A Fault-aware Sensor Fusion System for Autonomous Vehicles

A Fault-aware Sensor Fusion System for Autonomous Vehicles

By

Joshua Barkovic, B.Eng.

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements for the Degree

Master of Applied Science

McMaster University

© Joshua Barkovic, September 2020

Master of Applied Science (2020)	McMaster University
(Software Engineering)	Hamilton, Ontario

Title:	A Fault-aware Sensor Fusion System for Autonomous Vehicles
Author:	Joshua Barkovic, B.Eng. (McMaster University)
Supervisors:	Dr. Alan Wassyng & Dr. Mark Lawford
Number of Pages:	vii, 105

Abstract

There have been several accidents involving autonomous vehicles on public roadways under scenarios that are normally avoidable by competent human drivers. This thesis contains a review of these accidents and their causes as a result of inadequate hazard mitigation. As a solution to this problem, a novel design pattern is proposed. This design pattern was developed from a hazard analysis using Systems Theoretic Process Analysis (STPA) methodologies that analyzed the circumstances common to several of these accidents. To demonstrate the effectiveness of the novel design pattern, an example system is constructed and tested in simulation against several accident scenarios similar to the ones studied. The results are then explained to demonstrate the effectiveness of the proposed design pattern.

Acknowledgements

I would like to thank my supervisors Dr. Alan Wassyng and Dr. Mark Lawford for their endless patience and guidance throughout the lengthy process of writing this thesis, Dr. Douglas Down, Spencer Deevy, and Dr. Wenbo He for their contributions and understanding throughout the process of defending it.

The EcoCAR Mobility Challenge - Advanced Vehicle Technology Competition provided me with an invaluable learning opportunity and a chance to explore these topics in a real world setting. I owe a lot of my sanity to the members of the McMaster team.

Finally, none of this would have been possible without the unwavering support of family, friends, and fellow grad students in room 102.

Table of Contents

Descriptive Note ii

Abstract iii

Acknowledgements iv

Table of Contents v

List of Figures viii

List of Tables xi

List of Listings xii

List of Acronyms xiii

1 Introduction 1

1.1 Motivation 1

2 Background 3

2.1 Existing Sensor Fusion Architectures 3

2.1.1 Dasarathy's Classification Scheme 3

2.1.2 Society of Automotive Engineers (SAE) Levels of Driving Automation 6

2.1.3 Assured Clear Distance Ahead 6

2.1.4 Automotive and Autonomous Vehicle Domain-Specific Terms and Definitions 8

2.2 Sensor Technologies and Limitations 9

2.2.1 Vision Sensors . . . 9
2.2.2 Radar Sensors . . . 10
2.2.3 Ultrasonic Sensors . . . 11

3 Accidents **12**
3.1 Accident Review Methodology . . . 13
3.2 Tesla - May 7, 2016 Florida . . . 13
3.3 Uber - March 18, 2018 Arizona . . . 20
3.4 Tesla January 22, 2018 California . . . 25
3.5 Summary . . . 28

4 Hazard Analysis and System Requirements **30**
4.1 Analysis of an Autonomous Vehicle System . . . 30
4.1.1 Notes . . . 31
4.1.2 Brief overview of STPA . . . 32
4.1.3 Step 1: Control Flow . . . 32
4.1.4 Step 2: List of Accidents . . . 33
4.1.5 Step 3: System Hazards . . . 34
4.1.6 Step 5: Unsafe Control Actions . . . 35
4.1.7 Step 6:Causes of Sensor Fusion Unsafe Control Actions . 38
4.2 Safety Requirements . . . 41
4.3 Discussion . . . 44

5 Risk Mitigation Through Better System Design **46**
5.1 Design Pattern using Nodes and Monitors . . . 47
5.1.1 Definition of Nodes . . . 47
5.1.2 The Purpose of Monitors . . . 49
5.1.3 Combining Monitor and Node Output to Derive Confidence in Sensor Fusion Results . . . 51

6 A Practical Example **53**
6.1 Recreation of the Tesla AEB/FCW Systems in Simulink . . . 55
6.1.1 Components of Reference System . . . 55
6.2 Addition of a Monitor for the Sensor Fusion Subsystem . . . 58
6.2.1 Sensor Fusion Metrics . . . 58

6.2.2 Additional Sensor Information 63
6.2.3 Calculating Probability of Risk and Uncertainty Values . 64
6.2.4 Detecting False Negatives Using a Monitor for the Automated Emergency Brake (AEB)/Forward Collision Warning (FCW) Node 69
6.2.5 Warning Output to the Driver 71
6.3 Discussion 72

7 Testing **74**
7.1 Notes on Testing Methodology 75
7.2 Validating the Unmodified Reference System 76
7.3 Evaluation Against the Recreation of the May 7th Tesla Accident in Florida 79
7.3.1 Unexpected Results 82
7.4 Discussion of Results 83

8 Conclusions & Future Work **87**
8.1 Future Work 88

Appendices **91**

A Meta-model **92**

B Simulink Model Parameters **96**
B.1 Vision Detector Parameters 96
B.2 Radar Detector Parameters 98

C Extra Test Results **99**

List of Figures

2.1 Dasarathy's Levels of Sensor Fusion – rearranged from[4] 5
2.2 Terminology used in this thesis to discuss levels of autonomy in autonomous vehicle systems, as defined in SAE J3016[8] 7

3.1 Overhead view of crash site on US-27A, showing intersection with left turn lanes[15]. 15
3.2 Approach to the crash scene from the truck driver's point of view[15]. 16
3.3 Approach to the crash scene from the Tesla driver's point of view[15]. 16
3.4 Overhead view of the recorded data from the Uber ADS system 1.3 seconds before impact. From [16] 20
3.5 A photo of the accident scene that appeared in the National Transportation Safety Board (NTSB) report brief [17] 25
3.6 Figure from the NTSB report brief that shows the usage of Autopilot during the trip, ending at the time of the crash [17] . . . 27
3.7 Figure from the NTSB report brief that shows the hands on/off status from the data they collected as well as the timings of warnings issued to the driver to keep their hands on the wheel. The data is for the last autopilot segment of the trip, ending with the crash[17]. 27

4.1 Control flow diagram used for STPA analysis 33

5.1 An example arrangement of sensor fusion nodes (in white) and associated monitors (in grey). 52

6.1 Plot results of calculated probability of risk (top graph) and uncertainty (bottom graph) during a test of the example system using a scenario simulating the Tesla Florida accident from Section 3.2. Values are represented on a scale from $[0-1]$ and are mapped to $[blue-yellow]$ by the plotting tool. The position of the ego car is depicted by the rectangle of 6 cells centred at position (40, 0) of each plot; these are added artificially for clarity. Direction of travel of the ego vehicle is from left to right. 59

7.1 A test case designed to test AEB and FCW algorithms, partially resembling the events of the accident in Section 3.4, but without the additional lead vehicle 77
7.2 A recreation of the May 7, 2016 Tesla accident, documented in Section 3.2 80
7.3 The sensor array on the virtual test vehicle, comprised of only long range radar and a forward facing camera detector. The red dots shown are simulated detections from the radar sensor and the red and blue 'pie slices' represent the radar and camera Field of Views (FOVs), respectively. 81
7.4 The conditions leading up to the second unexpected result. In the image you can see 2 different detection locations on the vehicle. Due to its close proximity, it straddles the detection coverage limit of the camera sensor and therefore the detection produced by the camera is too far from that produced by the radar and the sensor fusion algorithms do not produce a consistent detection location. This, over time increases the risk measured by the early warning system which then triggers a warning. In the image, squares represent confirmed detections produced by the base system, triangles are suspected detections that have not been confirmed, and diamonds are those identified by the early warning system as potential false positives. 84
7.5 The level of risk measured due to inconsistent sensor and sensor-fusion readings 85

C.1 Risk over time captured during testing of the AEB/FCW functionality. At around 5 seconds the total risk begins to drop quickly due to the first detections of the stopped vehicle by the main tracking algorithm. Interestingly, the accident occurs around 7.7 seconds in (at the location of the scope cursors), where the risk, interpreted as 'risk of false negatives' is at its lowest. Due to the low risk relative to the long term average, the early warning system, labelled 'monitor-initiated warning' in the plot, does not activate 100

C.2 The test results from running the simulated May 7^{th} test case against the system with monitors. The top graph shows a large increase in short term system risk as the vehicle approaches the truck without it being detected by the main sensor fusion node. An early warning is issued, labelled as 'monitor-initiated warning' in the plot. The warning can be seen in the middle graph, triggered at around 8.5 seconds, with the actual collision taking place at 11.7s, shortly before the steep drop-off in detected risk. This drop-off is due to the fact that the vehicle has passed the obstacle in the simulation, and there are no longer any spurious sensor readings. 101

List of Tables

6.1 Data produced by each sensor monitor block 64

List of Acronyms

OOC Object of Concern 31

ACC Adaptive Cruise Control 8

LKA Lane Keep Assist 8

FCW Forward Collision Warning vii

AEB Automated Emergency Brake vii

UCA Unsafe Control Action 32

SAE Society of Automotive Engineers v

V2X Vehicle to X Communications 9

STAMP Systems-Theoretic Accident Model and Processes 31

STPA Systems Theoretic Process Analysis iii

TP True Positive 66

FP False Positive

FN False Negative 44

FOV Field of View ix

NHTSA National Highway Traffic Safety Administration 13

NTSB National Transportation Safety Board viii

LTAP Left Turn Across Path 17

ODD Operational Design Domain 19

FTA Fault Tree Analysis 52

GNN Global Nearest Neighbour 56

JPDA Joint Probabilistic Data Association 57

ACDA Assured Clear Distance Ahead 6

Chapter 1

Introduction

1.1 Motivation

There have been an increasing number of fatal accidents on public roadways involving autonomous vehicles that are the result of situations which many experienced drivers would be familiar with. The concerning part about these accidents is that while they are atypical, they are also not uncommon and are frequently avoided by experienced and alert human drivers. Furthermore, it is often the case that these autonomous systems were not designed to handle these situations due to engineering limitations at the time of production, and that this is known by system designers but inadequately communicated to users of these vehicles. Additionally there are often ineffective design measures in place to prevent these kinds of accidents. Instead, designers rely on human oversight from vehicle operators to mitigate hazards associated with these limitations. This practice is highly inadequate due to the tendency for operators to become complacent and distracted as the level of implemented autonomy increases.

During investigations of such accidents, the design of the autonomous

system is often not listed as a main cause of the accident. Instead focus is commonly placed on the misuse of the vehicle's systems by the operator and their interactions with the system leading up to the accident.

It is my belief that this is unacceptable and it is the purpose of this thesis to examine the causes of some of these edge case accidents through case studies and to provide a method of identifying and reducing their likelihood during the design phase of such systems. The intention is to develop a way of accomplishing this task using only the equipment and levels of system complexity that would be expected in current SAE Level II vehicles. This will demonstrate that it is in fact possible to make safer Level II autonomous vehicles without resorting to more advanced hardware, better sensors, and higher levels of autonomy.

Chapter 2

Background

2.1 Existing Sensor Fusion Architectures

There are several existing software architectures for designing sensor fusion systems in the literature. One of the earliest pushes to standardize sensor fusion terminology and system design came from the U.S. Joint Directors of Laboratories (JDL)[5]. in the 1980's. The push for a standardization of sensor fusion software architectures is relatively recent and aims to increase the speed of implementation[9].

2.1.1 Dasarathy's Classification Scheme

An existing system for classifying sensor fusion levels was proposed by Belur Dasarathy[4]. This system classifies sensor fusion algorithms based on the levels of data abstraction of both the inputs and outputs of the algorithm. Dasarathy proposes the following levels of data abstraction:

- Data (DA) - The raw data originating from a sensor or other type of data source. Data at this level is what is perceived by the sensor before

a target is identified and has had attributes associated with it.

- Feature (FE) - Features are the attributes of a target, such as position, size, and velocity, etc, in the context of detecting vehicles on a roadway. These are in comparison to radar return signals, and camera sensor data, which would be classified as Data level information.

- Decision (DE) - A decision is the result of higher level reasoning about the features of a target, such as a classification of the target or its motion and intentions.

With these levels of data abstraction, the author then describes 5 levels of sensor fusion for classifying algorithms, shown in Figure 2.1:

- DAI-DAO - Data Input Data Output: For fusion between similar sensor types, such as an array of homogeneous sensors.

- DAI-FEO - Data Input Feature Output: For fusing raw data from one or more sensors and producing feature level measurements of an object, such as depth perception in the human eye[4].

- FEI-FEO - Feature Input Feature Output: The combination of measurements of object features as measured by multiple sensors.

- FEI-DEO - Feature Input Decision Output: The production of decisions about an object (such as classification) based on the feature input of one or more sensors.

- DEI-DEO - Decision Input Decision Output: The combination of decisions from multiple sources, such as with a voting scheme.

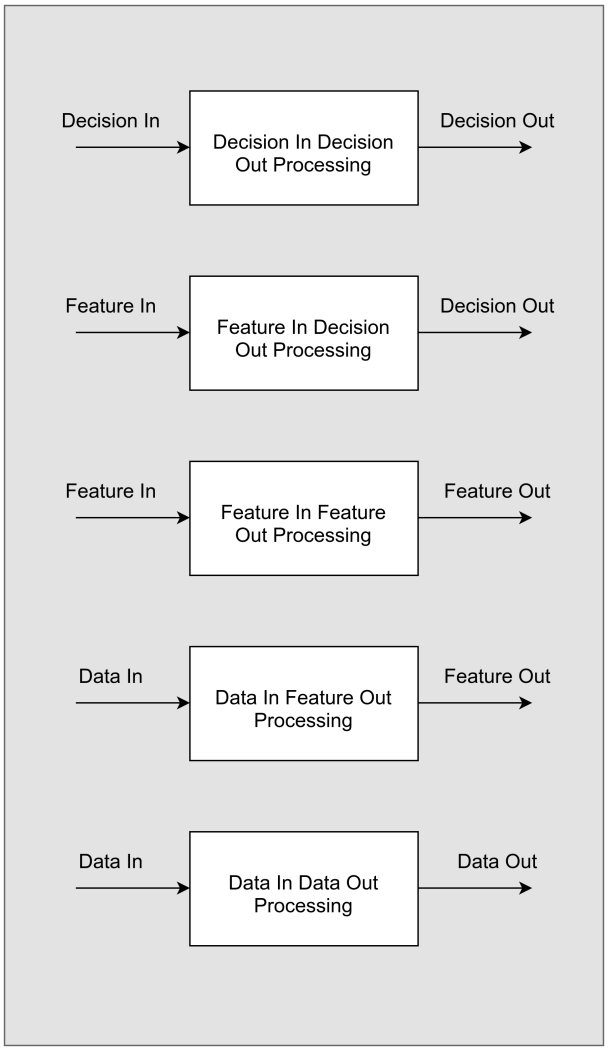


Figure 2.1: Dasarathy's Levels of Sensor Fusion – rearranged from[4]

The author has also suggested that you can form levels with larger jumps in data abstraction, such as DEI-DAO in the case of a sensor with complicated built-in pre-processing, such as the Mobileye camera system.

2.1.2 SAE Levels of Driving Automation

The SAE has designated several levels of driving automation in an attempt to standardize terminology used in industry to describe levels of increasing automation. The standardization also dictates the collections of features and use cases that are normally associated with each level. However, people will often combine levels when describing a vehicle that is not easily pigeon-holed by saying, for example: '.. a level 3 or 4 autonomous vehicle...'. These levels and their definitions are shown in Figure 2.2, taken from the standard, and are referred to at various points in this thesis as a stand-in for the things they describe.

2.1.3 Assured Clear Distance Ahead

The concept of Assured Clear Distance Ahead (ACDA)refers to the required sight lines that a driver must have in order to carry out their driving tasks at a given speed. It takes into account a combination of typical human reaction times and vehicle stopping distances. The resulting distances for given speeds are then used to improve road layout and set recommended speed limits. This ensures that a driver following the speed limit and driving according to current conditions always has the required time to react to sudden events[11].

There are many examples in everyday driving where ACDA may not be met, such as when driving at night with headlights or negotiating blind corners.

