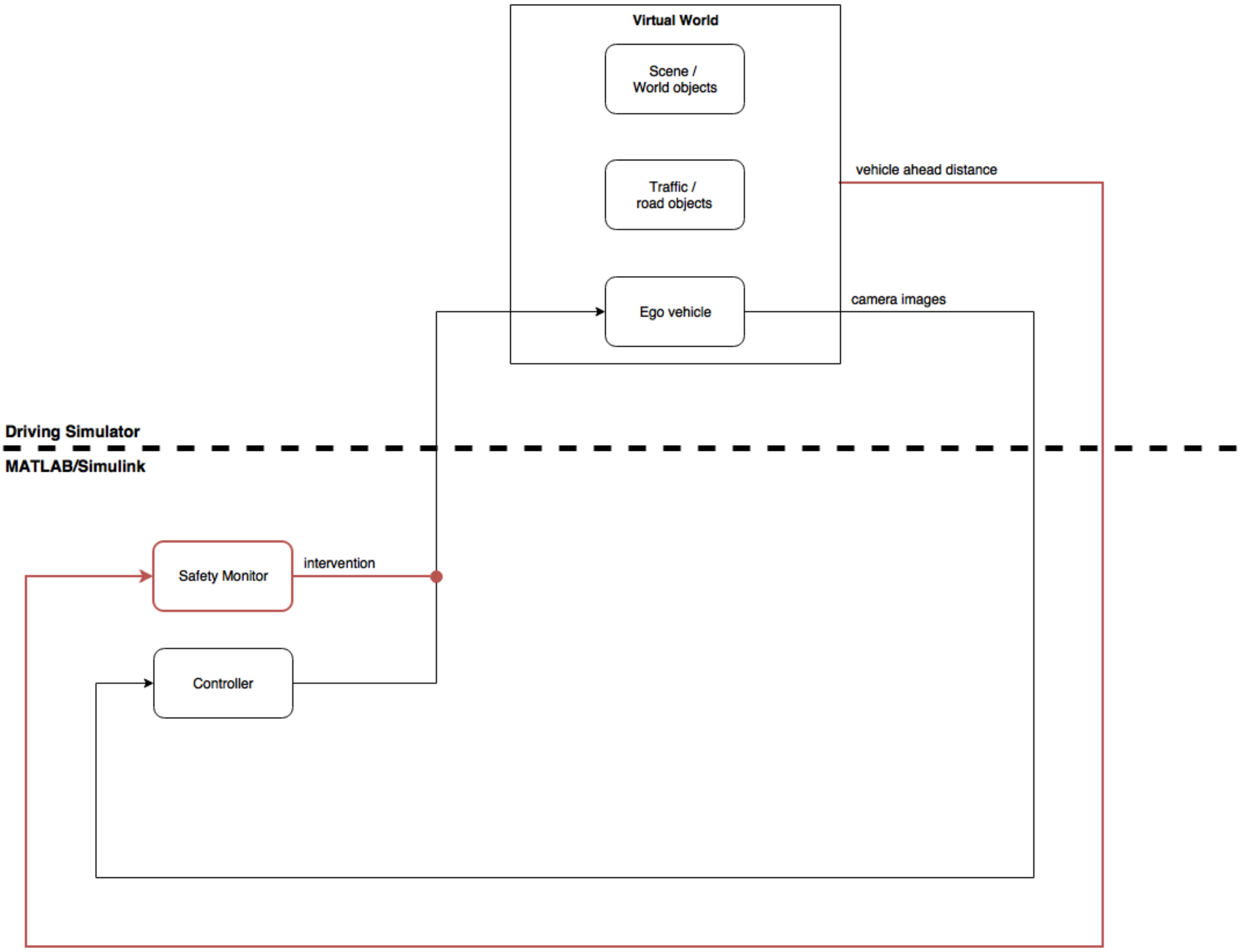


Figure 12: Control loop with safety monitoring

The goals for experiments are twofold: firstly to simulate the safety monitoring and vision-based control toolchains and demonstrate collision avoidance in cases of hazardous speed control; and secondly to discuss an example trustworthiness indicator for the learning system, which can help warn in advance before severe safety interventions are needed. This is related to the Safe Fail strategy mentioned in Section 2, in which the system does not attempt to offer an output in case of low confidence, or closeness to the decision boundary, and the operator makes the decision instead. However, in the speed control context, it is not possible to expect for the driver to immediately take over. The trustworthiness indication approach in this thesis allows for the driver to be warned regarding the learning system in a low confidence scenario.

4.3.1 Design of Experiments

The Neural Network Toolbox in MATLAB 2017a offers an implementation of the Faster R-CNN deep learning system, including a class object and a function to train the detector. Although code from the Faster R-CNN authors is publicly available [23, 24], The implementation in the toolbox was opted for, due to ease of use and possibly quicker development time. Flexibility in regard to tweaking and tailoring algorithms for better performance is not a primary goal for this project.

The virtual world contains a straight stretch of road with some trees along the sides, and one vehicle fixed 300m ahead of the ego vehicle. The only variable that affects the camera images is the ego vehicle's position along the road. Therefore, by simply passing the camera along the road it is possible to capture images at any needed distance resolution.

This scenario avoids needing to perform image processing online. Image processing and driving simulation are computationally expensive tasks. At the time of conducting the work for this thesis, an adequate computational setup was not yet available, and the available GPU was not supported by MATLAB. This greatly limits the achievable frame rate as the detector is run on CPU instead. For 500x375 images, the detector computes at around 0.15 fps on the available machine, rather than 5 fps in the Faster R-CNN author's papers. For a closed loop setup, the detector would need to run in Simulink as an extrinsic function, since code generation for the Faster R-CNN object detector class is not supported in Simulink. The 0.15 fps would therefore not be improved. In addition, driving simulators are also computational expensive which further limits the speed. A closed-loop setup with such computational limitations would hinder the prototyping and evaluation of systems.

On the other hand, if a more complex scenario with moving objects were used, the camera images would depend on several variables that would be difficult to control, necessitating a closed-loop setup.

As online image processing is not pursued, any of the previously mentioned virtual vehicle simulators could be used: IPG CarMaker, TESIS DYNA4, and TASS PreScan. CarMaker was selected primarily because it was made available before the other two. PreScan, along with a suitable computer setup for simulation was available only after completing the work for this thesis. The test vehicle chosen is the demo Tesla S model example provided in CarMaker, which uses an electrical powertrain, and a camera was added behind the front windshield.

The ego vehicle was passed along the road at low speed and the frontal camera's video stream was exported, and then processed to produce images at 0.1m resolution. These images are used for training and testing, as well as to precompute detections at 0.1m intervals along the road for access during simulations. The labeling process requires manually drawing the bounding box of the leading vehicle in each image. This process was automated with the help of the MATLAB Ground Truth Labeler app, which can interpolate between manually made labels at key frames.

Due to computational limitations, small 128x228 images were used. Training of a Faster R-CNN is a four step process. In the first step, the region proposal network (RPN) is trained. In the second, a separate detection network is trained using the proposals from step 1 RPN. In the third step, the detector is used to initialize the RPN, but only the layers unique to the RPN are trained. Finally in the fourth step, the layers unique to the detector are trained. One problem that occurred frequently in my training attempts is that no positive or negative samples may be found in the fourth step, when the proposals from the third step are used. At far distances from the leading vehicle, less pixels are used to represent the vehicle, which presents a challenge in training. The training process could complete only if far distances are excluded.

The leading vehicle is 300m ahead along the road. Detectors for two different distance ranges were investigated:

1. A detector was trained for the distance ranges 250-270m and 280-295m, using 351 training images.

2. A detector was trained for the distance range 270-295m, using 251 training images.

4.3.2 Safety Monitoring

For calculating the safety distance dynamically at run time, the deceleration curve of the ego vehicle, and the expected acceleration curve of the leading vehicle are needed. Modeling the behavior of the leading vehicle for collision avoidance scenarios is an interesting topic for dedicated research. Discussed in [22] is the concept of decision functions, in which dynamic models of the two vehicles and the expectation of their future actions are used as the basis for decisions. The scenario in this simple study uses a stationary leading vehicle, and the safety monitor is designed based on this operating condition. Also, the braking curve can vary with different road conditions. In this study, the effect of different road conditions is not investigated, and could be also a topic for further work.

To determine the braking curve of the Tesla demo vehicle model, braking tests were performed on the same road to be used in the experiments. Figure 13 shows the braking behavior for different starting speeds until the vehicle is stopped. It takes approximately 1.5 seconds to reach maximum deceleration, which continues until the vehicle is almost stopped. As shown in the figure, this initial phase is the same regardless of the starting speed. Inspecting the distance traveled in the final phase when going back to zero acceleration shows that it is only a few centimeters (2-4 cm). For safety monitoring, only the initial phase and the maximum deceleration phase will be used in safety distance calculation.