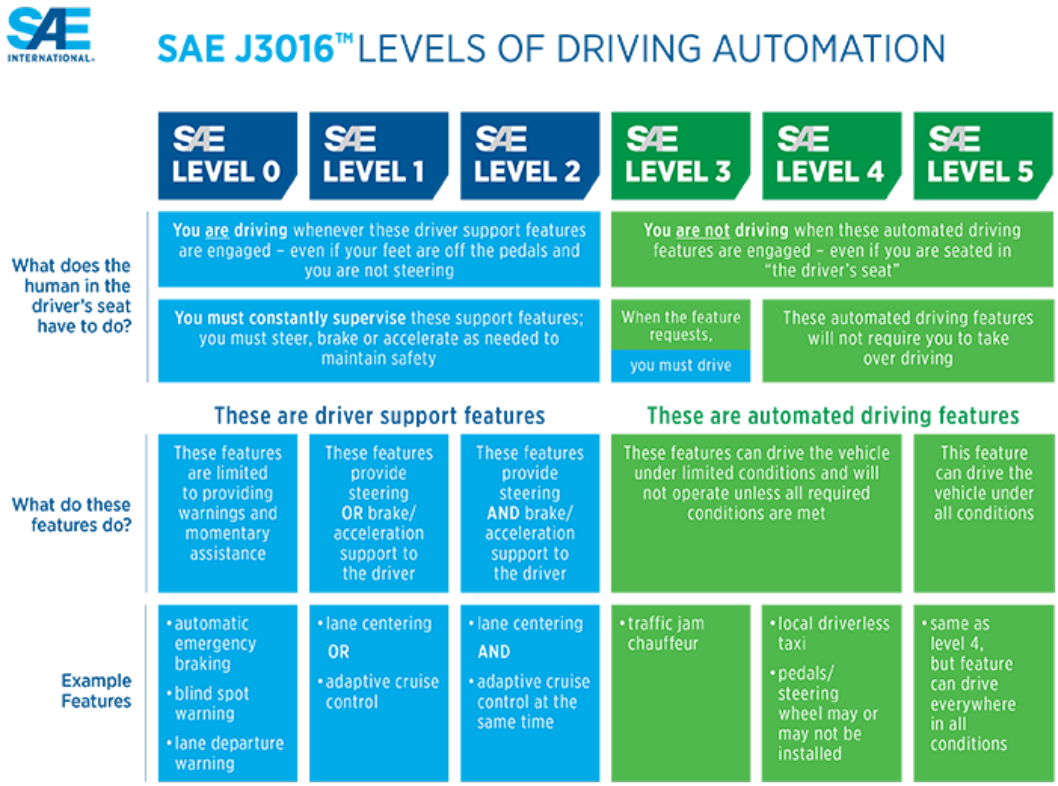


Figure 2.2: Terminology used in this thesis to discuss levels of autonomy in autonomous vehicle systems, as defined in SAE J3016[8]

.

Human drivers are able to eventually recognize the danger in these situations with experience, but this needs to be explicitly accounted for in the design of autonomous systems.

2.1.4 Automotive and Autonomous Vehicle Domain-Specific Terms and Definitions

- **Adaptive Cruise Control (ACC)** An advanced system of cruise-control that can maintain a configurable following distance, as well as a set speed. More advanced versions of such systems can also negotiate stop-and-go traffic.

- **Automated Emergency Brake (AEB)** A system designed to prevent some types of frontal collisions, with newer models of cars being able to prevent some rear end collisions during reversing manoeuvres. Maximum allowed braking force is usually limited to prevent accidents from the system activating incorrectly at high speed.

- **Forward Collision Warning (FCW)** A driver alert system designed to warn of an impending collision with an object. Most implementations are currently designed to prevent frontal collisions, however some systems can warn about and prevent rear end collisions when reversing, in combination with AEB systems

- **Lane Keep Assist (LKA)** A system designed to automate the task of lane following on long repetitive stretches of highway, with support for limited curve negotiation depending on system design. The maximum allowed turn rate the system can command is usually limited for safety

reasons.

- **Ego*** Terms such as **ego vehicle** and **ego lane** are commonly used by designers of autonomous systems to unambiguously refer to the vehicle containing the system under question, as opposed to surrounding traffic. In this case the definition of 'ego' referring to one's self and internal state is the best way of understanding these terms.

- **Vehicle to X Communications (V2X)** Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I) (and others), are relatively new technologies that enable vehicles and other road actors, such as traffic lights to communicate with each other via short distance radio communications using on-board modems. These communication channels can significantly simplify negotiation logic for intersections, increase fuel economy, and ease traffic congestion by enabling 'borg-like' vehicle cooperation in some driving scenarios.

2.2 Sensor Technologies and Limitations

Several common sensor technologies used in the design of autonomous vehicles are listed below along with a brief description of them and their limitations.

2.2.1 Vision Sensors

Vision sensors such as cameras and lidars are capable of accurately detecting object shapes and sizes, however they are all susceptible to poor atmospheric conditions such as snow and rain.

With the exception of calibrated stereo vision systems, camera based sensors

are generally most accurate at performing lateral and object width measurements, with limited accuracy measuring longitudinal distances. For this reason, cameras and radars are a common pairing used as the bare minimum in most autonomous vehicle designs on the market.

Lidars are active sensors that incorporate one or more laser beams and can estimate the relative distance, angle and speed of a nearby object by scanning the environment with laser pulses and analyzing the timing of the reflected signals. Additionally, they can estimate the reflectivity of objects by measuring the intensity of the returned signal. This can be used to detect lane markings due to their increased reflectivity compared to the road surface. 3D lidar sensors are additionally used to assist vehicle localization due to their ability to rapidly scan their surroundings and provide accurate point-distance readings that can be associated with pre-recorded lidar data from the same location.

All vision sensors are vulnerable to occlusion from objects in their environments due to the use of reflected light for providing detections. They are unable to track objects that are fully or sometimes partially occluded.

2.2.2 Radar Sensors

Radar sensors are very effective at measuring longitudinal relative position and velocity, with some sensors able to provide lateral measurements, but with much less accuracy. Radars are less vulnerable to precipitation than vision sensors and can sometimes 'see past' obstacles by analyzing secondary reflections off of road surfaces and barriers. However, they can suffer from clutter in scenes that contain many reflective objects.

2.2.3 Ultrasonic Sensors

Ultrasonic sensors are capable of detecting the distance of objects very close to the vehicle and were originally used by the automotive industry to provide front and rear park-assist features. They can also be used by autonomous systems during low speed manoeuvres as an extra source of object position data for use in sensor fusion processes in order to produce more accurate measurements for close-range obstacles at low relative speeds. However, their range is often limited to a few meters and are only accurate under most conditions for a fraction of that distance. Additionally, they rely on the transmission and reception of sound waves to detect obstacles, and are therefore vulnerable to effects from precipitation, dirt, and other things that can dampen the sound or produce noise.

Chapter 3

Accidents

In this chapter, several recent accidents where it has been determined that autonomous features were controlling the vehicle at the time of collision are reviewed. For each accident, a summary of publicly available details about potential causes and factors involved is provided. Investigations were conducted after each accident by US government agencies and details for these summaries were pulled from the corresponding reports. In each case, it was determined that the autonomous systems installed in the vehicles were functioning as designed during the collision and major factors in each were driver/operator misuse caused by inadequate awareness of the limitations of the systems they were using as well as inadequate monitoring of the driver/operators. In each case, the manufacturers relied on driver awareness to recognize and react to hazardous situations that the system was not designed to handle. In each case the operator failed to do so. These examples are the types of situations that the work in the remainder of this thesis strives to avoid.

3.1 Accident Review Methodology

Accidents were selected based on the availability of reliable information about their cause and their ability to demonstrate, among other things, the effects of poor design decisions regarding sensor fusion and faulty assumptions about the reliability of human operators of autonomous systems. Information for the accidents in this section was gathered from official reports summarizing investigations conducted by the US NTSB (National Transportation Safety Board) and National Highway Traffic Safety Administration (NHTSA).

3.2 Tesla - May 7, 2016 Florida

On May 7th 2016 a Tesla Model S was travelling eastbound on Highway US-27A when it collided and passed under the trailer of a combined tractor-trailer vehicle that was in the process of making a left turn in front of the car, killing the driver of the Tesla. The Tesla was operating in 'Autopilot' mode at the time of the crash.

Autopilot is a marketing term used by Tesla for its implementation of an SAE J3016 Level II autonomous system that is a combination of several common ADAS features, including Traffic Aware ACC, Autosteer for lane centring, and Lane Change on Demand. The combination of these features is intended to allow the vehicle to automate the mundane portions of driving on certain types of highways. The version installed in the vehicle involved was designed to be operated on 'limited access' highways and freeways only. These are roads divided with a physical barrier, and where access is limited to on/off ramps that connect to overpasses. Driving tasks in this type of environment are easier to

automate due to the predictability of traffic flow and the absence of pedestrians, intersections, and oncoming traffic. Due to these design restrictions, there are many typical driving situations that Tesla's Autopilot and indeed other industry equivalent systems cannot handle, such as another vehicle failing to yield and turning in front of the ego vehicle and other situations that would be unusual, but not impossible when travelling at highway speeds. These restrictions are necessary due to capability limitations of current sensor deployments in vehicles combined with the difficulty of predicting abnormal actions of other road users. Tesla's Autopilot however, does not prevent activation on 'non-preferred' roads and relies on the driver to constantly monitor the road for potential hazards and to intervene if the autonomous system does not safely react to a hazardous situation. This is enforced through a large section of the user manual that describes proper usage of the system, a one-time disclaimer that appears on first-time activation of the system, and by monitoring driver engagement through measurement of driver-applied steering wheel torque. There are several problems with this technique, including that not everyone will read the user manual for their car, if there is a second driver of the vehicle they will not see the one-time disclaimer. Additionally, the monitoring system can allow the driver to remove their hands for several minutes at a time before requiring interaction and the system is easily 'gamed' since it only requires users to briefly touch the wheel to reset the hands off timer interval[14][15].

US-27A is a divided highway with a grass median that contains grade-level intersections with side roads and driveways; it has a speed limit of 65 mph (105 km/h) and a design limit of 70 mph (113 km/h). It permits left turns into oncoming traffic and provides direct access from perpendicular local roadways and driveways without the use of overpasses, service roads and on-ramps to

Figure 3.1: Overhead view of crash site on US-27A, showing intersection with left turn lanes[15].

isolate and protect traffic. Instead the highway contains intersections with dedicated left turn lanes requiring drivers to cross oncoming highway traffic (see Figures 3.1 to 3.3). This type of road is divided but not limited-access. Despite this, the NTSB report indicates that the Tesla system classified this road as a preferred roadway and would have allowed the car to travel at speeds up to 90 mph (145 km/h). The vehicle was travelling at 74 mph under the control of Autopilot when it collided with the trailer[15].

Both the NHTSA and NTSB investigated the accident and concluded that neither driver was likely impaired at the time and, despite the presence of several electronic devices in the Tesla it could not be determined whether its

Figure 3.2: Approach to the crash scene from the truck driver's point of view[15].

Figure 3.3: Approach to the crash scene from the Tesla driver's point of view[15].

driver was distracted by them. They also found that glare from sunlight was not a factor, and the weather was clear and dry that afternoon. Based on data from the vehicles and eyewitness accounts, it was calculated that the Tesla and truck drivers had 7 and 10.4 seconds (respectively) to see and react to each other before the collision[14][15]. The difference in potential reaction time is due to the effect that vehicle height can have on sight distance.

According to the data retrieved from the Tesla, neither the autonomous system nor the driver provided any steering or brake input before the collision. The FCW/AEB system did not detect the truck, therefore providing no warning to the driver or braking intervention. In fact, the last vehicle detection received by the AEB system had occurred 1 minute 35 seconds before the collision[15]. The NHTSA investigation concluded that the Autopilot system on the Tesla was working correctly as it was not designed to prevent collisions with turning cars (Left Turn Across Path (LTAP) scenarios) and that this was industry standard at the time for similar vehicles (SAE Level II autonomy)[14]. Due to this limitation, most manufacturers of cars with level II autonomy strongly recommend that these systems only be used on limited access highways and with the driver's constant supervision. Unfortunately, these warnings sometimes use official wording when describing allowed road types that may end up being confusing or ambiguous to most users. For example, there are similarities and contradictions in the definitions of 'limited access highway' and 'freeway'; and depending on your accepted definition of 'highway', the first is actually an oxymoron. These problems can make it hard for such written warnings to be effective when crossing regional, educational and language boundaries.

Since the Autopilot system was being used outside of its intended environment, both investigations determined it was functioning correctly and instead

focused on the Tesla driver's misuse of the Autopilot, his driver history, and his interactions with the driver monitoring system. The latest version of Autopilot software at the time of the crash could allow drivers to operate hands-free indefinitely if specific lead vehicle status, vehicle speed, and max lateral acceleration (estimation of road curvature) conditions were met[14][15]. Otherwise, the system would warn the driver to put their hands on the wheel after a time delay dependent on which combination of conditions were met. If the driver still did not respond, this would be followed by a sequence of further warnings, eventually resulting in the vehicle turning on its hazards and slowing down and stopping in the lane of travel. These warnings could be cancelled by drivers briefly touching the steering wheel and there was no limit put on the number of times this pattern of inattention would be allowed to occur in a single drive cycle[15]. Several of these points were addressed by Tesla following the crash investigation and implemented on existing vehicles via software updates.

Analysis of the data showed that the driver had been warned 7 times during the accident trip to keep his hands on the wheel. During the entire 41-minute trip, Autopilot was active for 37 minutes, of which the driver had their hands on the wheel for a combined total of only 25 seconds. The data also indicated that the driver had last touched the steering wheel more than 4 minutes before the crash, when the car first turned onto US-27A and Autopilot was re-enabled.

The FCW and AEB systems in the Tesla used information from a windshield mounted, forward facing Mobileye camera with an EyeQ3 processor[14], and a centre-mounted forward facing radar. This type of system is best at detecting moving vehicles travelling in the same direction as the ego vehicle with some ability to detect pedestrians, cyclists, and stationary vehicles. Due to the lack of accurate height measurements from the radar, this system is unable to reliably

distinguish between harmless overhead stationary objects like bridges and stationary objects on the roadway. To prevent unnecessary warnings and brake application, Tesla's implementation required agreement from *both* the camera and radar before the system would react to potential stationary objects[15]. It is the author's belief (and opinion) that in this case, the radar likely would have provided some detection information about the truck due to it being an ideal target for radar reflections and that these detections were not supported by the data from the Mobileye camera, which is known to have a limited ability to recognize vehicles perpendicular to the direction of travel[6][7].

In 2016 the NHTSA released its AV Policy, a set of guidelines for (HAV) Highly Autonomous Vehicles (SAE J3016 Level 3 and above). In these, they include a concept of Operational Design Domain (ODD) of an autonomous vehicle. The ODD can include factors such as: roadway type, geographic location, clear roadway markings, weather conditions, speed range, lighting conditions, and other manufacturer-defined system performance criteria. The Insurance Institute for Highway Safety recognizes the importance of ODDs, even for level II autonomy, stating: 'Driving automation systems should self-enforce their use within the operational design domain rather than relying on users to do so'. The NTSB recommends that manufacturers of vehicles with level II autonomous systems incorporate system safeguards that limit the use of those features outside of their ODD. They also recommend that the NHTSA develop a method to verify that manufacturers of these vehicles actually implement these safeguards[15].

After the accident, Tesla upgraded their fleet of vehicles to Autopilot version 8.0 via an over-the-air update. In this revision, they made changes to the logic that affects the timing of hands off alerts to the driver including reducing the

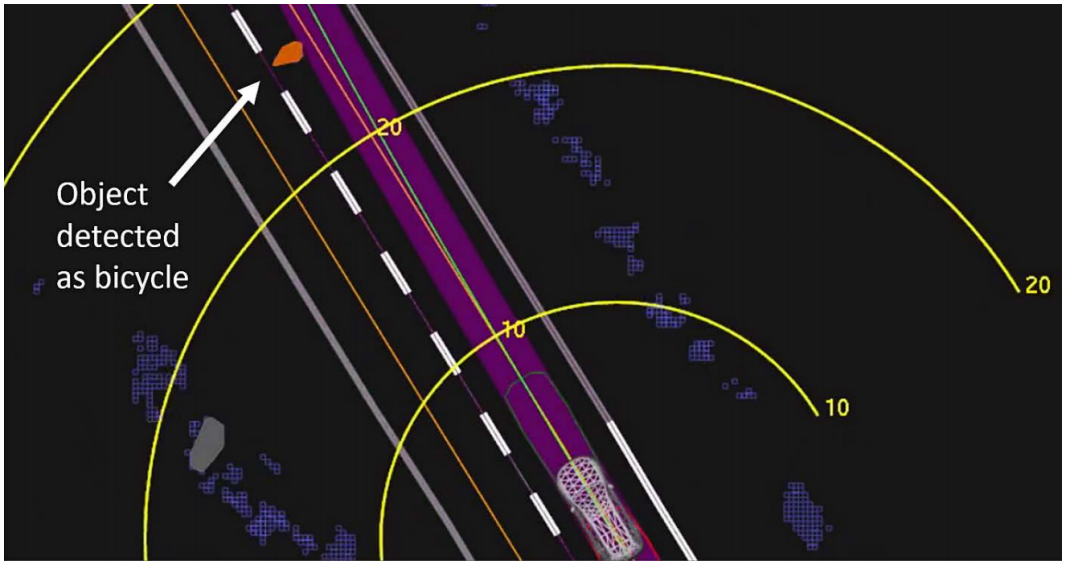


Figure 3.4: Overhead view of the recorded data from the Uber ADS system 1.3 seconds before impact. From [16]

time before an alert is issued on a non-preferred road. They also implemented a 'strike-out' feature that disables Autopilot until the next ignition cycle if the driver is warned 3 times during a trip. However, 'Autopilot' can still be activated on non-preferred roadways.

3.3 Uber - March 18, 2018 Arizona

On the night of March 18, 2018, an SUV used by Uber to test its developmental Autonomous Driving System (ADS) struck and killed a pedestrian that was crossing in front of the car outside of any designated crosswalk. The vehicle was being controlled by the autonomous system at the time, and neither the ADS nor the safety driver in the vehicle took effective action to avoid the collision.

The pedestrian was pushing a bicycle with them while crossing the road and did not react to avoid the SUV. They were found to be impaired at the

time of the crash.

Uber was testing its ADS on public roads in several US and Canadian cities. They required safety drivers in their vehicles to monitor the system while conducting test drives and to intervene in emergency situations to prevent collisions. The safety driver in this incident was watching a video on her phone at the time of the accident and did not see the pedestrian in time to react. Uber had also recently started using only one safety driver per vehicle instead of two, increasing the load on the driver and the likelihood of distraction[16].

The Uber system used three main sensor arrays, each providing 360° coverage around the vehicle for object detection: a Velodyne Lidar with a range of 100 m, 8 radars, and 11 cameras; the lidar was used for both object detection/classification and vehicle localization. Additionally there were 12 bumper-mounted ultrasonic sensors, primarily used to assist with lane change manoeuvres[16]. Because the radars and ultrasonic sensors do not require visible light to operate, and the lidar is an active sensor, only the cameras would have been affected by the decreased visibility at night, meaning at least two types of sensors should have detected the pedestrian.

Despite having an adequate sensor suite with 3 complimentary sensor types and overlapping coverage, the ADS did not detect and prevent a collision with the pedestrian. Upon detailed investigation of the logic used by Uber developers, the NTSB found several flaws in the version of the system in use at the time of the crash[16]:

1. If the ADS detected an emergency situation that required braking of more than $7m/s^2$ or a jerk greater than $\pm 5m/s^3$, the system would enter an 'action suppression' state lasting 1 second where the ADS would suppress

braking while it verified the nature of the situation and calculated an evasive manoeuvre. Action suppression was added by developers to prevent unexpected and dangerous vehicle movements caused by false alarms from the system. It would allow the ADS to avoid performing an extreme manoeuvre if the hazardous situation disappeared or resolved itself. The safety driver was expected to intervene if the situation was actually hazardous. However, the system was intentionally designed to not warn the driver that action suppression was occurring in an effort to reduce driver confusion, according to Uber engineers. Due to this, the system did not warn the safety driver about the pedestrian, even though it had started to detect them several seconds before collision.

2. At the end of the action suppression state, if the operator had not noticed the problem themselves and taken control, the system would then execute the evasive manoeuvre as long as it could still avoid the collision without exceeding the limits on deceleration and maximum jerk imposed on it. If avoiding the collision was not possible, or would require more braking than was allowed, the system would not attempt to prevent the collision and would instead auto-disengage and begin a gradual slowdown while warning the driver that is was disengaging. The system was not designed to provide partial braking to reduce the severity of an accident, as is common on production AEB systems.

3. In order for the Uber system to track an object, it had to maintain a consistent classification status. If the object was detected more than once, but received a different classification result, it was considered to be a 'new' object by the sensor fusion system and any previous location detections

or track calculations were discarded when estimating the objects future motion. Additionally, when an object had a classification assigned to it, the system would assume a set of potential future paths that it may take based on the assigned object type and any existing trajectory information. These future paths would also change rapidly as the classification changed.

The Uber system first detected the pedestrian 5.6 seconds before impact with the vehicle travelling around 44 mph ($71km/h$). The system failed to correctly classify her, initially detecting her as a vehicle, then alternating between a vehicle and 'other' for several seconds, before finally detecting her as a bicycle and on a collision course 2.5 seconds before impact. After further uncertain detections as well as the 1 second 'Action Suppression' period, the system finally reacted to the threat only 0.2 seconds before impact by disengaging and issuing a warning to the driver and finally getting their attention[16].

The vehicle being used as a test platform was a 2017 Volvo XC90. All Volvos come with AEB/FCW as standard equipment. Volvo markets their system as 'City Safety' and the particular version in this vehicle uses fused results from a forward-facing, windshield-mounted camera and a forward-facing, centre mounted radar. It can prevent or mitigate collisions with pedestrians, cyclists, large animals, and other vehicles in the lane of travel. These features were disabled by Uber due to concerns regarding radar interference between the two systems and the fact that the Volvo brake controller was not design to handle conflicting brake signals from two autonomous systems[16].

After the accident, Volvo engineers ran simulation tests to examine how their system would have reacted if it had been active at the time. They simulated 20 variations in the pedestrian's movement accounting for uncertainty in the

available data. They found that the stock system would have detected the pedestrian when they were fully within both the camera and radar's sensor range, which would have occurred at around 50 m. They also determined that the system would have warned the driver 2.5 seconds before impact and AEB would have activated at 1.4 seconds before impact. The Volvo system is designed to brake using its maximum allowed brake force, even if the collision cannot be prevented. With no additional response from the driver, and only considering the actions of the AEB system, they predicted that the factory AEB would have avoided a collision in 17 out of 20 scenarios and in the other three, the impact would have happened at less than 10 mph. One of Uber's changes since the accident has been to allow Volvo's FCW and AEB systems to remain active during their testing after working with the manufacturer to solve both the radar interference and brake controller compatibility problems[16].

The NTSB determined that Uber had an 'Inadequate safety culture'[16]. Additionally, the Arizona state government was found to have inadequate processes in place to review the safety plans submitted by companies intending to test autonomous driving systems in their state.

3.4 Tesla January 22, 2018 California

Figure 3.5: A photo of the accident scene that appeared in the NTSB report brief [17]

On January 22, 2018, a Tesla Model S was operating in Autopilot mode in the High Occupancy Vehicle (HOV) lane of Interstate 405, heading to Los Angeles. The Tesla was following a lead vehicle at a reduced speed (around 21 mph) when suddenly the lead vehicle merged right and revealed a stopped fire truck parked diagonally in the HOV lane that was protecting the scene of an earlier accident. AEB in the Tesla either did not see the truck or did not recognize it as a threat in time and accelerated to the set cruising speed shortly after the lead vehicle disappeared, eventually striking the rear corner of the fire truck seconds later at 30.9 mph, according to data recorded by the car and examined

by the NTSB [17].

I-405 is a divided highway and one that Tesla considers an appropriate roadway type for Autopilot operation. Additionally, the driver admitted to being distracted by the car's radio at the time of the accident, as well as eating a bagel and drinking coffee during the trip. He also admitted to receiving an owners manual when he purchased the car 6 month prior (which explains the limitations of Autopilot and how to use it), but never read it[17].

The preliminary findings of the NTSB in this investigation are similar to those of the 2016 Tesla accident in Florida. They focus on driver misuse of autonomous technology and its interaction with a situation that it was not designed to handle adequately. From examining the data, the NTSB determined that the car did not detect driver-applied steering wheel torque in the last 3 minutes 41 seconds before the crash. During the entire trip that lasted around 66 minutes, Autopilot was engaged for 29 minutes 4 seconds, and of that time, the drivers hands were on the steering wheel for only 78 seconds.

The situation experienced just before the crash, called a 'cut out', is extremely difficult for autonomous driving systems to react to. It occurs when a lead vehicle suddenly merges into a different lane and reveals a stationary or slow moving object in the path of the vehicle with little time for the autonomous system to react. In some cases, such as this, an autonomous driving system may not react to the threat in time to prevent damage. This delay can be due to sensor limitations that require time to build a confident detection of the object, as well as a requirement to limit the number of false positive reactions from the AEB system. These can occur from other stationary objects that are close to, but not in the lane of travel, such as a guard rail or bridge pier.

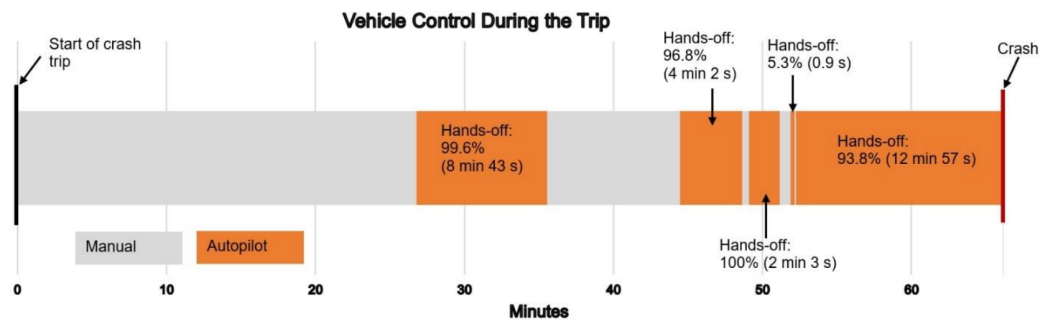


Figure 3.6: Figure from the NTSB report brief that shows the usage of Autopilot during the trip, ending at the time of the crash [17]

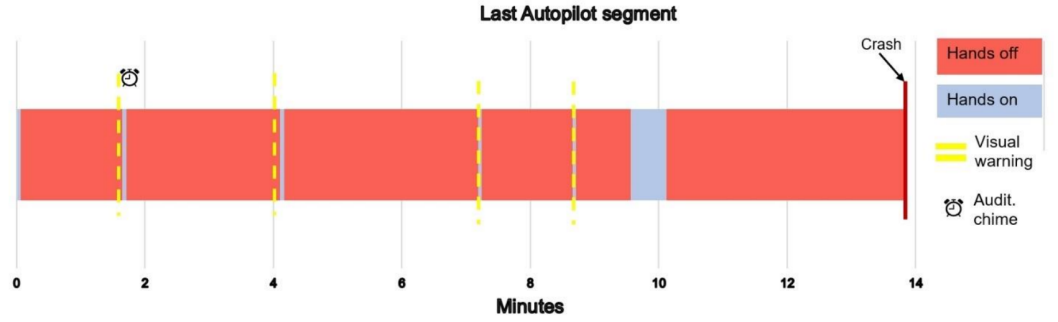


Figure 3.7: Figure from the NTSB report brief that shows the hands on/off status from the data they collected as well as the timings of warnings issued to the driver to keep their hands on the wheel. The data is for the last autopilot segment of the trip, ending with the crash[17].

3.5 Summary

The accidents in this chapter were selected because they demonstrate the complete inability of current generation SAE Level II autonomous driving systems to handle certain scenarios due to improperly mitigated safety hazards in their designs[1]. Each of these scenarios share the common characteristic that they have a low probability of occurrence per drive cycle, and per driver, but are arguably common enough that most drivers will eventually experience something similar, often without severe consequences. When considering the scale of a fleet of autonomous vehicles, all running the same software, and the combined mileage and area that they operate in, these events can no longer be assumed to be too rare to matter.

These are the type of accidents of particular interest to safety researchers because they demonstrate how safety hazards can be exposed in highly specific conditions that are not easy to identify for end users and can easily lead to fatal accidents. Without detailed knowledge of the design hazards of each system there is no way to differentiate these conditions from those where the user has experienced safe system performance. It is increasingly evident that drivers cannot be relied upon to prevent accidents caused by such rare conditions due to the limits of human psychology and attention spans. This therefore demonstrates that **driver oversight is not an adequate means to mitigate hazards in an autonomous vehicle**, even in a level II system where driver oversight is expected by definition. Moreover, in each accident report the autonomous systems were determined to have functioned as designed and each

[1]Even though the goal of Uber's autonomous research efforts are for a vehicle with higher than level II autonomy, it is considered in this thesis to be a level II system due to the requirement of driver oversight, the availability of features, and the quality of implementation of the systems involved at the time of the accident.

investigation, instead focused on identifying the failings of human oversight to prevent the accident, despite also including details of large safety hazards in their design and the ineffectiveness of their driver monitoring capabilities.

The accidents here were also selected because each of them resulted in investigations by government agencies and therefore extremely precise details could be found using publicly available reports. This meant that the information obtainable for them was of a consistent detail and quality.

The rest of this thesis aims to find an alternative method for mitigating these types of hazards that does not rely as heavily on human oversight. This included the identification of specific hazards associated with these accidents using hazard analysis techniques, the results of which are presented in the next chapter.

Chapter 4

Hazard Analysis and System Requirements

This chapter provides an STPA analysis of the hazards of a generic SAE Level II autonomous system, intended to resemble the systems involved in the accidents discussed in the previous chapter, without being specific to either design. Specific focus was put on identifying hazards associated with the root cause of each accident studied. These hazards are then used to develop a set of high-level system requirements that can be used in the designs of future systems to prevent similar accidents.

4.1 Analysis of an Autonomous Vehicle System

STPA is a hazard analysis process initially developed by Nancy Leveson at MIT with the goal of improving system safety and security through the application of systems thinking to deal with the increasing complexity of system design. It is different from other hazard analysis techniques due to its more general

approach to understanding accident causality. STPA and the associated Systems-Theoretic Accident Model and Processes (STAMP) theory state that accidents are caused, not by component failures, but by inadequate control of safety critical systems - for which, component failure is only one possible explanation. It then seeks to identify the potential accidents caused by inadequate control, and the associated hazards so they can be mitigated. By casting a wider net, STPA is able to identify hazards in systems where no component failure has occurred and the hazard is caused by unintended consequences of integrating system components. Application of STPA on a generic SAE Level II system should produce the high-level requirements from which a safer design can be developed.

4.1.1 Notes

Restricted application of STPA

While STPA would normally be applied to the vehicle as a complete system, this chapter will focus only on the autonomous sub-components that are directly related to collision avoidance. An example of an application of STPA to an autonomous vehicle system as a whole was begun by Syed Asim Shah in [19].

Objects of Concern

The phrase Object of Concern (OOC) has been adopted in this thesis as a short hand for any solid object in the environment around the vehicle that should be avoided due to potential for harm if the vehicle were to collide with it. This can include: people and animals, other vehicles, and any solid object dense enough to cause damage to the vehicle.

4.1.2 Brief overview of STPA

STPA involves first creating a control flow diagram of the system from a high level point of view. From this, all control actions in the system can be gathered. Then, after identifying the accidents that you would like to prevent, hazards describing system states that can lead to the accidents are documented. Next, Unsafe Control Actions (UCAs) are determined by applying a series of questions to each control action to discover their potential to lead to a hazardous system state. Finally, safety requirements are developed to prevent the occurrence of UCAs.

More detailed descriptions of the steps involved in STPA can be found in [12].

4.1.3 Step 1: Control Flow

The control flow diagram shown in Figure 4.1 was created to model a generic SAE Level II autonomous vehicle system. It depicts both driver and autonomous control paths for the vehicle along with the necessary feedback. The diagram was made using experience developing a similar system for the McMaster EcoCAR team and is designed to show the interactions between the autonomous system and the rest of the vehicle as well as certain important subsystems, without being too cluttered. This is a simplification made to reduce the amount of STPA work in this chapter that is not directly related to the detection and classification of objects for the purpose of collision avoidance. A full application of STPA procedures would involve a much more complicated diagram, and possibly multiple diagrams of different levels of abstraction, depicting each sub-system of the vehicle in order to capture all possible methods of their

interaction.

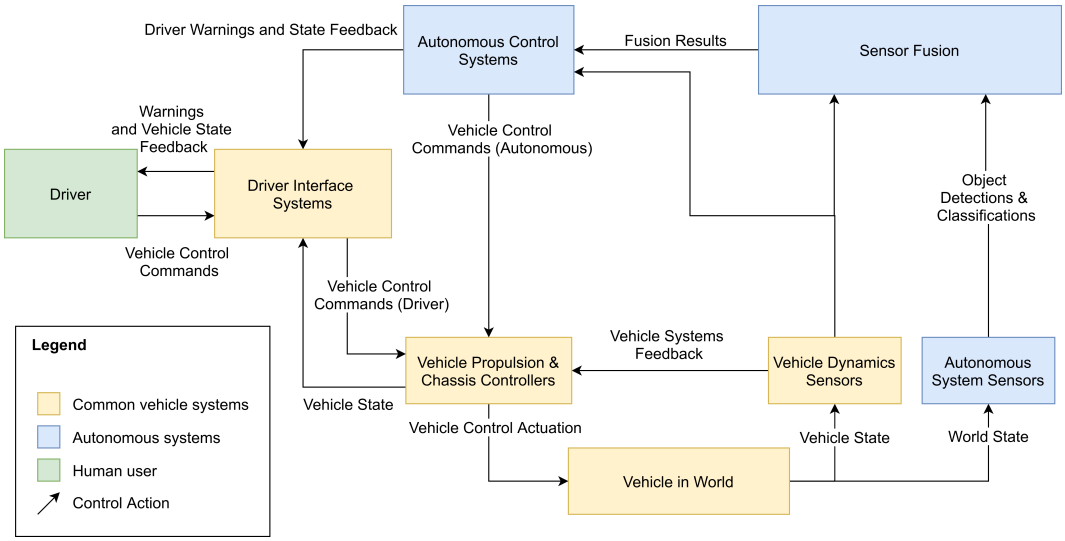


Figure 4.1: Control flow diagram used for STPA analysis

4.1.4 Step 2: List of Accidents

Using knowledge of the system, the next step is to identify the different types of harm the system can cause and list the types of accidents that should be prevented due to their potential to cause that harm. Since this thesis is focused on collision avoidance, the harm to be avoided are injuries and property loss due to a collision occurring, as well as the potential for injuring vehicle occupants if a collision avoidance event is extreme enough on its own. Two types of accidents have been chosen that capture these events. Also note that there is a distinction between a *car accident* and an *STPA accident*, in the latter, *any* undesirable system state that can lead to harm is called an accident.

A-1 The Vehicle collides with another vehicle or object in the environment

A-2 Planned vehicle acceleration (or deceleration) movements are severe enough to injure its occupants, without necessarily causing a collision.

4.1.5 Step 3: System Hazards

The following hazards were determined to be potential causes of the accidents previously listed. The list of hazards includes some of those that are hard to define but are understood by experienced drivers, such as the need to sometimes drive below the posted speed limit if a street is full of pedestrians and parked cars.