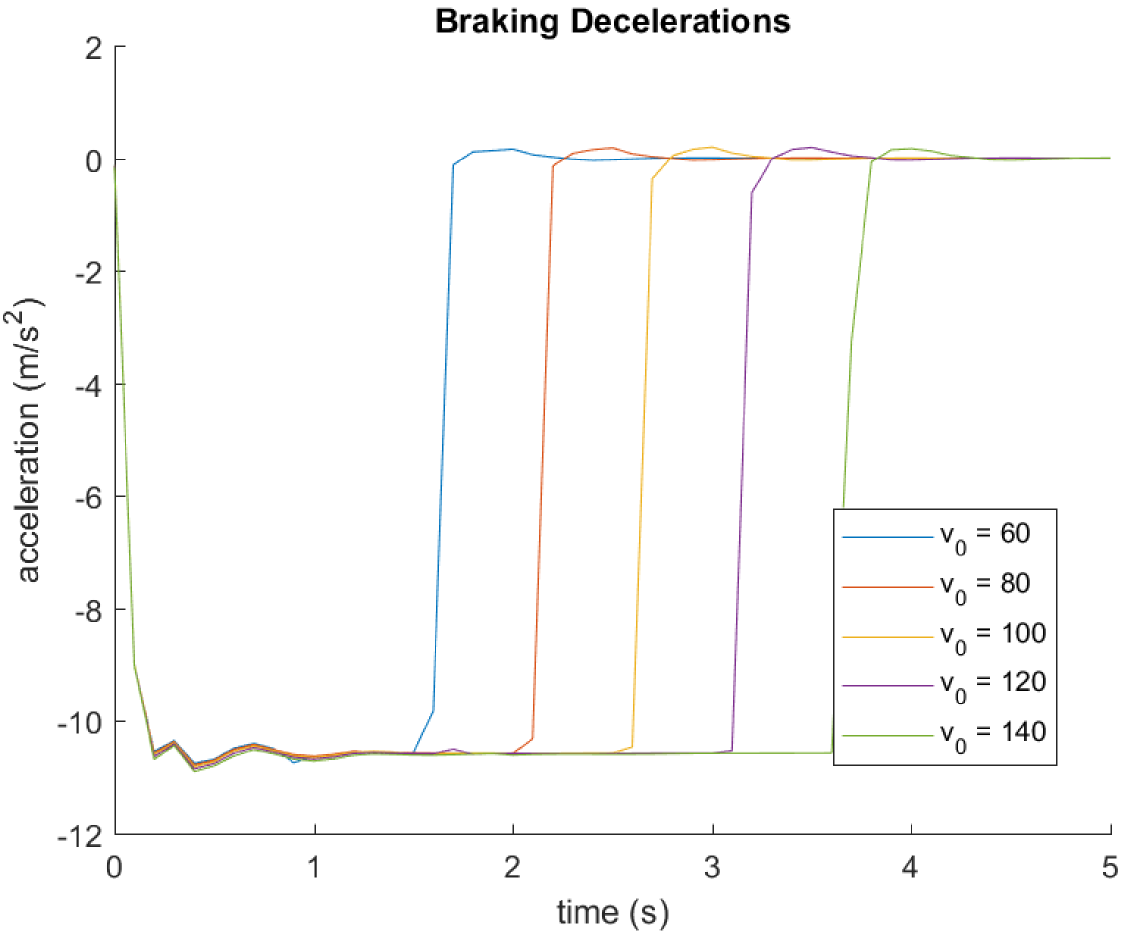


Figure 13: Braking deceleration for CarMaker test vehicle

4.3.3 Control System

The speed of the ego vehicle is controlled via the gas and brake pedal position signals in the CarMaker Simulink model. The Faster R-CNN system outputs bounding boxes (this locates the object within the image) and corresponding scores of detected objects classified as the leading vehicle. The area of the highest scoring bounding box is used as an indirect indicator of the leading vehicle's distance. This applies since the vehicles are configured to be in the same lane. The control law then used to obtain an intermediate signal is in the following form

$$u = A^{\alpha}$$

where A is the bounding box area (width × height pixels) divided by the image area, and α is a tunable parameter that affect the cautiousness of the resulting control. The resulting signal u takes values from the 0-1 range. The gas and brake pedal positions can be derived from u according to:

$$brake = \begin{cases} 2\,(u-0.5) & u > 0.5 \\ 0 & u \leq 0.5 \end{cases}$$

$$gas = \begin{cases} -2\,(u-0.5) & u < 0.5 \\ 0 & u \geq 0.5 \end{cases}$$

which results in braking with no gas when u is larger than 0.5, and gas with no brakes otherwise. Both gas and brake pedals are not engaged when $u = 0.5$.

This control strategy is implemented in Simulink as shown in Figure 14, with also a moving average included to smooth out u. This is helpful especially for malfunctioning detector behavior with momentary object appearance or disappearance. The window size used is 200 samples. The simulation is configured for variable step size in the CarMaker Simulink interface, but inspection shows that 200 samples correspond to about 0.2 seconds of simulation time.

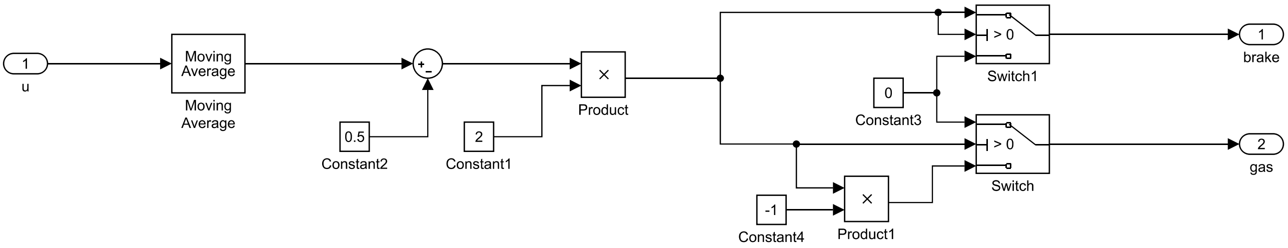


Figure 14: Simulink control system for brake and gas pedal

Simulations were performed using ground truth bounding box information in order to calibrate the α parameter, and a suitable value found was 0.1. Figure 15 shows the resulting vehicle speed and the corresponding control signals. The vehicle was initialized at 100km/hour (ca. 27.8m/s). Figure 16 shows the distance trajectory, with the accident distance marked.

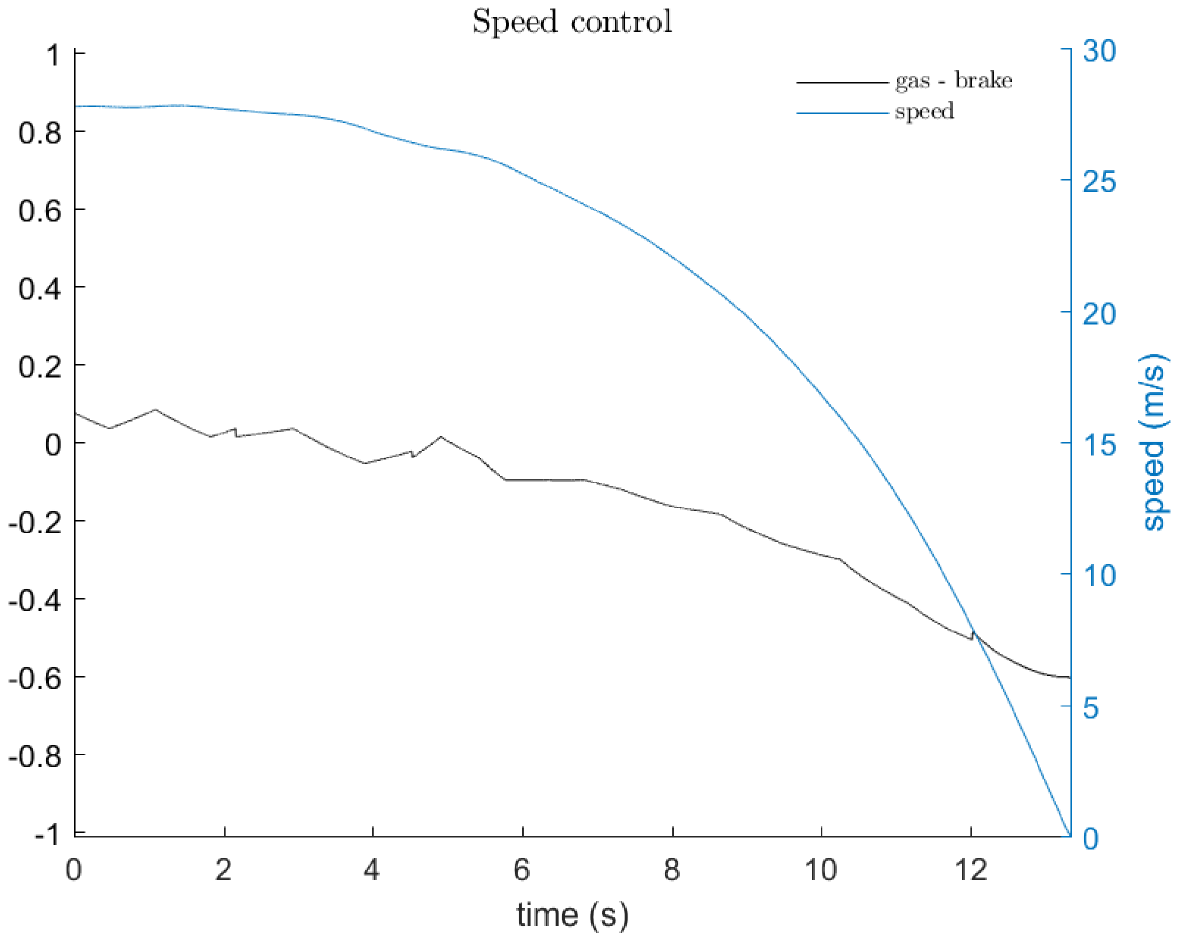


Figure 15: Speed control for $\alpha = 0.1$, and ground truth detections.