[H-1] Vehicle gets too close to an OOC for the given speed and driving conditions, and it is no longer likely that the autonomous system can react sufficiently on its own to avoid a potential collision; including those with stationary objects and moving objects experiencing a change in velocity. *Example: The vehicle should not 'tail-gate' other vehicles and it should not come close to hitting stationary objects. Explanation: Sufficient room is needed to account for errors in object position, vehicle control, and erratic movements characteristic of some objects like pedestrians*

[H-2] Vehicle is not able to react to potential OOCs that are obscured by other objects or that may currently be in a sensor blind spot. *Example: The vehicle should plan on having to react to the sudden appearance of new OOCs when they are likely to occur, such as from behind parked cars, or any time the system cannot detect far enough ahead while controlling the vehicle due to a blind spot or other sensor system limitation*

[H-3] Vehicle violates a safety-related rule of the road while in Autonomous mode. *Explanation: Road regulations are often written in a way that maintains*

safety. In addition, they add predictability that other drivers rely on to avoid collisions

[H-4] Vehicle behaves in an erratic manner while driving. This can include sudden changes in speed and direction without proper signalling, irregular acceleration, or unexpected behaviour for a given driving environment or location

Based on the review of accident reports contained in Chapter 3, it is the author's claim that the root cause of all three accidents was as a lack of adequate hazard mitigation for one or a combination of the above hazards that was the root cause of each accident. In the case of each autonomous system discussed in that chapter, human oversight was the only mitigation strategy provided for these hazards, and each investigation ended with the conclusion that the strategy had failed and was inadequate. Specifically, it is claimed that Hazard **H-1** caused the Tesla accident in Florida (Section 3.2), and Hazards **H-1** and **H-2** caused the Tesla accident in California (Section 3.4). Interestingly, **H-1** was the primary cause of the Uber accident (Section 3.3), and its likelihood of occurrence was probably increased due to unexpected consequences of design additions intended to mitigate **H-4**. This is the specific type of outcome that the STPA process is intended to prevent.

4.1.6 Step 5: Unsafe Control Actions

"An Unsafe Control Action (UCA) is a control action that, in a particular context and worst-case environment, will leave the system in one of the hazardous states defined in Section 4.1.5" [12]

Using the STPA process, unsafe control actions were derived for the system

defined in Figure 4.1, specifically the subset of the system concerned with implementation of autonomous features. In STPA this is accomplished by analyzing whether the following conditions can cause a normally safe control action to become unsafe[12]:

1. Not providing the control action leads to a hazard.
2. Providing the control action leads to a hazard.
3. Providing a potentially safe control action but too early, too late, or in the wrong order.
4. The control action lasts too long or is stopped too soon (for continuous control actions, not discrete ones).

The condition 'poor accuracy' has been added to the third scenario from above. It is the author's belief that much of the safety of an Autonomous Vehicle relies on its ability to accurately sense its surroundings and not enough justification could be found that it belonged to one of the existing scenarios typically mentioned in STPA literature.

In addition, only the control actions emanating from the Sensor Fusion sub-system are considered due to the scope limitations of this chapter.

Sensor Fusion Control Actions

Control Action	Not providing causes hazard	Providing causes hazard	Too early, too late, wrong order, or poor accuracy	Stopped too soon / applied too long
Object Detection	**[UCA-1]** Sensor Fusion does not detect a moving OOC. (**H-1**) **[UCA-2]** Sensor Fusion does not detect a stationary OOC. (**H-1**)	**[UCA-3]** Sensor Fusion detects an OOC that does not exist, causing the Autonomous System to react dangerously. (**H-3** and **H-4**)	**[UCA-4]** Sensor Fusion does not detect an OOC in time for the Autonomous System to safely react to it (**H-1**, **H-2** and **H-4**) **[UCA-5]** The position or velocity information reported by Sensor Fusion about an object is not accurate enough for the Autonomous System to avoid it. (**H-1**)	**[UCA-6]** Sensor Fusion stops tracking an object of concern while it is still close enough to pose a threat (**H-1**, **H-3** and **H-4**)
Object Classification	**[UCA-7]** Sensor Fusion does not provide a classification for an object, when it could help more accurately predict the position of the object in the future (**H-1** and **H-4**)	**[UCA-8]** Object is missclassified by Sensor Fusion, causing the Autonomous System to make incorrect assumptions about position of object in the future (**H-1** and **H-4**)	**[UCA-9]** Sensor Fusion does not classify an object in time for Autonomous System to predict the future position of the object (**H-1**, **H-3** and **H-4**)	N\A

4.1.7 Step 6:Causes of Sensor Fusion Unsafe Control Actions

Causes of the unsafe control actions listed in Section 4.1.6 are given below.

Causes of UCA-1

UCA-1-C1 Sensor(s) do not provide any information about the moving object

UCA-1-C2 Sensor(s) do not provide enough information to produce a corresponding output for the object from Sensor Fusion

UCA-1-C3 Sensor Fusion incorrectly removes information about the moving object from its output

Causes of UCA-2

UCA-2-C1 Sensor Fusion does not receive information about the stationary object

UCA-2-C2 Sensor Fusion does not receive information about the stationary object from the Sensor(s) adequate to produce corresponding output

UCA-2-C3 Sensor Fusion incorrectly removes information about the stationary object from its output

Causes of UCA-3

UCA-3-C1 Sensor Fusion receives detections from the Sensor(s) for objects that do not exist.

UCA-3-C2 Sensor Fusion algorithms incorrectly create output for an object that does not exist.

Causes of UCA-5

UCA-5-C1 Sensor Fusion receives inaccurate or conflicting data about the object state from the Sensor(s)

UCA-5-C2 Sensor Fusion loses accuracy when combining information about the state of the object

UCA-5-C3 Sensor fusion receives inaccurate vehicle velocity information from the Vehicle Sensors when performing relative-to-absolute velocity calculations

Causes of UCA-4

UCA-4-C1 Sensor Fusion does not receive any data about the object in time to produce output for it

UCA-4-C2 Sensor Fusion does not receive adequate data to produce on time output for the object

UCA-4-C3 Sensor Fusion takes too long to process the data for the object, before producing corresponding output

Causes of UCA-6

UCA-6-C1 Sensor Fusion stops receiving detection data about the object from the Sensor(s) due to things such as inadequate sensor

coverage overlap, sensor FOV, environmental conditions, or other sensor failure.

UCA-6-C2 Sensor Fusion starts to receive conflicting or poor quality detection data about the object from the sensors

UCA-6-C3 Sensor Fusion incorrectly stops producing output for the object

Causes of UCA-7

UCA-7-C1 Sensor Fusion does not receive classification information from the Sensor(s) for the object

UCA-7-C2 Sensor Fusion receives conflicting classification information from the Sensor(s) and is unable to reconcile it

UCA-7-C3 Sensor Fusion incorrectly removes classification information from the output for the object

Causes of UCA-8

UCA-8-C1 Sensor Fusion receives incorrect classification information from the Sensor(s) for the object

UCA-8-C2 Sensor Fusion receives conflicting classification information from the Sensor(s) and incorrectly reconciles that information

UCA-8-C3 Sensor Fusion incorrectly modifies classification information for the object

Causes of UCA-9

UCA-9-C1 Sensor Fusion does not receive any classification information for the object from the Sensor(s) in time to produce on-time output

UCA-9-C2 Sensor Fusion does not receive consistent classification information for the object from the Sensor(s) in time to produce on-time output

UCA-9-C3 Sensor Fusion takes too long to process the classification information from the Sensor(s)

4.2 Safety Requirements

The Unsafe Control Actions from Section 4.1.7 are transformed into safety requirements in the following chart. This process involves taking the condition that was found to be a cause of the UCA and inverting it to produce the corresponding safety requirement. The resulting requirement should ensure that the condition does not happen, thus mitigating the hazards associated with the UCA. Similar safety requirements have been combined to reduce clutter and repetition. Each safety requirement is traceable to its associated UCA cause via the third column of the chart.

Where needed, reasoning or additional context and examples have been provided with the safety requirements in emphasized text.

Req. #	Requirement	UCA Causes
SR-1	Sensors must report data for all OOCs	**UCA-1-C1**, **UCA-2-C1**
SR-2	Sensor Fusion must use all input from Sensors to construct output. *All input is useful in determining the state of the system, even faulty information*	**UCA-1-C2**, **UCA-1-C3**, **UCA-2-C2**, **UCA-2-C3**, **UCA-4-C2**, **UCA-7-C3**
SR-3	The Sensor array on the vehicle should be designed to produce a low False Positive (FP) rate of OOCs after Sensor Fusion.	**UCA-3-C1**
SR-4	The Sensor Fusion sub-system should be designed to reduce the rate of False Positive (FP) object detections that are output to the Autonomous Control System.	**UCA-3-C2**
SR-5	The Sensor Fusion sub-system should be designed to reduce the effects of inaccurate object state information from the sensors.	**UCA-5-C1**
SR-6	The Sensor Fusion sub-system should not produce output with less accuracy than the most accurate detection from the Sensors.	**UCA-5-C2**
SR-7	The Vehicle Speed sensor feed should be as accurate as reasonably possible.	**UCA-5-C3**
SR-8	The Vehicle Sensor data should be updated frequently enough to reduce errors when used for sensor fusion calculations	**UCA-5-C3**
SR-9	The Sensor Fusion sub-system should be able to detect and handle non-realistic changes in the vehicle speed data that it receives.	**UCA-1-C3**
SR-10	The Sensor system and the data connections from it shall be designed to allow messages about OOCs to be received by the Sensor Fusion sub-system in time to make an appropriate decision.	**UCA-4-C1**

Req. #	Requirement	UCA Causes
SR-11	The Sensors should be able to produce reliable data about an OOC in time to prevent a collision for the given speed and driving conditions.	**UCA-4-C2**
SR-12	The Sensor Fusion sub-system should be able to produce output for an OOC in time to avoid a collision with it	**UCA-4-C3**
SR-13	The Sensors system shall be designed so that the rate of detection-dropouts of OOCs is reasonably low	**UCA-6-C1**
SR-14	The Sensor Fusion sub-system shall be able to detect premature detection dropouts from the Sensors	**UCA-6-C1**
SR-15	The Sensor Fusion sub-system should be designed to tolerate periods of conflicting or poor quality data.	**UCA-4-C2**
SR-16	The Sensor Fusion sub-system should continue to provide state updates for OOCs as long as there is still data from any sensor for the object.	**UCA-6-C3**, **UCA-6-C2**
SR-17	The Sensor system should be designed to produce classification information at multiple levels of abstractions (eg: car vs. 4-wheel vehicle) *The Neural networks involved can fail to classify an object due to simple things such as obstruction or changes in lighting conditions*	**UCA-7-C1**, **UCA-7-C2**, **UCA-8-C1**, **UCA-8-C2**
SR-18	The Sensor Fusion sub-system should be designed to handle conflicting object classification information. This uncertainty should be reflected in the output for that object.	**UCA-7-C2**
SR-19	The Sensor system should be designed with redundant sources of classification information using different techniques	**UCA-8-C1**, **UCA-8-C2**, **UCA-9-C2**

Req. #	Requirement	UCA Causes
SR-20	The Sensor system should have adequate FOV coverage for both detection and classification of OOCs in an adequate amount of time to avoid a collision	**UCA-1-C1**, **UCA-2-C1**, **UCA-1-C2**, **UCA-2-C2**, **UCA-4-C2**, **UCA-4-C1**, **UCA-6-C1**, **UCA-6-C2**, **UCA-7-C1**, **UCA-7-C2**, **UCA-9-C1**, **UCA-9-C2**, **UCA-9-C2**
SR-21	The Sensor Fusion sub-system should be able to perform calculations relating to object classification in enough time for the Autonomous System to use the information to avoid the object.	**UCA-9-C3**

4.3 Discussion

In the safety requirements developed in this chapter, the concepts of False Positives (FPs) and False Negatives (FNs) appear frequently, however most requirements written here are intended to prevent unsafe control actions related to the absence of, or inadequate detection of an OOC. This makes sense intuitively since the absence of a detection for such an object (a FN) poses an immediate risk to the system due to the inevitable lack control to avoid it, whereas an accident caused by a false positive is less of a guarantee due to the requirement of other factors, such as nearby traffic and vehicle speed. Despite this, collision warning and avoidance systems implemented by manufacturers

are often tuned to minimize the occurrence of false positives or 'nuisance alarms' as a measure to improve user acceptance and to avoid over-reliance on the alarms as a means to engage in non-driving tasks behind the wheel[3]. While the rate of FPs should be minimized in systems implementing these requirements, it is clear that it cannot be at the expense of an increased FN rate.

Chapter 5

Risk Mitigation Through Better System Design

As a strategy to mitigate the safety risks associated with the design and use of sensor fusion systems in autonomous vehicles, this chapter proposes a design process to be followed during the development and integration of such systems.

The ideas in this chapter were inspired by the findings of some recent accidents involving vehicles with autonomous features, where discussions for several of them are included in Chapter 3. Additional inspiration was drawn from the author's own experience designing a sensor fusion system for an SAE Level II autonomous vehicle for the McMaster EcoCAR Mobility Challenge team. These ideas were influenced by the process and discussions found in the 2018 STPA Handbook[12], as well the concept of system monitors as described in [20].

5.1 Design Pattern using Nodes and Monitors

The following section presents a design pattern for a sensor fusion system that aims to reduce the introduction of safety hazards into the system by separating individual algorithms and sources of information into nodes and assigning to each node a dedicated monitor. These monitors produce a confidence score that indicates to the rest of the system the likelihood that the node is functioning correctly and its output can be trusted. These confidence values can be calculated for each output dimension of the node using statistical methods or produced as an all-encompassing heuristic measure.

5.1.1 Definition of Nodes

The design pattern proposed here was based on the idea of the Pipe and Filter software architecture style. In that context, nodes represent filters that either perform individual sensor fusion steps, such as containing a Kalman Filter or performing track assignment, or they represent sources of data, such as sensors. By combining these nodes using 'pipes' one can describe the design of a sensor fusion system in a way that will be familiar to users of Simulink and other model-based design tools.

A meta model was created using the Alloy Analyzer modelling tool in order to formalize the syntax of the design pattern. This was especially helpful when developing the syntax to describe the relationship between nodes and monitors and when verifying the usefulness of the model. The model that was produced in Alloy has been provided in Appendix A.

Sensor Fusion Nodes

Sensor fusion nodes should contain individual algorithms or fusion steps needed in the system, including: signal processing, tracking, and other algorithms. This makes it easier when designing the associated monitor for such a node, since the requirements for the monitor will closely resemble unit test cases for the node, and are a familiar concept for many people with Software Engineering backgrounds.

In the case where an algorithm is used more than once but with different goals (such as a system that contains a variety of similar tracking algorithms for different purposes), there should be a node included in the design for each instance. The reasoning for this again, has to do with the eventual creation of monitor nodes. Since each instance of an algorithm can be an individual source of unsafe control, it should be assigned its own node in the design for individual monitoring.

Sensor Nodes

Sensor Nodes are entry points of data into the system. They can represent both traditional sensors, including cameras, and radar, as well as 'non-traditional' sensors such as internet or GPS based weather and traffic data.

A node should be created for each individual sensor on the vehicle, or other source of system information. This can simplify the development of effective monitors for those sensors and may increase one's confidence in the safety of the system. Even if a vehicle has many sensors of a similar type (radars for example), they are still separate devices with minor, but perhaps important differences in the tasks that they perform for the system. An engineer may be

able to determine a likely and unique failure mode for one of the sensors based on things specific to that sensor. This could include details such as mounting location on the vehicle and how it can affect performance in various scenarios. One example related to sensor location concerns the typical build-up of snow and dirt on a vehicle during less than ideal driving conditions and how that can have different effects on sensors of the same type purely based on their location. By having individual node/monitor reporting for each sensor, this data can be received and analyzed by other nodes in the system in order to extract higher-level information about the current operation level of the vehicle as a whole.

If one wishes to detect large, systemic faults in interconnected sensors systems, such as the failure of more than one radar, you can create another node that processes and monitors the output of other sensor nodes (thus beginning to form a hierarchy of system monitoring).

5.1.2 The Purpose of Monitors

Monitors are the key to improving safety with this design pattern. Their purpose is to evaluate the ongoing performance of each node in comparison to expected norms and produce a confidence measure for each output dimension of the node in an ideal implementation. These confidence values are then combined in the monitors of upstream nodes in order to evaluate confidence in the output of those nodes. This is an important feature and distinction, to the author's knowledge, from other implementations in that there is a requirement of some accounting for the accuracy of inputs to other algorithms throughout the data paths in the sensor fusion system. Without this or other forms of self

monitoring, it is easy for the designer to build a system that is only safe under ideal driving scenarios. The role of each monitor is therefore to determine whether its associated algorithm is producing sane results, through analysis of the algorithm's current output compared to its input, and by taking into account the accuracy of input as reported by other monitors in the system. Based on these findings, the monitor should output its own approximation of the accuracy of its node's output. The intended effects of applying this pattern to a larger system are described in the next section.

The type of information that is computed in a monitor has been left abstract on purpose. It is up to the implementers of such a system to determine what needs to be analyzed in the monitors to assure safety. However, a brief example is provided here for the sake of understanding the concept.

A brief example of monitors

A reasonable goal of a system designer might be to require that the autonomous system is able to detect at run time whether or not it is capable of reacting to events that are likely to occur in the near future, given the current speed and conditions affecting both vehicle handling and perception abilities. This goal is a combination of several of the safety requirements identified in Section 4.2).

One way to accomplish this with the node and monitor approach described here would be to start with a sensor node/monitor pair for each sensor. From this, the system needs to determine whether or not it is able to perceive the road environment far enough in advance to handle reasonable expected events (ACDA).

A proposed method for this would be to analyze the quality of sensor detections over time and at different locations in the effective FOV of the

autonomous system. Changes in detection quality and spatial consistency, especially given known information about the current vehicle speed and steering rate, can be used to indicate that the system may be unable to perceive potential dangers in advance. This type of detection would likely not be able to prevent a specific collision from occurring, but can be used, along with vehicle location and weather data to determine whether it is appropriate for the vehicle to be operating autonomously.

5.1.3 Combining Monitor and Node Output to Derive Confidence in Sensor Fusion Results

When node/monitor pairs are combined to form a sensor fusion system, the structure should closely resemble a forest of trees data structure, where each tree represents an individual sensor fusion feature (such as pedestrian detection). Tree structures are an abstraction used to indicate the flow of data through the system from sensor sources, contained in leaf nodes, through processing algorithms in the branch nodes with the outputs of each feature being presented in the root nodes. These results can then be used in other 'feature trees' or used by other autonomous vehicle subsystems (such as ACC or collision avoidance). An example of such a tree representation is shown in Figure 5.1. This diagram is designed to show the ability for there to be multiple consumers of the same data in the hierarchy, potentially with a variety of performance and safety aspects being monitored. Correlations in the monitor output of multiple nodes can be used to identify potential system wide faults.

The analysis needed to represent the system using monitor nodes and a tree-like structure should provide the means for system designers to easily

identify key sensors and information paths that safety critical features depend on. Additionally, it can become easier to see where the introduction of error can occur when detecting the locations of Object of Concerns (OOCs), which was a key component of the Safety Requirements developed in Section 4.2.

Analyzing the system in this manner might also facilitate application of Fault Tree Analysis (FTA) techniques to identify the risk due to over-reliance on sensors with similar failure modes, such as cameras and lidars which are both susceptible to fog and heavy precipitation. This is, however, not covered in this thesis.

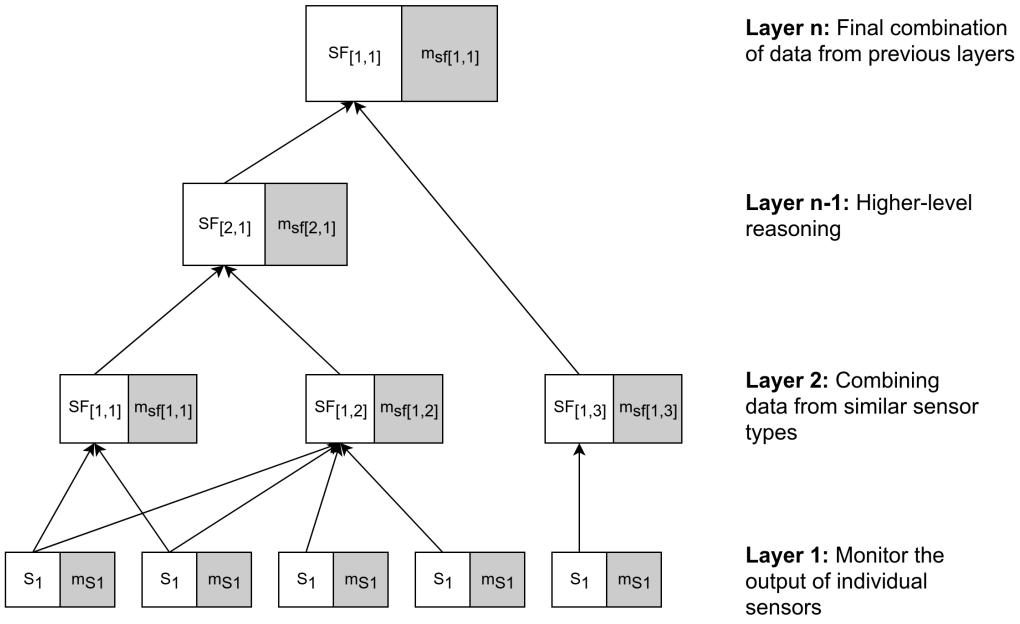


Figure 5.1: An example arrangement of sensor fusion nodes (in white) and associated monitors (in grey).

The remaining chapters explore the application of this design pattern on a simple collision avoidance system in an effort to provide earlier warning to the driver in order to prevent the types of accidents documented in Chapter 3.

Chapter 6

A Practical Example

This chapter outlines work that was done to test the theories developed earlier in Chapter 5 as a means to reduce the likelihood of False Negatives (FNs) in situations such as the events of the May 7th Tesla accident in Florida, discussed in Section 3.2. To accomplish this task, a reference model of an autonomous vehicle system was created in Simulink using virtual sensors, a multi-object tracker, and a pre-built AEB/FCW model provided by Mathworks. This reference system was tested using the scenarios and procedures discussed in Chapter 7 to verify that it performed in a manner similar to that expected from the Tesla vehicle involved in that accident. Monitor blocks were created for each component in the sensor fusion system in order to add functionality that can estimate the likelihood of the presence of false negatives for an OOC in front of the vehicle.

This reference system, its improvements using monitors, and the test results documented in Chapter 7 serve to demonstrate a simplified example of the safety improvements that can be achieved through better handling of abnormal situations, while still maintaining user acceptable functionality under normal

conditions.

In order to mitigate the hazards associated with FN detections, monitor blocks were designed that produce a metric called 'Probability of Risk' that attempts to capture rising 'likelihood' of false negatives. In order to do this, the sensor fusion monitor block processes spurious sensor detections that are not temporally consistent enough to produce tracks in the main sensor fusion tracker block, but may have consistent spacial information. That is, even though they are fleeting, if they consistently appear in the same location they could indicate the presence of an undetected OOC. The frequency of these incidents as well as the consistency of their reported location directly cause a proportional increase in the output of the metric.

Finally, an upper threshold was developed to detect increase in this metric over a short period of time, above which the driver is to be notified. Exceeding this threshold requires a substantial increase in perceived risk over time, and can indicate that the system may no longer be adequately monitoring the road ahead and may require driver intervention and possible forced deactivation of autonomous features. This upper threshold was determined through experimentation with a simulation of the Tesla accident in Florida. The threshold was set at the minimum level that would cause the system to provide early warning in that scenario, but would also not cause false positive activations in other simulations of normal traffic conditions.

The early warning produced here is separate from that of a FCW, such as the one in the reference model, and is tuned to occur before an actual FCW event would be triggered, as well as in some pre-accident conditions that would inhibit such a warning due to the presence of false negative detections due to sensor fusion limitations, such as what occurred in the May 7^{th} accident.

6.1 Recreation of the Tesla AEB/FCW Systems in Simulink

The reference autonomous system was intended to mirror the functionality of the AEB/FCW systems as configured in the Tesla involved in the Florida accident and as described in the investigation reports reviewed in Section 3.2. Features such as Adaptive Cruise Control (ACC) and Lane Keep Assist (LKA) which were present and active on the accident vehicle were intentionally left out in order to simplify system design and testing. Instead, work focused on improving the safety of the AEB/FCW, through the use of monitors, so that the systems are able to recognize and react to these types of abnormal situations.

6.1.1 Components of Reference System

Many components of the reference system are blocks and completed subsystems that were provided by the Mathworks Automated Driving, and Sensor Fusion and Tracking Toolboxes. These were left largely un-modified from their default parameters with some exceptions, such as modifying sensor Field of View (FOV) and detection count limits to more closely resemble the sensors installed in the Tesla. In some cases these parameters are known explicitly, such as those related to the Mobileye used in the vehicle. Parameters for the radar sensor were based on prior knowledge configuring similar simulations for the McMaster EcoCAR team, using their Bosch MMR radars. Full configuration profiles of the blocks used are provided in Appendix B.

Model Subsystems

The main subsystems of the Simulink model were:

- **Simulated Camera Sensor**

  Created using the Vision Detection Generator block of the Automated Driving Toolbox with minimal parameter changes. However, the simulated camera sensor is by default able to detect side profiles of vehicles, which the particular model of Mobileye sensor on board the Tesla was not able to do due to design limitations of that sensor model, see Sections 2.2.1 and 3.2. To make the model more accurate, an additional block was added to remove detections produced by the simulated camera for any simulation objects that are oriented at 90 or 270 degrees to the ego vehicle ±25% at the time of detection. The margin of error was chosen to cover many cases where the side profile of the vehicle is the most prominent side facing the camera.

- **Simulated Radar Sensor**

  Created using the Radar Detection Generator block of the Automated Driving Toolbox. Detections from this block were unmodified, except for the use of the 'Detection Clustering Block' used in some Mathworks examples for the toolbox. This block groups multiple radar reflections from a single object into one central detection using a Global Nearest Neighbour (GNN) approach and is necessary to meet an assumption of the tracking algorithm blocks, regarding the number of detections per sensor, per object.

- **Joint Probabilistic Data Association Multi Object Tracker**

  This block was provided by the Sensor Fusion and Tracking Toolbox. It

uses a Joint Probabilistic Data Association (JPDA) algorithm to assign detections to tracks and was configured with a Constant Velocity Extended Kalman Filter to perform track updates. It was chosen due to its assumed better performance over the standard Multi-Object Tracker block which uses a GNN track assignment algorithm. The effects of using one algorithm over the other were not evaluated, but were not expected to be a factor in gathering results. See Section 7.2 for more discussion on the evaluation of the performance of the reference system before modifications were made.

- **A FCW and AEB Subsystem**

  This was taken from an example used in the Mathworks documentation for the Automated Driving Toolbox [13]. It provides a reference implementation that is used in the model to indicate whether the simulated Tesla system would have reacted to an OOC.

The parameters of the blocks and algorithms used for the reference system were not thoroughly tuned, beyond verifying that the AEB/FCW features were functioning in a manner compliant with the description of the system from the reports in Chapter 3. This was determined by testing the reference system against several simulated scenarios consisting of normal driving situations where no intervention would be expected, and one abnormal situation with a stopped vehicle partially blocking the lane of travel for the ego vehicle. Testing was only performed with highway driving scenarios. More details on the test cases used can be found in Chapter 7.

6.2 Addition of a Monitor for the Sensor Fusion Subsystem

A monitor block was created for the Joint Probabilistic Data Association (JPDA) tracking algorithm in order to improve its safety by attempting to detect the possible presence of False Negative Detections. Its assigned role was to catch OOCs that might exist on the road in front of the vehicle, based on *any* available sensor data, but are not currently being detected by the tracker, thus indicating a possible FN detection. This is in accordance with the primary findings of the hazard analysis and safety requirements produced in Chapter 4, which identified False Negative detections as a major cause of safety hazards.

In order to capture tracks that might be missed by the 'main tracker' a secondary multi-object tracker with a simplified GNN implementation was included in the sensor fusion monitor and was tuned to allow most sensor detections to form tracks (specifically, if detections exist that qualify for a new track for more than 1 update cycle). This is referred to as the 'monitor tracker' and by comparing its output over time with the 'main tracker', the sensor fusion monitor can identify tracks that are possible FN detections for real life OOCs.

6.2.1 Sensor Fusion Metrics

In order to infer the presence of potential FN detection when comparing the output of both trackers, two different metrics were developed. One measures the potential for a FN in a given area around the vehicle, and the other measures an estimate of uncertainty with which the system can rely on sensor readings in the same area.

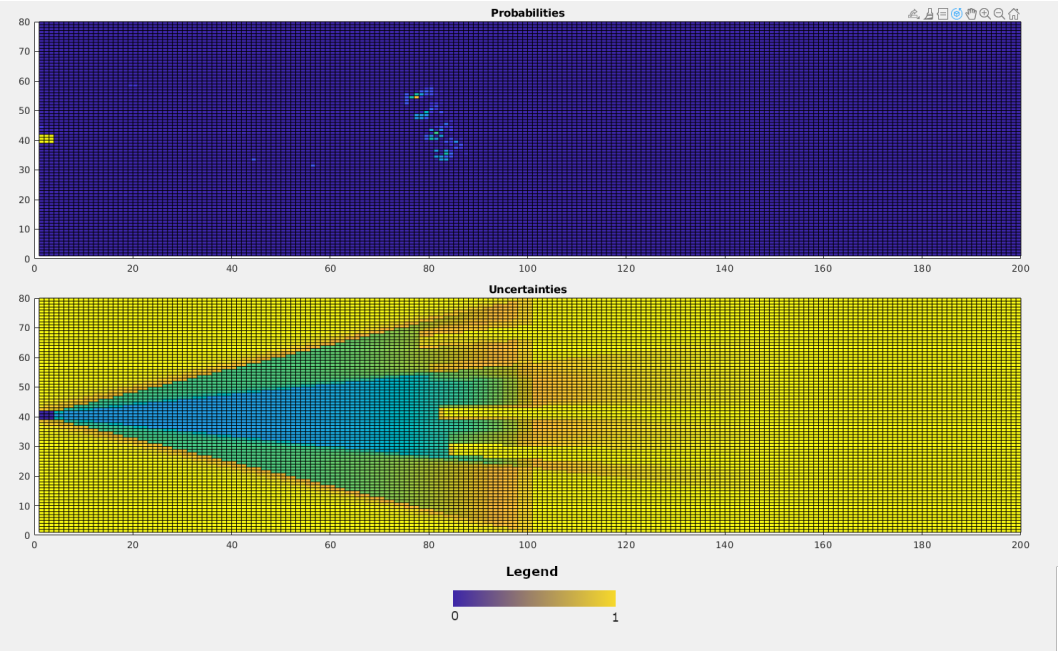


Figure 6.1: Plot results of calculated probability of risk (top graph) and uncertainty (bottom graph) during a test of the example system using a scenario simulating the Tesla Florida accident from Section 3.2. Values are represented on a scale from $[0-1]$ and are mapped to $[blue-yellow]$ by the plotting tool. The position of the ego car is depicted by the rectangle of 6 cells centred at position (40, 0) of each plot; these are added artificially for clarity. Direction of travel of the ego vehicle is from left to right.

These were then calculated by the sensor fusion monitor over a region in front of the vehicle that was subdivided into an 80 by 200 grid of 1 meter cells. The size of the region was chosen because it contains the full nominal FOV of both sensors used in simulation, plus an additional buffer area in the longitudinal direction due to the focus on preventing frontal collisions that is common throughout this thesis.

The dimensions of this region should be altered for different sensors and for different use cases. The cell size of 1m was chosen as a compromise between reducing computation overhead, while providing fine enough resolution that there would be multiple cells within one lane, given an assumed lane width of at least 3m. The layout of the region with respect to the ego vehicle can be seen in Figure 6.1. Additionally, in the uncertainty plot of that figure you can see the relative shape of each sensor FOV region compared to the overall 80x200 detection area. This appears in that figure due to the use of individual sensor FOV in estimating areas of uncertainty for each sensor. It is also useful for understanding the scale of the region, compared to sensor FOV.

The data in Figure 6.1 depicts a time slice from the recreation of the Tesla Florida accident, as perceived by the system described here. In it, there are 4 radar detections produced by the simulated truck. As the simulation progresses, the number of these detections remains consistent, but their detected positions along the length of the truck are very noisy, especially compared to the level of noise from a smaller vehicle travelling normally in a lane of traffic. Due to noise in detection location over time, caused by the large profile of the truck, these detections are not fused into tracks by the main tracker which is tuned to expect much more consistent detection information when performing track assignment.

The lack of track output from the main tracker is eventually what prevents the triggering of a collision warning in the simulated AEB system. This is due to the presence of actual radar detections of the truck that are spatially consistent in front of the car, when adjusted for ego motion, when observed over a longer time scale, and when the limits of location noise are relaxed. The observation of this phenomenon in other simulations, and in recorded real-life sensor data was the inspiration for this method of detecting false negatives.

A human driver when presented with a similar situation (such as occasional glimpses of a road hazard through inclement weather such as fog or rain) would be able to recognize that they should slow down and proceed with caution, even though they do not have a complete view of what it is they are trying to avoid. This behaviour is harder to accomplish with simplified sensor fusion systems involving single object trackers that are tuned for operation in normal conditions, and is the inspiration for the use of monitors as a means to detect the occurrence of this phenomenon.

The metrics developed to overcome these shortcomings are 'Probability of Risk' and its related uncertainty due to known sensor limitations. They are intended to identify areas in front of the vehicle that may contain an undetected OOC using as few assumptions as possible about the objects, such as classification and motion tracking.

In the end, only Probability of Risk was used to detect these situations, as the measurement of uncertainty was not determined to have an appreciable effect on results, compared to Probability of Risk alone, and requires a more intensive study on its potential useage and effects.

Metric Definitions

Rough definitions of Probability of Risk and Uncertainty that can be used to interpret their values are as follows:

- **Probability of Risk** Represents the proportional likelihood that there is a false negative detection and increases with repeated occurrences of sensor detections in a cell without any correlating output from the main sensor fusion subsystem.
- **Uncertainty** A measure indicating lack of confidence in the reliability of sensor detections in a given area. Uncertainty is affected by a sensor's FOV and range limitations, as well as possible effects of occlusion by detected obstacles.