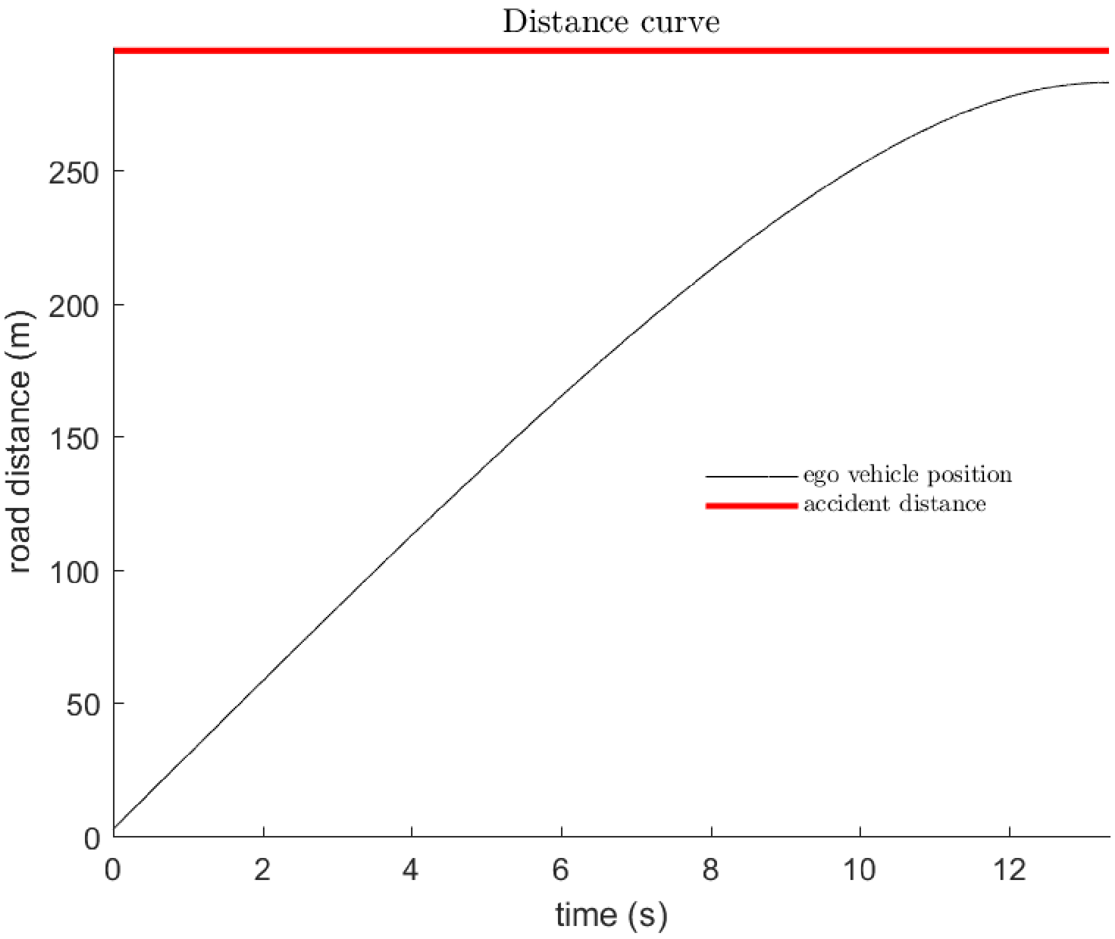


Figure 16: Distance curve for $\alpha = 0.1$, and ground truth detections.

4.3.4 Evaluation Metrics

Two detectors for different distance ranges will be tested. The training difficulties were previously noted with regard to far distances, and training had to be restricted to closer distances for it to succeed. The first detector was trained for 250-270 and 280-295m, using 351 training images. Training did not yield good performance even within this training set, but this presents an interesting case for safety studies, as will be discussed. The second detector was trained for the easier 270-295m, using 251 training images. Much better performance could be achieved for this case within the training range, but this limited range helps to illustrate how a learning system can perform well for some data that was represented in the training, but cause hazardous behavior with unseen data.

The metric we will use for evaluating the performance of the learning system is the Intersection over Union (IoU) which is defined as

$$IoU(A, B) = \frac{area(A \cap B)}{area(A \cup B)}$$

where A and B are the ground truth and detector bounding boxes. When multiple detections are produced by the detector, the highest scoring detection is used. The IoU does not distinguish between false positives and false negatives, but is a common metric for evaluating detectors and is sufficient given that false positives would trigger unwarranted braking.

Another measure that is of interest in this research is related to the similarity/dissimilarity of incoming camera images with the training images. Since the network is trained entirely using synthetic data from CarMaker, the influence of the training data can be controlled and accounted for. In addition, the random seed for the training process was fixed to insure reproducibility of the training results.

An example measure which utilizes SURF [25] features is used in these experiments. The SURF features of the highest scoring interest point in images are computed and used as descriptors. The disparity measure used in the experiments computes the Euclidean distance between the computed features of the runtime images to the nearest neighbor in the training set. SURF was chosen due to its speed advantages and scale invariance property. Other methods are not investigated in this study, but could be a topic for further research.

4.3.5 First Detector

This detector was trained with 351 128x228 images for 250-270, and 280-295m distances. As previously mentioned, the less pixels used to represent the vehicle, the more difficult is the learning task. Attempts at training with the full 0-300m distance range have failed, and hence training was focused on closer distances where the leading vehicle is larger in the image. This detector was trained for 25 epochs.

Figure 17 shows the IoU and disparity for the first detector, with the two distance ranges shaded. The measurements were filtered using a moving average of the same length as the moving average used for u, and 1 standard deviation curves are also plotted. The disparity is zero within the training intervals, as the incoming data was represented in the training set. On the other hand, distances away from the training intervals yielded high disparity, which decreases closer to the intervals. Ideally, the disparity measure should vary smoothly with distance, but this was seen not to be the case with using SURF features of a single interest point. Nonetheless, the metric helps to demonstrate that a disparity measure can be used to judge how well new data was represented in training. The detections were completely erroneous in the first trained interval, 250-270, as indicated by the IoU, while some detection capability is seen in the second trained interval 280-295.