The above terms are not intended to strictly adhere to traditional definitions of probability, or uncertainty; they were however inspired by Dempster-Shafer Theory[18] and Bayesian probability[1][2][10]. This is a simplification that could be eliminated in future work.

Metric Derivation Discussion

The specific equations used to calculate these properties in simulation can be found in Section 6.2.3. Both metrics proposed here are intended only as components of a proof of concept for ideas outlined earlier in this thesis and are the result of a parallel area of research investigation that took place at the same time. There is much more work that should be done in the future to more thoroughly evaluate their effectiveness and compare them to alternate methods. In this work they exist only as part of an example showing the steps

and effects of adding monitors to an existing sensor fusion component of an autonomous system.

6.2.2 Additional Sensor Information

In order to fully utilize the abilities of a monitor node to evaluate node performance, it is useful to know similar information about the sources of information for that node.

In order to provide this, monitor nodes were created for each of the two simulated sensors in order to provide the sensor fusion monitor with required information about each sensor's capabilities. These include constants such as each sensor's FOV, range, and vulnerability to occlusion, and are used by the monitor node to create estimated uncertainty fields based on each sensor's characteristics. For example, uncertainty for a vision based sensor should be 100% in areas behind a possible detection, but less so for some automotive radar sensors that can 'see past' obstacles due to reflections from road surfaces and barriers, see: Section 2.2.

Table 6.1: Data produced by each sensor monitor block

Parameter	Vision	Radar	Source	Description
FOV	$[43.6, 22.6]$	$[20\,\text{deg}]$[1]	Data sheet	Field of View
d_{max}	100m	150m	Data sheet	Maximum sensor range
d_{rated}	Unused[2]	100	Data sheet	Distance of rated detection probability
u_{base}	0.2	0.2	Estimate	Base uncertainty value [3]
u_{occ}	1.0	0.2	Estimate	Uncertainty due to occlusion [4]

[1] The simulated radar block was configured without elevation support. However, in the end only azimuth FOV is used by the sensor fusion monitor.

[2] Some sensor data sheets provide a nominal detection probability, and will also supply a distance at which that was calculated by the manufacturer.

[3] A base value of uncertatinty for each sensor, arbitrarily chosen, but consistent between both sensor blocks used in the model. A system designer could increase this value if they were incorporating sensors of similar types and with known relative detection reliability.

[4] Uncertainty value for cells behind a suspected object in the FOV.

6.2.3 Calculating Probability of Risk and Uncertainty Values

Probability of Risk formula

Probability of Risk is calculated as a sum of 3 different moving averages of an internal 'hit-rate' value. This value is designed to give greater influence to tracks that have existed for more than one update cycle of the monitor tracker.

Due to the nature of the prediction/update scheme of the Kalman filters used in these tracker blocks as well as their internal track assignment logic, this method assigns a higher hit-rate weighting to tracks that are based on sensor detections that have demonstrated predictable position state changes (and are therefore more likely to indicate the presence of real objects, compared to random noise). When a similar track is produced by both trackers, this indicates a potential true positive result from the main tracker, and the hit-rate value is derived using Section 6.2.3

For each track, t_i, produced by the tracker in the sensor fusion monitor, (minus any that are affected by criteria in Section 6.2.3) the following equation is used to calculate a 'hit-rate'(H) value for the cell corresponding to the track's reported location:

$$H_n = \min(1.0, H_{n-1} + [1 - \frac{1}{1 + Age_t}]) \tag{6.1}$$

Where Age_t is the age of the track produced by the sensor fusion monitor tracker, measured in tracking algorithm update steps, and H_n and H_{n-1} represent the current and previously stored hit-rate values.

After this, a final value of Probability of Risk for each cell is calculated as the sum of three moving averages of past hit-rate values in the cell (after accounting for vehicle motion) with window lengths of 10, 20, and 50 update cycles each. With the sample rate used during testing of 0.1s, these correspond to moving averages with windows of 1, 2 and 5 seconds. These values were chosen empirically based on their effectiveness when testing against the simulation of the Tesla Florida accident.

Reducing Risk in the Presence of a Track from the Main Tracker

Each track that is produced by the monitor tracker is compared to the output of the main tracker to identify tracks that are possible True Positives (TPs). That is to say that they have been detected by the main algorithm and therefore might not pose a FN risk to the system as they will probably trigger a FCW/AEB event if necessary. The selection requirement used for these 'true positive' tracks requires that they be within 2m of each other, this was chosen through observations of typical deviations between the outputs of both tracker types under known true positive conditions as well for its relative size compared to the 1m cell size. When this is the case, an alternative formula for calculating hit-rate is used, shown in Eq. (6.2). In this formula, a low hit-rate value is still registered in order to account for the possibility that there is still hidden danger. The value calculated from Eq. (6.2) is set as the hit-rate value for the cells corresponding to each track and increases with their separation distance. This means that the lowest possible hit-rate for a TP detection requires that the tracks overlap in the same cell region. One advantage of this is that a small number of TPs with $> 1m$ of separation will not have a large effect on total risk, but a large number of TPs with non-negligible errors will cause a larger increase in calculated risk.

$$H_n = 0.05 \times \Delta^2 \tag{6.2}$$

Where Δ is the vector norm distance between the two tracks, and 0.05 is the scaling factor chosen to provide a low, but non-zero hit-rate.

Uncertainty Formula

Base uncertainty is calculated for each sensor to account for reasonably expected variation of detection accuracy accross the sensor's FOV. This is calculated using Eq. (6.3) with Δ_{long} representing the longitudinal component of the distance of a cell from the ego vehicle origin. The formula is essentially an inverted and shifted 3^{rd} order Butterworth low-pass filter, with the -3dB point located at the d_{rated} value for each sensor and defaults to d_{max} when d_{rated} is unknown. This was chosen for its ability to simulate relatively low uncertainty up until d_{rated}, after which it models a steep increase that might be reasonably expected. A more accurate model can be hard to achieve due to a lack of detailed information that is common practice with third-party automotive sensors, though parameters can be estimated through empirical testing using a hardware sample.

$$U^2_{sensor} = 1 - \frac{(1 - U_{base})^2}{1 + (\frac{\Delta_{long}}{d_{rated}})^6} \tag{6.3}$$

For each cell within the FOV

Base uncertainty is defined as 1.0 (total uncertainty) everywhere outside of the sensor's FOV.

Finally, total base uncertainty for a given time step is calculated in Eq. (6.4) using the results of Eq. (6.3) for each sensor and the previous total uncertainty value for the cell, after accounting for ego vehicle motion, see Section 6.2.3. Base uncertainty can be added to at run-time using factors such as the estimated effect of occlusion from known true positive obstacles. These details can be

derived from the output of the monitors for each sensor.

$$U_n = U_{visionBase} \times U_{radarBase} \times U_{n-1} \tag{6.4}$$

Accounting for Ego Vehicle Motion

In the implemented algorithm, the previously computed values of uncertainty and hit-rate, which are used in the calculation of future values, are stored in state matrices corresponding to the region of cells in front of the vehicle being examined. As the vehicle moves, the absolute position of the region in the world changes.

In order to account for this, the internal state representations are shifted at the beginning of each time step to account for ego vehicle motion since the last update. This allows spurious sensor detections from the same absolute world coordinates to be clustered over time in the same region in vehicle-relative coordinates. As more of these detections occur in the same motion-adjusted region, the magnitude of the risk will increase for the cells of that region, leading to the identification of possible false negatives in the presence of this type of detection noise.

This method is best suited for detecting false negatives for stationary OOCs, but is shown in Chapter 7 to also have potential to detect moving OOC false negatives when relative speed is low.

6.2.4 Detecting False Negatives Using a Monitor for the AEB/FCW Node

Since there is now the metric of Probability of Risk which is designed to increase when the likelihood of FNs increases, all that is needed to reduce hazards associated with such an increase is to define a threshold above which the driver should be notified.

One process for doing this that was derived experimentally is as follows:

1. Multiply the matrix of Probability of Risk values produced by the sensor fusion monitor by a window function of the same size in order to scale risk values directly in the path of the vehicle, or other areas of interest as deemed necessary.

2. Calculate the matrix sum of the scaled Probability of Risk values.

3. Maintain short-term and long-term moving averages of these sums using 3 and 5 second windows respectively. These values were determined by evaluating the performance of the algorithm during testing against a recreation of the accident in Section 3.2, as discussed in Chapter 7.

4. When the short-term moving average moves above the long-term, with a scaling factor, send a warning to the driver to take over from the autonomous system as this indicates a sudden and large increase in the risk of false negatives. The final value used for this was a factor of 1.2 multiplied by the short-term average, and it was required that the pattern be maintained for > 1 second. These values were determined experimentally using both the accident recreation scenario (where the threshold should be exceeded) as well as several normal highway traffic

scenarios to confirm that the method does not cause unwanted false positives in at least those specific 'normal' conditions. In future, this value should be evaluated with more rigour, including a study of its effects as it is changed, as well as additional testing from a greater variety of traffic scenarios, such as city driving, parking lots, etc. Additionally, there may need to be other values for different driving conditions identified based on factors such as speed, traffic density, and weather. Furthermore, a scaling factor may not be necessary if more effort is put into studying the effects of different time windows on the moving averages.

Discussion

One result that was noticed while testing this part of the system was that a sudden increase in detectable objects, such as rapid increases in traffic density (common in some types of generated test cases used in this domain), can lead to false positive warnings without careful parameter tuning and test case validation. This is due to the use of moving average windows that were only several seconds in length, and the fact that even a normally functioning sensor fusion system will take many time steps to detect, confirm, and output a true positive track. This length of time in a typical sensor fusion system on an autonomous vehicle is commonly in the range of $[0.05s - 1s]$ and is highly variable between different systems due to differences in system requirements from the manufacturer based on the different feature sets and capabilities intended for each vehicle model. These requirements affect the selection of sensors and communications buses with different refresh rates, latencies, and data throughput capabilities; eventually also affecting tracking algorithm selection and parameter tuning.

In the case of the model used here for testing, the total output latency of

the monitor tracker from the first detection is much shorter than that of the main tracker, and many detections that will eventually produce true positives will initially be counted as suspected false negatives for a few frames. A large number of them at roughly the same time can be enough to exceed the threshold. When making test cases for this thesis, specific effort was put into reducing unrealistic increases in object density due to over-simplification of the scenarios, and ensuring that scenario fidelity was as consistent as possible throughout the duration of each test case.

The method of using moving averages and a threshold was chosen for its ability to handle the noise inherent in these types of detections, as well as the ability to compare current conditions to the recent past on the same road. This should be a good method to measure increased risk as it measures relative risk increase and can normalize to the 'background noise' of a particular stretch of highway, such as that caused by different environmental conditions, and different patterns of common road side objects such as buildings, bridges, pylons, and barriers. The same approach, but over a longer time span may be useful in comparing system performance over several drive cycles, as well as detecting gradual system deterioration over the lifespan of the vehicle.

6.2.5 Warning Output to the Driver

The modifications made to the system through the use of monitors were designed in consideration of the known limitations of the reference system and are intended to complement its defects. The warnings produced by analyzing the output of the sensor fusion monitor node should be used in addition to those already produced by the AEB and FCW subsystem components. The

appearance of this new warning should resemble those already used for driver take-over requests.

6.3 Discussion

This chapter describes only one method of mitigating the risks associated with FN detections, there are likely others, especially in the machine learning domain. However, due to the black-box nature of many ML techniques, it is difficult to rigorously prove the safety of a system that relies heavily on their usage in safety critical features. At the time that this work was performed, this was still an active research topic with very few available results directly applicable to automotive use cases and industry procedures.

There is an opportunity in the future to use ML optimization techniques for exploring the parameter space of the equations in this chapter. However, this was beyond the scope of this chapter, and is left for future work. Additionally, any parameters chosen still need to be justified when making claims about the safety of the system. This is typically required when assuring system safety for safety-critical features in the automotive industry, which currently follows the ISO 26262 standard for this process.

The formulas for Probability of Risk and Uncertainty, as well as the method of using moving averages were developed specifically to prevent the type of accident discussed in Section 3.2. The desired effect of this chapter is to demonstrate an ability to prevent this type of situation from happening, without drastically increasing system cost. Therefore, most parameters in this chapter were tuned by hand through repeated testing against a simulation of that accident, followed by verification of other qualities of the system using control

simulations where no intervention was expected. The values of uncertainty were not used in the final system model used for testing in the next chapter as they were found not necessary to produce an adequate warning, and were removed for simplification of results. They are left in this chapter because they may be useful in a related system or future work.

Additionally, the time lengths used for the moving averages in this chapter were determined through rapid experimentation and were chosen for the ability to quickly detect hazardous FN conditions to prevent accidents. There may be a benefit to lengthening these time scales in order to detect long-term risk trends, and potentially using the results from multiple time lengths to evaluate the recommended availability of autonomous features.

Chapter 7

Testing

A main goal of the work in this thesis has been to prevent the types of accidents seen in Chapter 3 through a better understanding of system hazards, leading to improved system designs. As part of this process, several accident scenarios were defined that were designed to expose latent vulnerabilities to the same hazards that lead to the accidents in that chapter, including one such scenario that is an accurate recreation of the accident in Section 3.2. The scenario was made using all measurements, photos, and event sequencing information available in the reports of the resulting NHTSA and NTSB investigations. These simulated accident scenarios were used during the development of the refined system design described in Chapter 6. They were used to evaluate the effectiveness of the refined system's ability to detect hazardous situations that are ignored by the Automated Emergency Brake (AEB)/Forward Collision Warning (FCW) components of the reference system it was built on.

Several additional accident scenarios were created to represent 'control' situations in order to test three key aspects:

1. The continued ability of unmodified AEB/FCW implementations to detect

pre-accident conditions in the presence of reliable sensor detections and sensor fusion results.

2. The ability of the additional monitoring components to correctly filter and prevent false alarms in normal conditions when no alarm would be expected.

3. The interactions of the existing AEB/FCW functionality and the new monitor based warnings in various pre-crash and normal conditions.

The simulated accident scenarios used for testing were made using the Mathworks Driving Scenario Designer App and are compatible with the virtual sensor blocks used in the model produced for the reference system and its improved version. This tool-chain is one of the supported use-cases of the Automated Driving Toolbox provided by Mathworks.

7.1 Notes on Testing Methodology

The overall approach for testing the created system involved running the model of the vehicle with both the stock FCW and the new early warning system enabled and comparing when/if they provide their respective warnings. The systems were implemented in the model in a way that they do not influence each other's results. Additionally, the model used simulated sensors and was tested against highway situations with a variety of traffic densities. Some tests, such as the one shown in Figure 7.1 were intended to trigger both systems, others were meant to simulate 'normal' driving situations. The configurations for the simulated sensor blocks can be found in Appendix B. Within the model, data gathering points were created to analyze the performance of the

monitor functionality and to compare the timing of monitor-initiated warnings vs base-line FCW events, if any.

The implementation of the early warning system makes use of some moving average filters with windows of several seconds as part of the threshold detection scheme that initiates an early warning. In order to ensure that the system is ready before each test case began, each driving scenario was lengthened by at least 5 seconds. This involved lengthening the road the ego vehicle is travelling on leading up to the area or situation of interest, and replicating traffic conditions along the new stretch of road in an attempt to provide a consistent traffic scenario for the model to stabilize against.

7.2 Validating the Unmodified Reference System

Regular functioning of the AEB system was verified using an example of highway driving on a road with a centre barrier and a stopped vehicle partially blocking the lane of travel for the ego vehicle. It is shown in Figure 7.1.

This is an edge case test for an AEB system due to the fact that the stopped vehicle is only partially in the lane of travel and thus makes it difficult to determine whether or not it poses a threat within the margins of error in the sensor readings.

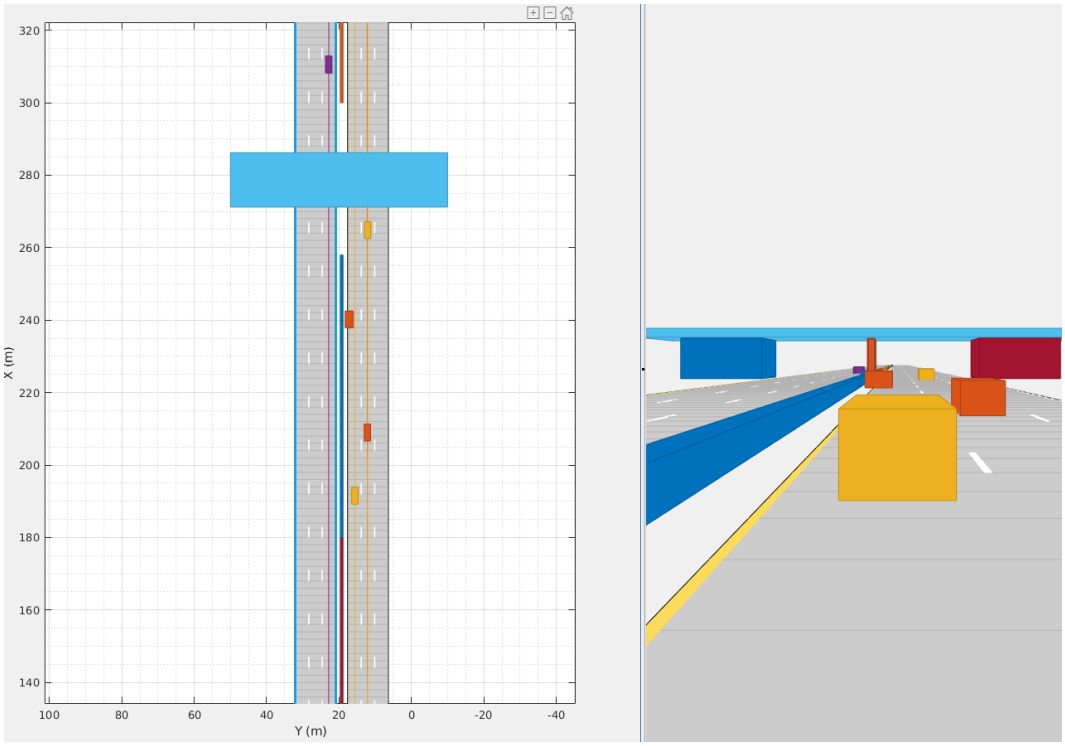


Figure 7.1: A test case designed to test AEB and FCW algorithms, partially resembling the events of the accident in Section 3.4, but without the additional lead vehicle

The results of this test are that a FCW event occurs 5.4s into the simulation, with the actual collision happening at 7.7s, providing both the vehicle and the driver with only 2.3 seconds to react. This may not be enough time for either to prevent a serious accident due to the difficulty of determining the vehicle's location. Part of the reason for the late warning is due to the saturation of the limited number of radar detection slots with detections from the centre barrier in the scenario, a real-life problem that designers of automotive radars, and the systems that use them are faced with. Nevertheless, the reference system FCW component does eventually detect the stopped vehicle and issue a warning to the driver.

Additionally, in this scenario, the output of the sensors and that of the sensor fusion system were in agreement and the new early warning system was not triggered. This can be seen in Figure C.1 in Appendix C, where the average risk begins to decrease at around 5 seconds, corresponding to the first instances of the main tracker detecting the stopped vehicle. Because of this intentional functionality and the radar saturation mentioned earlier, the monitor system is not able to provide any additional early warning.

The reference (existing AEB) system was also tested against the recreated May 7^{th} accident, before modifications were made. The system failed to recognize the truck consistently, and failed to perceive the truck as one object. Instead it was detected as several evenly spaced objects along the length of the actual vehicle. This is due to the absence of any vision sensor detections to provide a measurement of object width (from the camera's perspective), and the configurations of both the radar sensor and tracking algorithm that do not anticipate large cross-traffic vehicles (as is common with systems similar to what was installed in the Tesla, as indicated in Chapter 3). Since the vehicle was detected as multiple objects and there was no real detection of the true size of the obstacle, the AEB/FCW subsystem was unable to identify a specific target in the lane of travel of the ego vehicle, and therefore provided no warning. This is consistent with the accident events.

7.3 Evaluation Against the Recreation of the May 7th Tesla Accident in Florida

In order to test against this accident, sensors on the simulated vehicle were configured with parameters similar to those installed on the Tesla, or using best guess estimations where no reliable information could be found. The recreated accident and the sensor FOVs can be seen in Figures 7.2 and 7.3.

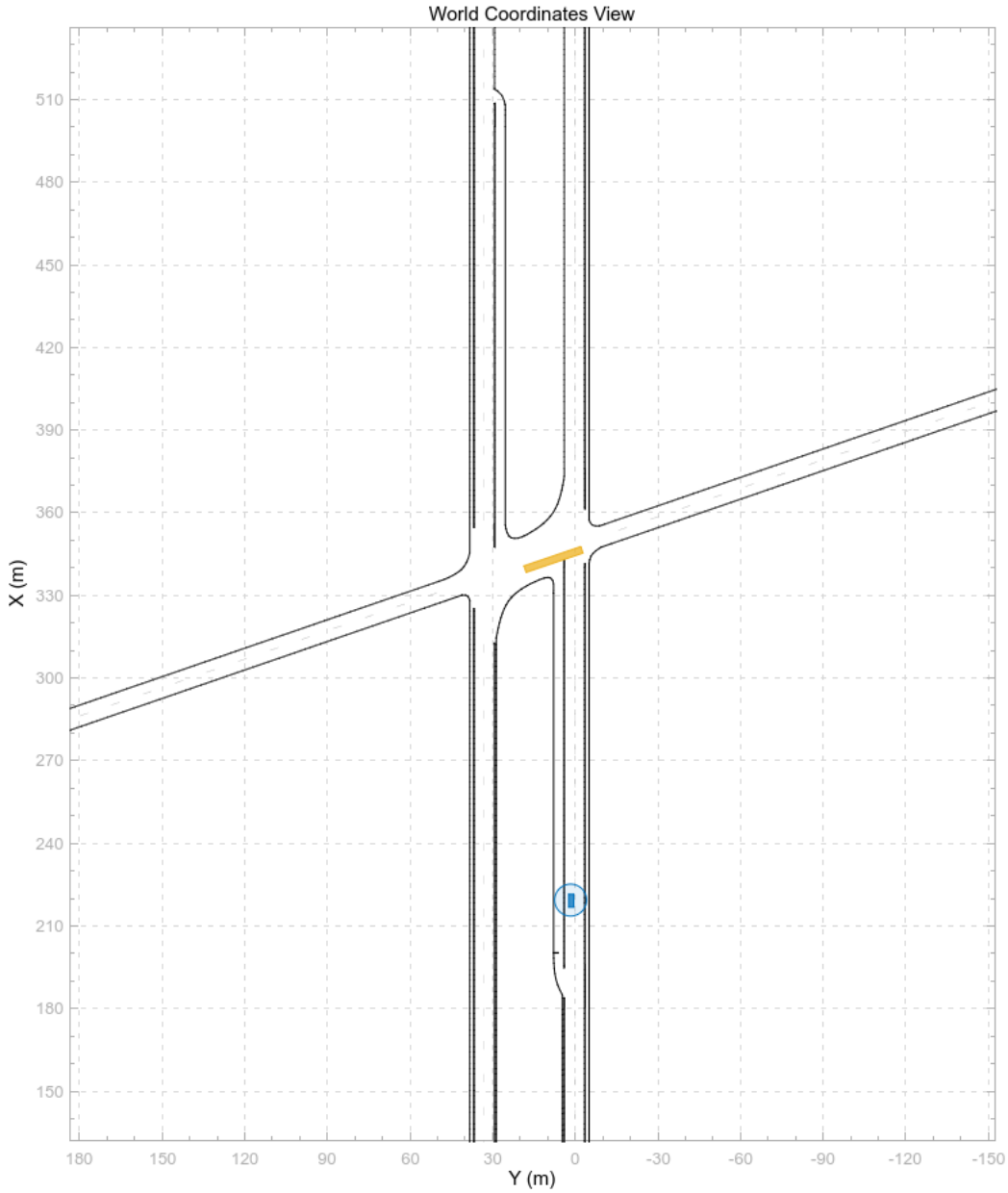


Figure 7.2: A recreation of the May 7, 2016 Tesla accident, documented in Section 3.2

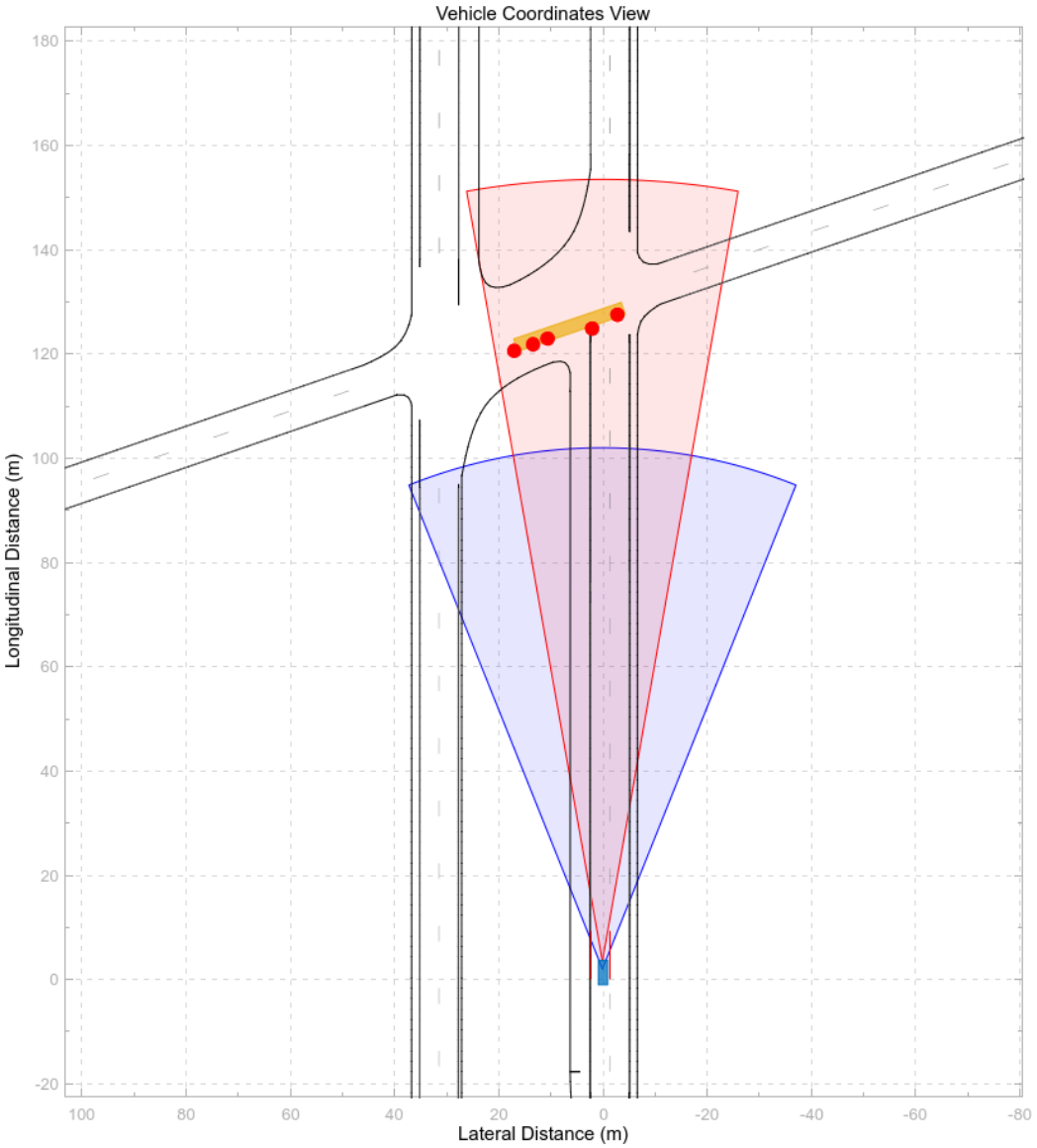


Figure 7.3: The sensor array on the virtual test vehicle, comprised of only long range radar and a forward facing camera detector. The red dots shown are simulated detections from the radar sensor and the red and blue 'pie slices' represent the radar and camera FOVs, respectively.

In this test, the 'stock' AEB system in the model was not able to produce any confirmed tracks representing the truck. This was due to the variations in location of the radar detections of the truck. These can be seen in Figure 7.2.

While they are regular enough for a human to identify an object with at least a straight edge, they were too noisy for the sensor fusion node, and were not adequate for it to confirm any of the candidate tracks it produced. Therefore, the AEB and FCW systems did not have any information in order to take action, similar to the real accident.

Additionally, the new early warning system was able to detect a high risk situation due to the presence of consistent sensor detections and a lack of any associated sensor-fusion object tracks. The system produced a warning approximately 4 seconds before the accident. This is a large improvement in a scenario where no other warning would have normally been issued to the driver.

7.3.1 Unexpected Results

There was an unexpected warning produced by the new early warning system on a test case that consisted of a car passing the ego vehicle on the right, going slightly faster. There were no other objects in the driving scenario capable of being detected by the simulated sensors, making it a low density traffic scenario. In this test, an early warning was issued in two cases:

1. Due to the low amount of other traffic and obstacles in the scenario, a small number of erroneous sensor detections was enough to trigger a warning at around 12 seconds.

2. As the other vehicle passed into the view of the sensors on the ego vehicle, there was a point where it was detected by both the camera and the radar, however, each sensor detected a different corner of the vehicle. This produced detections that were too far for the sensor fusion algorithm to combine consistently into one track. This can be seen in Figures 7.4

and 7.5

The first unexpected detection could be prevented with an early warning algorithm that better handles low traffic volume scenarios.

The second one is actually a result of a flaw in the configuration of the main sensor fusion algorithm and could be corrected using different tuning when combining tracks near the edges of sensor FOV, or with the use of corner radars in order to 'see' the passing vehicle earlier.

While the warnings produced in this test-case were unexpected, they help to determine potential future work for this type of system, and demonstrate its effectiveness, in the case of the second warning, by identifying a weakness of the tuning of the object tracker in the model, that had previously gone unnoticed.

7.4 Discussion of Results

There were several interesting outcomes of this testing, including the appearance of monitor-initiated warnings in a scenario intended to be a 'control' example. Further inspection revealed that one of them was in-fact caused by a specific case of ineffective object tracking by the main sensor fusion node that was previously unnoticed by the author. This unexpected result reinforced belief in the functionality of the monitor warnings since they correctly identified a case of false negative detections.

Testing also revealed that the combined system maintains correct FCW functionality under at least one normal condition scenario and there were no false alarm warnings issued by the monitor subsystem during this specific test.

Only a few tests were conducted in total, and while they were important edge cases, many more would need to be done before this system could be used

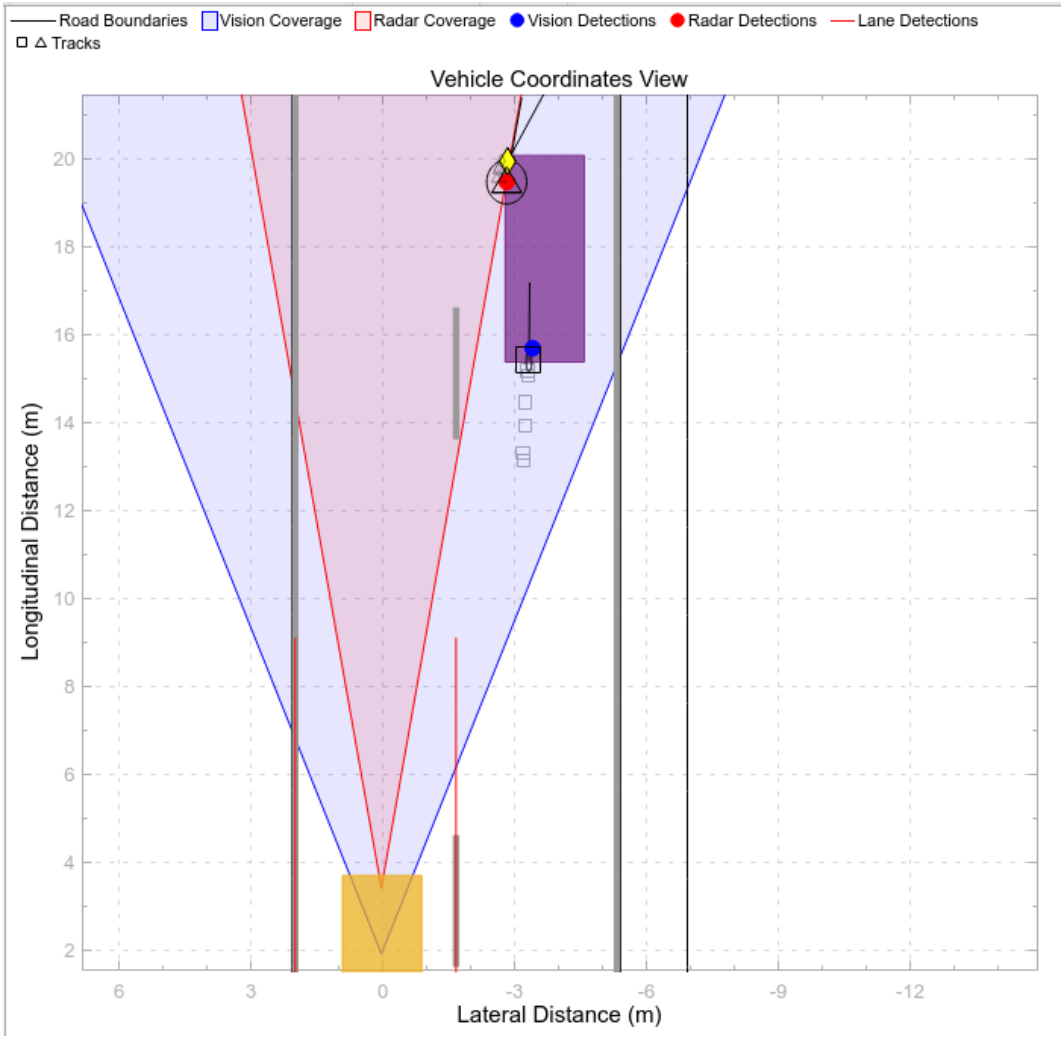


Figure 7.4: The conditions leading up to the second unexpected result. In the image you can see 2 different detection locations on the vehicle. Due to its close proximity, it straddles the detection coverage limit of the camera sensor and therefore the detection produced by the camera is too far from that produced by the radar and the sensor fusion algorithms do not produce a consistent detection location. This, over time increases the risk measured by the early warning system which then triggers a warning. In the image, squares represent confirmed detections produced by the base system, triangles are suspected detections that have not been confirmed, and diamonds are those identified by the early warning system as potential false positives.