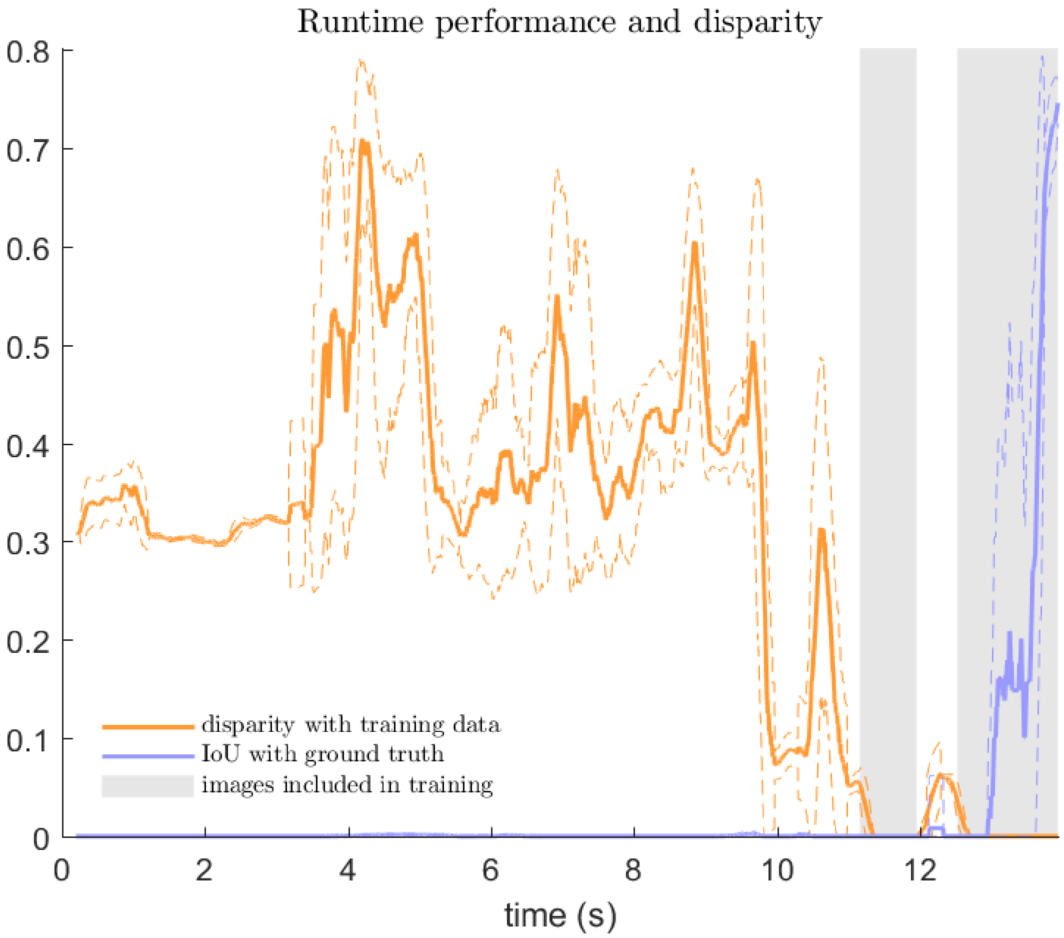


Figure 17: IoU and disparity mean and ± 1 standard deviation curves for the poor performing first detector. Notice low disparity close to and between training ranges.

Figure 18 shows the trajectory of the ego vehicle along the road. An intervention was triggered, which successfully brought the vehicle to a halt before colliding with the leading vehicle.

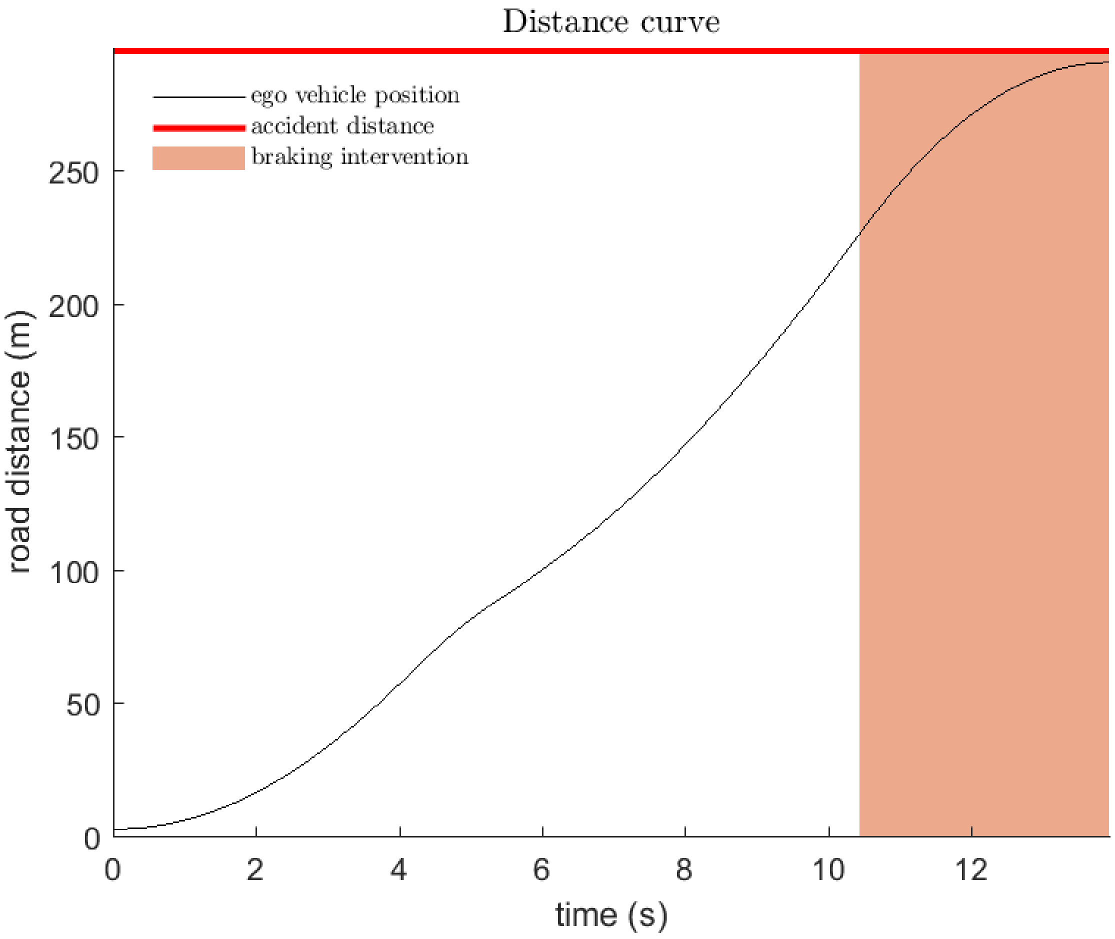


Figure 18: Distance trajectory along the road for the first detector, with braking intervention triggered

The speed control is shown in Figure 19, with the vehicle initialized at the start of the road with 0km/hour. A false positives period (vehicle detected as at close distance when it is actually farther away) caused braking starting at around 4 seconds. This is followed by a period where the detector is making some detections which are not accurate, as indicated by IoU in Figure 17. Afterwards, false negatives (vehicle detected as far away when it is actually near) caused an approach with high speed, which necessitates an intervention well before the trained intervals are encountered. In fact, the controller requested braking only towards the end of simulation when a good IoU is obtained.

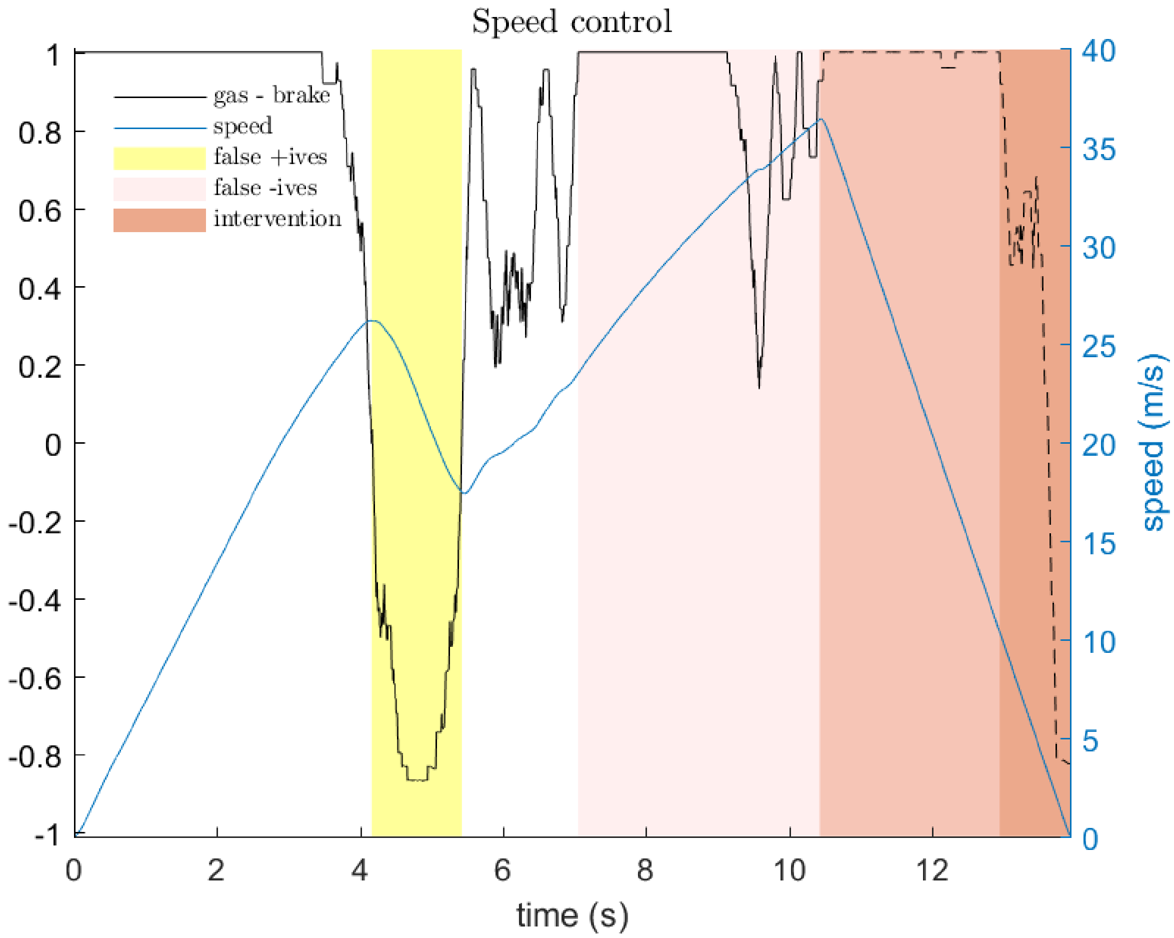


Figure 19: Speed control for the first detector. Dashed lines indicate overridden signal. False +ives indicate detecting a very near vehicle when it is in fact farther away, while false -ives indicate that the vehicle is detected as far away when it is in fact near.

4.3.6 Second Detector

In light of the poor performance of the first detector, especially in its first trained distance range, a second attempt was made to train another detector with closer distances, making the learning task less difficult. The distance range was limited to 270-295m, and a detector was trained for 25 epochs with 251 training images.

Training indicated some success in learning within the trained distances. Two experiments are performed to illustrate both non-malfunctioning behavior where safety interventions are not required, and malfunctioning behavior that triggers the monitor interventions. In a low speed experiment, the needed safety distance does not exceed the trained range, which allows the controller an opportunity to operate within the learning system's trained distance range. Another experiment with higher approach speed was then performed, and a monitor intervention was necessary, since the learning system malfunctioned outside its trained distance range.

Low speed experiment
For this experiment, the setup is initialized such that the learning system is allowed an opportunity

to function within its trained distance range. Approaching at 70km/h would trigger an intervention at the start of the trained distance range, precluding the possibility of observing the controller's performance. The detector was seen to suffer from a false positives problem that brings the vehicle to a halt prematurely if initialized at the start of the road. To address these problems, the vehicle was initialized at 250m along the road with 35km/h.

Figure 20 shows the IoU and disparity for the second detector. The measurements were filtered as before with detector 1 with the same moving average, and ±1 standard deviation curves are plotted. As indicated by the IoU, much better performance is attained within the learned distance range. The disparity is zero within the training intervals, as before, and is higher away from the trained data. The trend in disparity tends to be inversely related to the IoU performance measure.

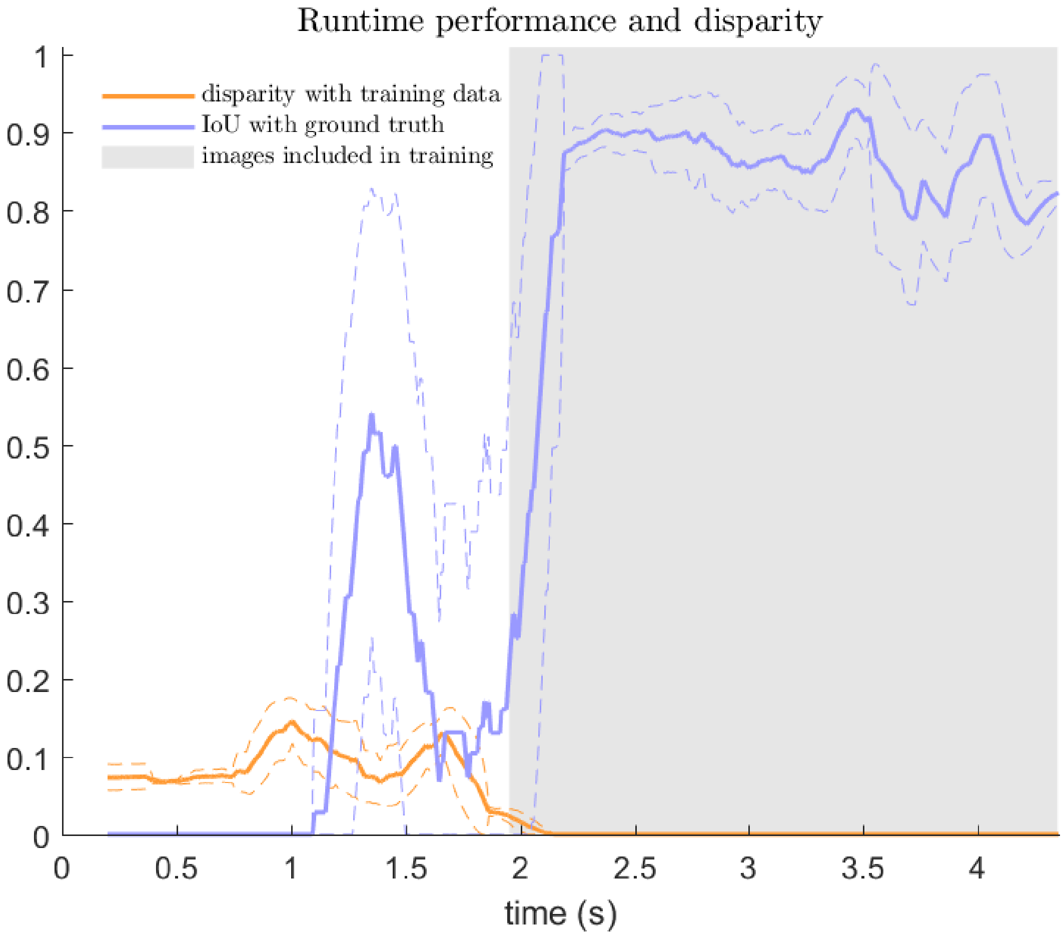


Figure 20: IoU and disparity mean and ± 1 standard deviation curves for the second detector low speed experiment.

Figure 21 shows the trajectory of the ego vehicle along the road. An intervention was not required, as the controller successfully brought the vehicle to a halt before colliding with the obstacle.

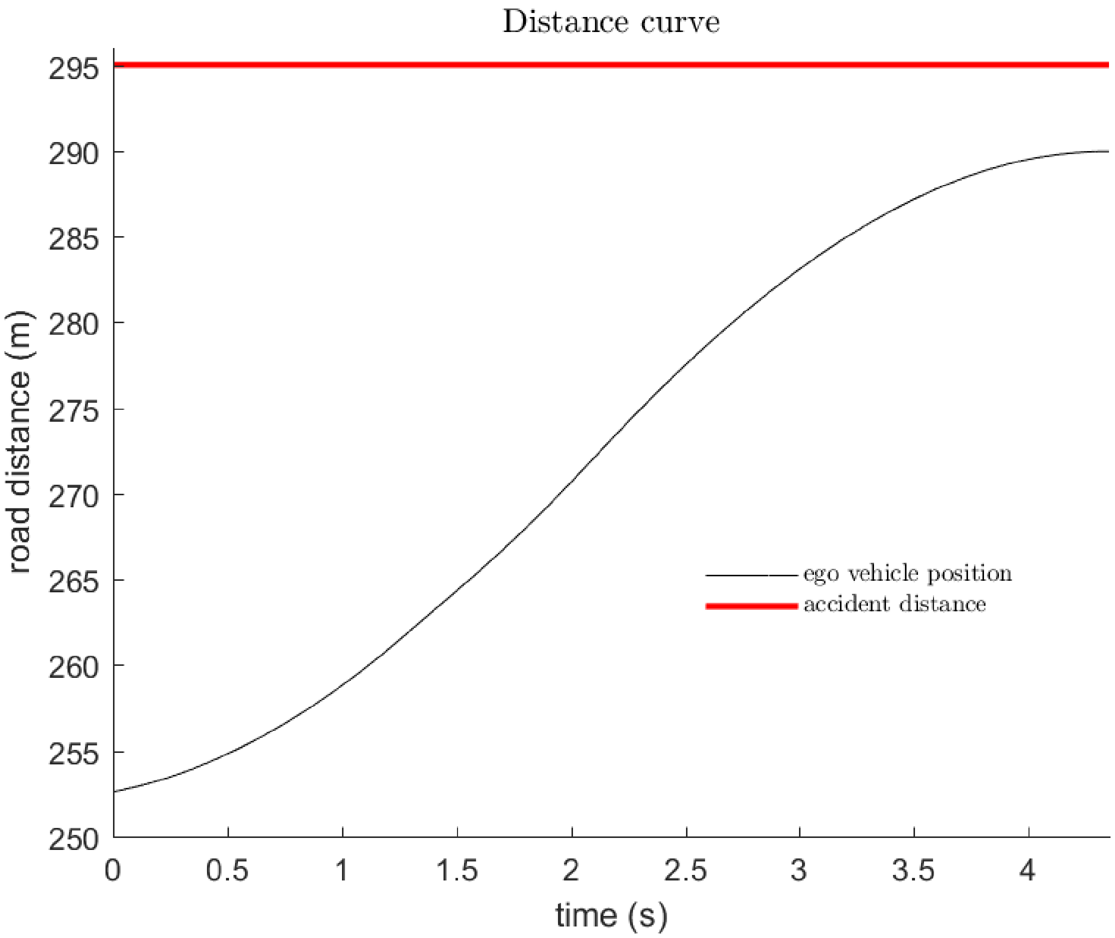


Figure 21: Distance trajectory along the road for the second detector low speed experiment, with braking intervention not triggered

Figure 22 shows the speed control of the vehicle, initialized at the start of the road with 35km/hour. Light braking was requested initially in an untrained distance before 1.5s. Notice that this coincides with the IoU peak in Figure 20. Afterwards, braking was revoked and the gas pedal was increased briefly. Breaking is triggered again after the trained distance range is reached, stopping the vehicle before a collision.

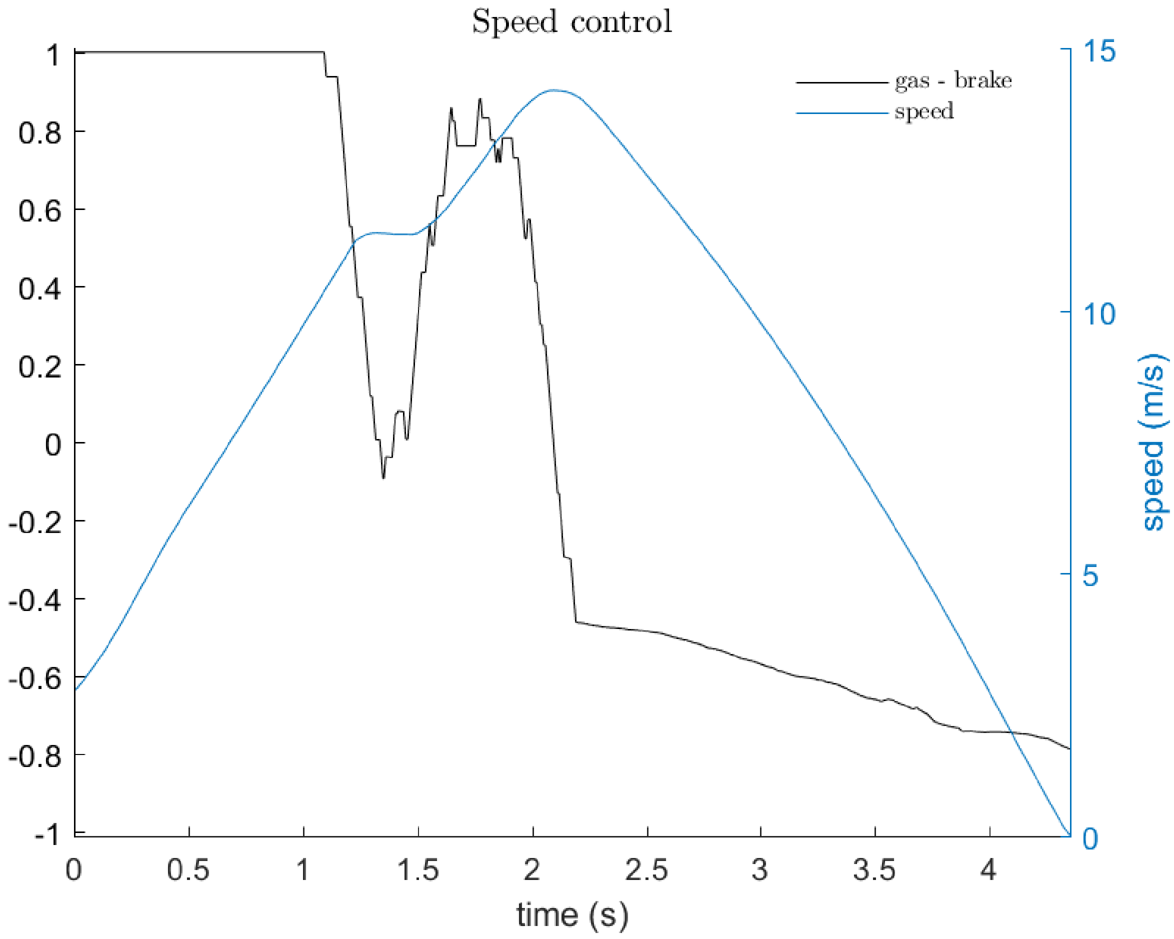


Figure 22: Speed control for the second detector low speed experiment.

High speed experiment

In the high speed experiment with the second detector, the vehicle is instead initialized at 150m along the road, with 120km/h. At this speed, the needed safety distance is beyond the learning system's trained range. If the control fails to adequately brake prior to reaching the trained range, then a safety intervention would be needed.

Figure 23 shows the filtered IoU and disparity curves in this experiment. The same trend is observed as previously in the other experiments, where the IoU performance tends to be inversely related to the run-time calculated disparity.

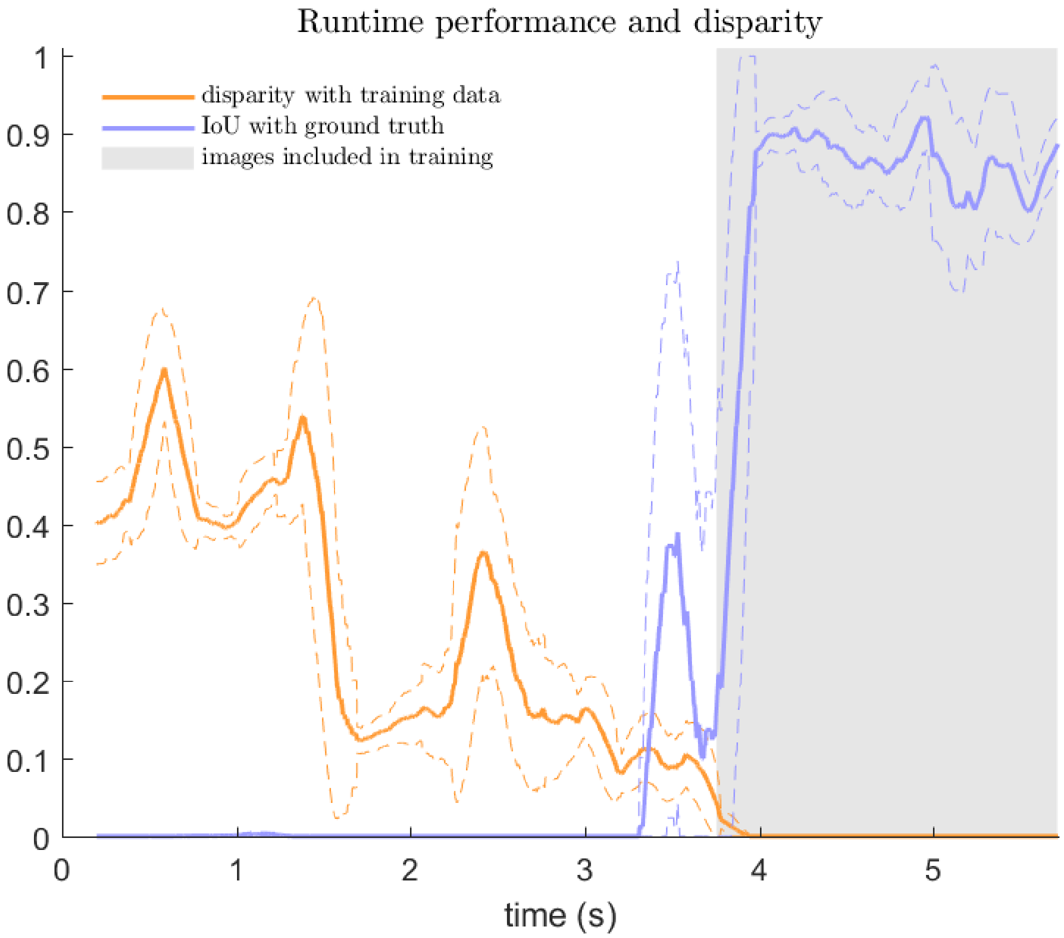


Figure 23: IoU and disparity mean and ±1 standard deviation curves for the second detector high speed experiment.

Figure 24 shows the trajectory of the ego vehicle along the road. As for the previous experiment with the first detector seen in Section 4.3.5, an intervention was triggered, which successfully stopped the vehicle and avoided the collision.

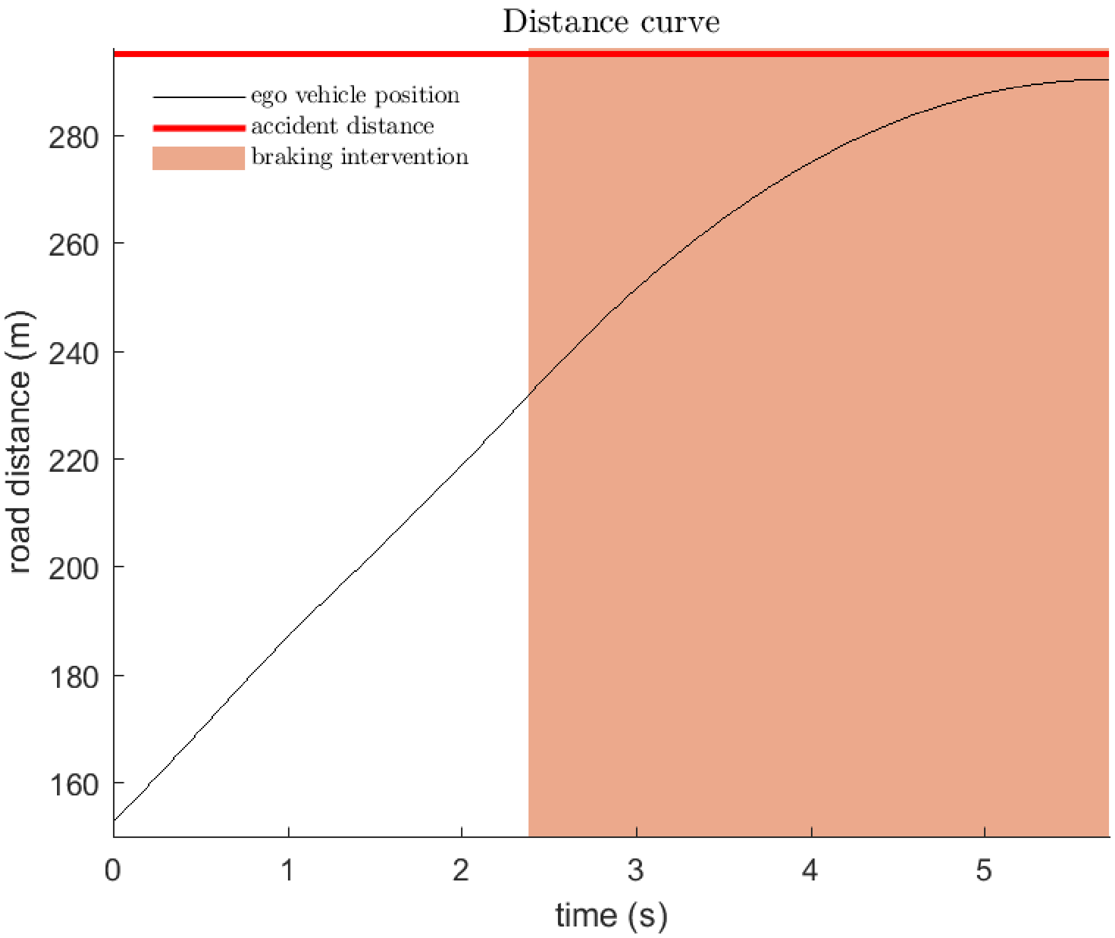


Figure 24: Distance trajectory along the road for the second detector high speed experiment, with braking intervention triggered

Figure 25 shows the speed control of the vehicle. Due to false positives (vehicle detected as at close distance when it is actually farther away), strong braking was activated before 1s. During this period zero IoU coupled with disparity peaks are observed in Figure 23. Shortly afterwards, false negatives (vehicle detected as far away when it is actually near) are obtained, which cause unsafe acceleration requiring an intervention. The controller continues to request acceleration until just before 4s where we have previously observed diminishing disparity. However, unlike for the low speed experiment, the controller requested braking too late and was already overridden by the safety monitor.

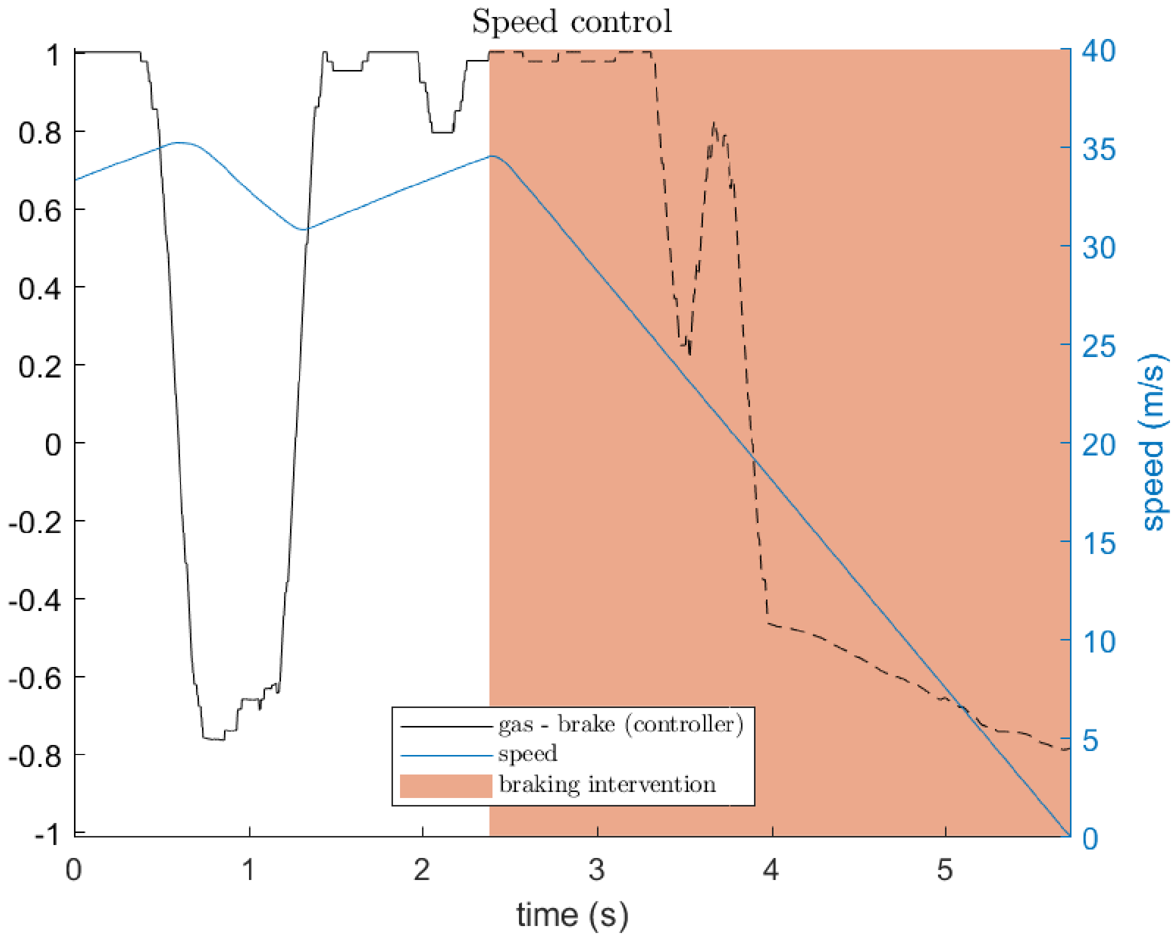


Figure 25: Speed control for the second detector high speed experiment.

4.4 Findings

In the experiments, the SMOF architecture has been used along with dynamic safety distances. The braking intervention was triggered at the right moments to prevent an accident, which resulted in the ego vehicle stopping only a few meters from collision. In addition, an example indicator for the learning system's trustworthiness has been demonstrated.

The disparity measure in the experiments tended to be inversely related to the learning system's performance, which is expected as performance should be better with data similar to the training set. This also suggests a potential for runtime trustworthiness indication using a disparity approach. Further research could investigate the development of better indicators and the desirable properties they should possess. For example, the disparity indicator used the features of only the highest scoring SURF interest point. In theory, SURF interest points are scale invariant and should not greatly fluctuate as the ego vehicle approaches the other vehicle. However, some fluctuations were observed in the experiments. Brief inspections showed that, for some distances, the highest scoring interest point can change abruptly for minor variation in scale. It could help to investigate combining several interest points, but this would increase the computational demand. Also, SURF operates on gray-scale images. This overlooks possibly essential color information that can affect the performance of the learning system. Other approaches can be tested that incorporate spatial as well as color features of images.

The motivation behind the disparity measure is to obtain a reliable trustworthiness metric that can be computed at run time for incoming data in order to indicate the performance of the learning system. The best ways to utilize this trustworthiness indication could also be a topic for further research. Using it within a safety monitor would yield a more complex monitor, and present challenges with verification or certification. However, such trustworthiness metric can be of help in alerting the driver before a safety monitor intervention is required. Safety interventions can be unpleasant for the driver, as is the case with the full force braking used in the experiments.

5 Discussion

This section discusses and summarizes the thesis, and points to possible directions for further work.

The safety concerns with learning systems were discussed in Section 2. Several concerns emerge from the loss functions and the training data used for developing learning systems. When using quantities such as average prediction error, loss functions may not account for the human cost that is relevant for safety. The training samples may not correctly reflect the scenario that was aimed for, and may inadequately represent rare cases that are highly relevant for safety. Learning systems were discussed for autonomous driving, with particular focus on end-to-end systems. The end-to-end paradigm, discussed in Section 2.1 is an extremely capable approach where complex learning problems are solved implicitly by the systems, without the need to identify and address the logical subproblems. From a safety viewpoint, this compounds the challenges associated with obtaining adequate training data, due to possible fault masking effects. Following a discussion of some relevant sections of the ISO 26262, a promising approach was indicated for meeting safety goals, which involve introducing separate architectural elements that are more amenable to meeting stringent safety requirements, while the learning systems themselves can be exempt from needing to comply with the standard. It was discussed also that verifying properties of learning systems at the technical/mathematical level may not be of help in autonomous driving applications, as that would involve specifying properties of input pixel values and output quantities. Instead, a safety monitoring approach was motivated.

Safety monitoring was discussed in Section 2.3. Formal methods and model checking are useful tools in the design of safety monitors. However, important to consider is the state space explosion problem that can limit the scalability of model checkers. The SMOF safety monitoring approach was found particularly interesting, due to the possibility for monitors that implement interventions, rather than only inhibitions, and the possibility of automatically generating safety strategies. This approach has been applied to a longitudinal collision avoidance scenario, and also a scenario targeting combined longitudinal and lateral collision avoidance. Technical implementation aspects of interventions were also discussed, and an approach was presented for calculating region boundaries using time-varying acceleration models.

A case study was presented in Section 4, involving the use of a learning system to control the vehicle's speed using camera pixel data. The use of simulations for supporting this study was discussed, with important considerations being functionalities for off-line data export, closed loop control, and an adequate virtual world accuracy. Three simulation tools were tried: IPG CarMaker, TESIS DYNAware, and TASS PreScan. CarMaker was not found to support a vision-based closed loop control setup, but was nonetheless used due to its earlier availability, and challenges with computational resources and time that anyway preclude online closed loop control. A simple scenario was pursued, for which offline processing helps in avoiding the online processing needed in a closed loop setup. Experiments were built within a simulation environment for frontal collision avoidance, including safety monitoring and control toolchains. The SMOF architecture was used, and interventions were triggered following dangerous malfunctioning behavior in the learning system. Finally, also an indicator was presented for the trustworthiness of the learning system during operation.

Further work could investigate simulations with more complex safety monitoring scenarios, beyond the frontal collision avoidance tested in this thesis. In implementing safety interventions with the approach presented in this thesis, an essential information is the expected future trajectory of the object. This was not studied, and the object was simply assumed to be stationary in the experiments.

This topic can be investigated in a further work. If the solution is probabilistic, that would present an interesting concern with regard to determinism in safety monitoring. Finally, a simple example was offered in this thesis to illustrate trustworthiness indication, and little effort was aimed at developing a good performing metric. This topic could be further investigated, including the application of such indicator within safety systems. It is important in particular that computation is feasible online.

References

[1] "Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems," Standard SAE J3016, Society of Automotive Engineers, 2016.

[2] P. Ross, "The Audi A8: the World's First Production Car to Achieve Level 3 Autonomy." http://spectrum.ieee.org/cars-that-think/transportation/self-driving/the-audi-a8-the-worlds-first-production-car-to-achieve-level-3-autonomy [accessed 2017-08-07].

[3] A. Kane, *Runtime monitoring for safety-critical embedded systems.* PhD thesis, 2015.

[4] F. Asplund, "Tool Integration and Safety : A Foundation for Analysing the Impact of Tool Integration on Non-functional Properties," 2012.

[5] J.-F. Bonnefon, A. Shariff, and I. Rahwan, "The social dilemma of autonomous vehicles," *Science*, vol. 352, no. 6293, pp. 1573–1576, 2016.

[6] B. Goodman and S. Flaxman, "European Union regulations on algorithmic decision-making and a 'right to explanation'," *arXiv preprint arXiv:1606.08813*, 2016.

[7] "Road vehicles - Functional safety," Standard ISO 26262:2011, Geneva, Switzerland, 2011.

[8] M. Machin, J. Guiochet, H. Waeselynck, J. P. Blanquart, M. Roy, and L. Masson, "SMOF: A Safety Monitoring Framework for Autonomous Systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. PP, no. 99, pp. 1–14, 2016.

[9] L. Masson, J. Guiochet, H. Waeselynck, A. Desfosses, and M. Laval, "Synthesis of safety rules for active monitoring: application to an airport light measurement robot." working paper or preprint, Feb. 2017.

[10] V. Vapnik, "Principles of risk minimization for learning theory," in *NIPS*, pp. 831–838, 1991.

[11] K. R. Varshney, "Engineering safety in machine learning," *arXiv preprint arXiv:1601.04126*, 2016.

[12] K. R. Varshney and H. Alemzadeh, "On the Safety of Machine Learning: Cyber-Physical Systems, Decision Sciences, and Data Products," *arXiv preprint arXiv:1610.01256*, 2016.

[13] D. A. Pomerleau, "ALVINN, an autonomous land vehicle in a neural network," tech. rep., Carnegie Mellon University, Computer Science Department, 1989.

[14] Y. LeCun, E. Cosatto, J. Ben, U. Muller, and B. Flepp, "DAVE: Autonomous Off-Road Vehicle Control Using End-to-End Learning," Tech. Rep. DARPA-IPTO Final Report, Courant Institute/CBLL, 2004.

[15] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[16] X. Huang, M. Kwiatkowska, S. Wang, and M. Wu, "Safety Verification of Deep Neural Networks," *arXiv preprint arXiv:1610.06940*, 2016.

[17] G. Katz, C. Barrett, D. Dill, K. Julian, and M. Kochenderfer, "Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks," *arXiv preprint arXiv:1702.01135*, 2017.

[18] Q. V. E. Hommes, "Review and Assessment of the ISO 26262 Draft Road Vehicle - Functional Safety," in *SAE Technical Paper*, SAE International, 04 2012.

[19] P. J. Ramadge and W. M. Wonham, "Supervisory Control of a Class of Discrete Event Processes," *SIAM Journal on Control and Optimization*, vol. 25, no. 1, pp. 206–230, 1987.

[20] R. I. Siminiceanu and G. Ciardo, "Formal verification of the NASA runway safety monitor," *International Journal on Software Tools for Technology Transfer*, vol. 9, no. 1, pp. 63–76, 2007.

[21] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi, "HYTECH: a model checker for hybrid systems," *International Journal on Software Tools for Technology Transfer*, vol. 1, no. 1, pp. 110–122, 1997.

[22] J. Jansson, *Collision Avoidance Theory : with Application to Automotive Collision Mitigation.* PhD thesis, Department of Electrical Engineering, 2005.

[23] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *CoRR*, vol. abs/1506.01497, 2015.

[24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 1137–1149, June 2017.

[25] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346 – 359, 2008. Similarity Matching in Computer Vision and Multimedia.

A Safety Distance for Time-Varying Acceleration

Consider that two objects are moving towards each other with relative acceleration $a(t) > 0$, given an initial relative speed v_0, which reaches 0 at the time of collision. The minimal initial separation x_0 that avoids a trajectory reaching 0 distance needs to be found.

Relative acceleration is the sum of the ego vehicle's and the object's accelerations towards each other, $a(t) = a_s(t) + a_i(t)$. Assume a conservative estimate of the object's predicted worst-case acceleration $\hat{a}_i(t)$ during the intervention time window, and a conservative estimate of the ego vehicle's ability to counteract the acceleration $\hat{a}_s$. Let the worst case evolution of relative acceleration be $\hat{a}(t) = \hat{a}_s(t) + \hat{a}_i(t)$. The task is to obtain the initial separation x_0 that avoids a trajectory reaching 0 distance.

The equivalent force causing relative acceleration $\hat{a}(t)$ can be expressed as

$$\begin{aligned} F &= m_1\hat{a}(t) + m_2\hat{a}(t) \\ &= (m_1 + m_2)\frac{dv(t)}{dt}. \end{aligned}$$

Work done by this force resulting in a small movement dx is given by

$$\begin{aligned} dw &= Fdx \\ &= (m_1 + m_2)\frac{dv}{dt}dx \\ &= (m_1 + m_2)\frac{dx}{dt}dv = (m_1 + m_2)vdv \end{aligned}$$

The amount of work done to cause a loss of the initial speed v_0 can be found using

$$\begin{aligned} W &= \int_{v_0}^{0} (m_1 + m_2)vdv \\ &= -\frac{1}{2}(m_1 + m_2)v_0^2 \end{aligned}$$

with the negative indicating work done in the opposite direction to the direction of increasing velocity. If this amount of work is done by F, then the relative speed between the objects goes to zero when they collide. We need

$$\begin{aligned} \int_{x_0}^{0} Fdx &= -\frac{1}{2}(m_1 + m_2)v_0^2 \\ \int_{x_0}^{0} (m_1 + m_2)a(t)dx &= -\frac{1}{2}(m_1 + m_2)v_0^2 \end{aligned}$$

$$\therefore \int_{x_0}^{0} \hat{a}(t)dx = -\frac{1}{2}v_0^2$$

This gives a relation between initial velocity and required separation under time-varying acceleration.

TRITA MMK 2017: 149 MES 015