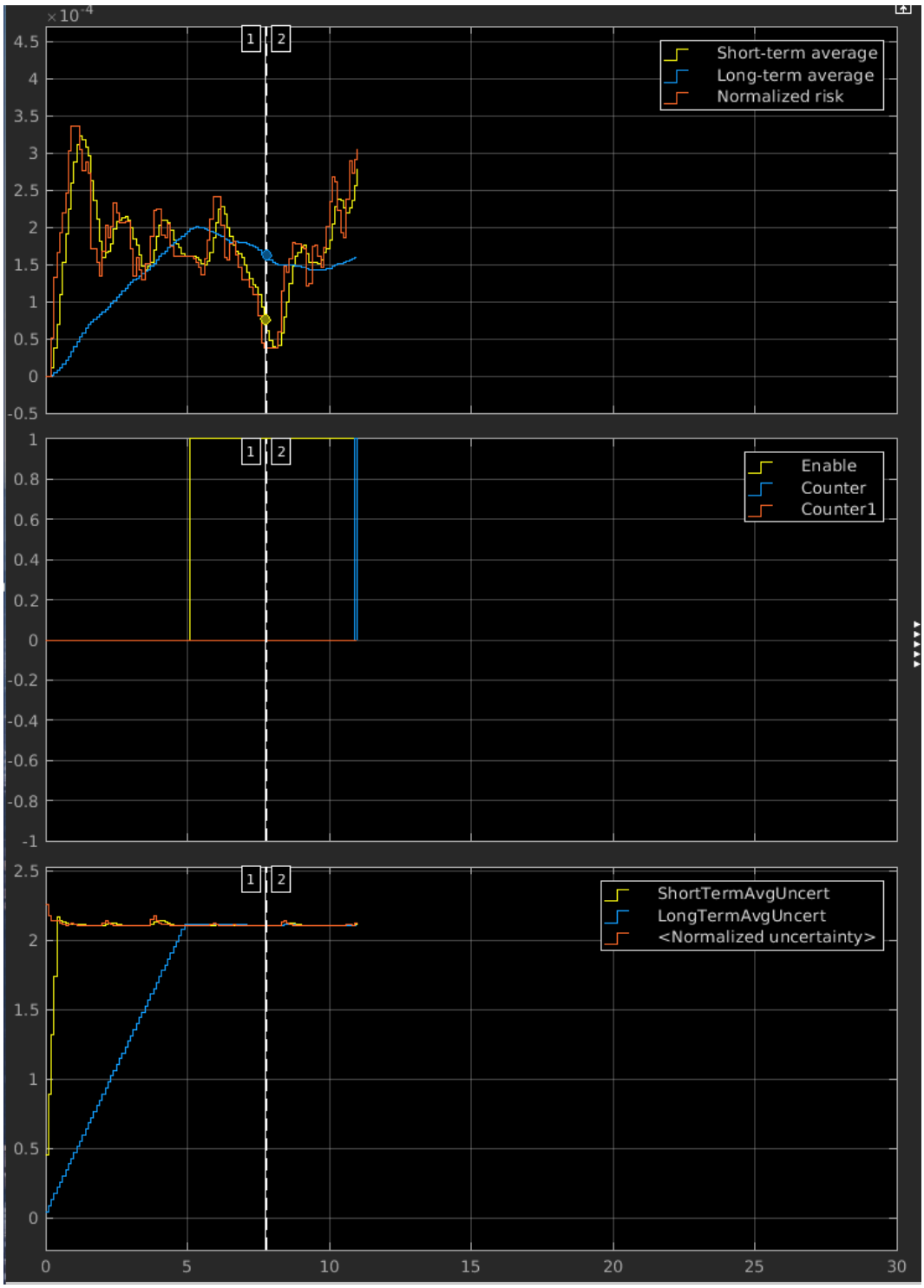


Figure 7.5: The level of risk measured due to inconsistent sensor and sensor-fusion readings

safely.

Finally, when testing against the recreation of the all-important May 7^{th} accident in Florida, the modified system was able to recognize the presence of a large number of false negative detections and would have alerted the driver up to 4 seconds before the collision, despite the absence of any FCW or AEB output, as is consistent with accident events. This would have provided the driver of that Tesla several seconds to react, possibly allowing him to brake or swerve to avoid the truck in his path and at the very least reduce the severity of any potential accident.

Chapter 8

Conclusions & Future Work

This thesis has outlined a new software architecture for use in designing sensor fusion systems for autonomous vehicles. It was inspired by several important early accidents involving SAE Level II autonomous vehicles that highlighted surprising safety hazards in their design that were justified by an over-reliance on operator oversight that was not properly implemented or enforced. In order to help prevent future accidents, it identifies several hazards common to all autonomous vehicles and compares the results of a proof of concept implementation of this architecture against a reference system that has similar design limitations to the vehicles involved in the accidents studied.

In the process of developing this new system architecture, we reviewed several key accidents that involved the use of autonomous vehicle systems and resulted in loss of life or property damage. In reviewing these accidents, it was discovered that, despite obvious flaws in the designs of these systems that were important contributing factors in the causes of these accidents, the resulting investigations instead focused on operator misuse of those systems and ineffective human oversight.

Based on these results, this thesis presents an architecture that mitigates these hazards, without significantly increasing system cost or requiring additional sensors.

To help accomplish this goal, a hazard analysis was performed using STPA, producing safety requirements that were used to influence the creation of a design pattern capable of mitigating the identified hazards, without heavy reliance on human oversight.

The proposed system architecture promotes the conceptualization of sensor fusion systems as tree-like structures that form a natural hierarchy based on levels of data abstraction already present in common sensor fusion design patterns. The architecture mandates the creation of monitor nodes designed to complement each sensor fusion process with the goal of detecting abnormal behaviour that can negatively affect system safety.

An example system that implements the identified safety requirements using the new monitor based architecture was developed and tested against several accident and normal driving scenarios. The results demonstrated that this new architecture shows promise in mitigating the types of accidents studied.

8.1 Future Work

- The implementation of the example system incorporated several simplifications, one of which was the limitation of ego motion to simple linear scenarios. This should be expanded to cover all types of possible motion including turning and speed changes. Additionally, there may be implications for this system design in scenarios involving extremely hilly terrain and other issues requiring height measurements.

- The parameters used in the proof of concept system need further study into their effects on system safety as a whole, outside of specific edge cases, and would benefit from more rigorous tuning practices.

- An additional benefit of mapping sensor fusion systems in a tree-like structure is ease of identifying data dependencies and over-reliance on a small number of sensors. There may be opportunities to develop tools that can assist developers and system designers in working with these models to identify these scenarios.

- There is opportunity to utilize more of the sensor characteristics that are known about specific models, such as susceptibility to adverse weather conditions, limitations of object classification, and common mode failure mechanisms that can affect detection reliability in multiple sensors simultaneously. When a specific sensor weakness is known, such as reduced detection quality in heavy precipitation, a well defined data flow such as what is described here can help identify its impacts throughout the system.

- More formal definitions of Risk, Probability and Uncertainty could be developed for the construction of monitors. In addition, other methods for detecting false negatives should be explored, as well as ways to limit the number of false positives.

- A simple window function was used to emphasize risk values directly in front of the vehicle. A more useful implementation would utilize a dynamic window function that accounts for potential future vehicle motion and specific driving scenarios, such as negotiating curves and analyzing

intersections.

- There is a lot of room to better tune the parameters used in the example system as well as alternative implementations or definitions of risk. This will improve the rate of false-negatives and false-positives (which were not thoroughly tested in this thesis due to time constraints). These efforts will increase system usability for vehicle operators and help build positive trust with them that encourages safe use.

- Many of the parameters in this thesis were tuned by hand in the interest of time and the limits of scope. There is an opportunity to leverage machine learning techniques to automate their tuning and evaluate their effect on system performance via parameter sweeps and ML optimization methods such as genetic algorithms.

Appendices

Appendix A

Meta-model

The following code describes a meta-model created in the declarative specification language Alloy. It was used to help formulate my ideas for Chapters 5 and 6, specifically the structuring of sensor fusion systems as tree-like structures.

```
abstract sig Process {
src:     set Process,
output: some Data
}{}
abstract sig Data {
}
sig Confidence {
value: Int,
}
sig Uncertainty in Confidence {}

abstract sig Monitor {
monNode: one Node, // monitored Node
```

```
monProcess: one Process, // monitored Process
inputConfidence: some Confidence,

confidenceOfProcess: Process -> some Confidence,
confidenceFcn: Confidence -> Confidence -> some Confidence,
confidence: disj some Confidence
}{
confidence = \
        confidenceFcn[inputConfidence][confidenceOfProcess[monProcess]]
#confidence = #monProcess.output
}

abstract sig Node {
process: disj one Process,
monitor: disj one Monitor,
depends: set Node
}{
// No cycles
this ! in this.^@depends
process.src = depends.@process
monitor.monNode = this
monitor.monProcess = process
Node.@monitor = Monitor
}
```

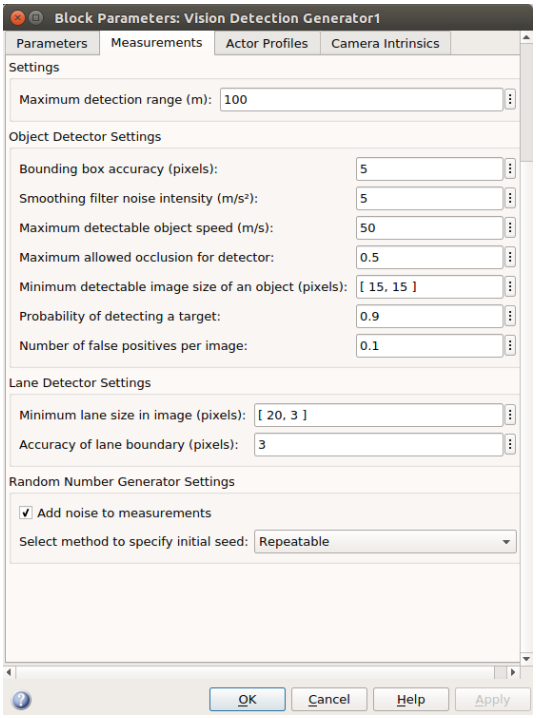
```
one sig CollectNode extends SFNode {}
{
// ColectNode can't depend on a sensor node
SensorNode & depends = none
/* CN depends on all other nodes through
reflexive transitive closure (CN is the root of the tree) */
Node in CollectNode.*@depends
}


sig SFMonitor extends Monitor {}{
inputConfidence = monNode.depends.monitor.@confidence
}
sig SensorMonitor extends Monitor {}{
inputConfidence in Uncertainty

}

sig SFNode extends Node {} {
some depends
process in SFProcess
monitor in SFMonitor
}
sig SFProcess extends Process {} {
some src
```

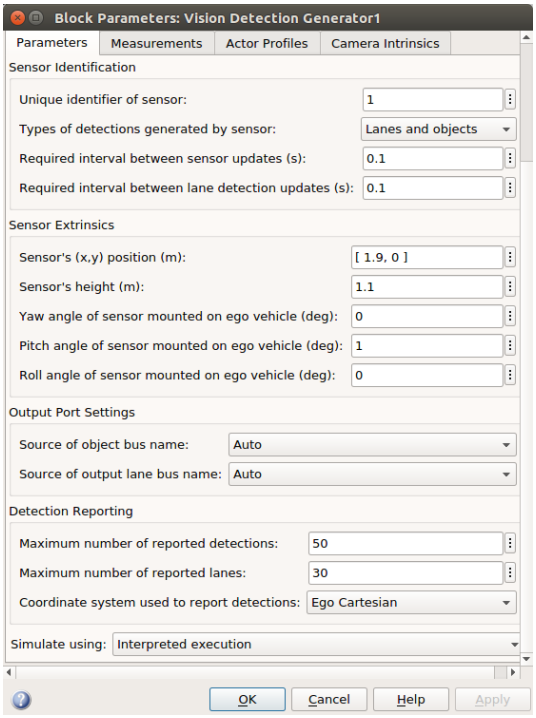
```
}

fact { all p:Process { one n: Node | n.process = p }}
fact { all m:Monitor | one n: Node | n.monitor = m }

some sig SensorNode extends Node {
}{
no depends
process in Sensor
monitor in SensorMonitor
}
some sig Sensor extends Process {}{
no src
}

pred SFHierarchy { }

run SFHierarchy for 10 but exactly 5 Node
```

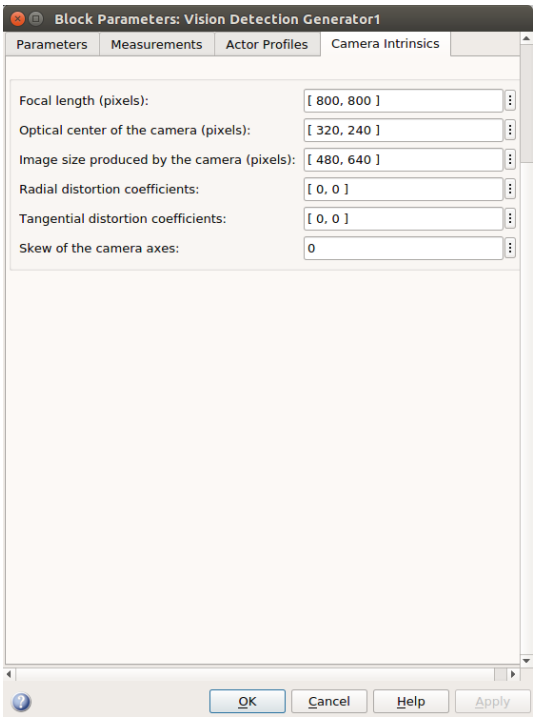
Appendix B

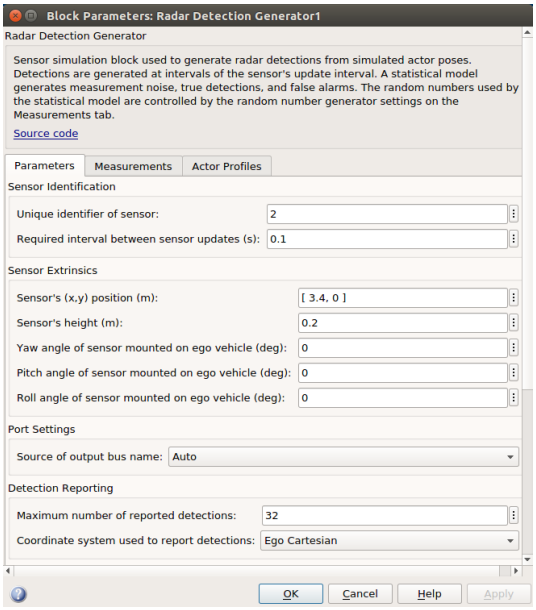
Simulink Model Parameters

B.1 Vision Detector Parameters

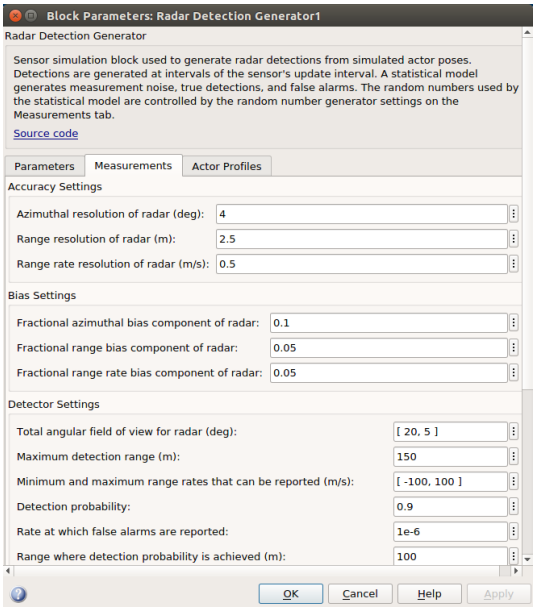



Block Parameters: Vision Detection Generator1
Parameters
Measurements
Actor Profiles
Camera Intrinsics
Sensor Identification
Unique identifier of sensor:
1
Types of detections generated by sensor:
Lanes and objects
Required interval between sensor updates (s):
0.1
Required interval between lane detection updates (s):
0.1
Sensor Extrinsics
Sensor's (x,y) position (m):
[1.9, 0]
Sensor's height (m):
1.1
Yaw angle of sensor mounted on ego vehicle (deg):
0
Pitch angle of sensor mounted on ego vehicle (deg):
1
Roll angle of sensor mounted on ego vehicle (deg):
0
Output Port Settings
Source of object bus name:
Auto
Source of output lane bus name:
Auto
Detection Reporting
Maximum number of reported detections:
50
Maximum number of reported lanes:
30
Coordinate system used to report detections:
Ego Cartesian
Simulate using:
Interpreted execution
OK
Cancel
Help
Apply


Block Parameters: Vision Detection Generator1
Parameters
Measurements
Actor Profiles
Camera Intrinsics
Focal length (pixels):
[800, 800]
Optical center of the camera (pixels):
[320, 240]
Image size produced by the camera (pixels):
[480, 640]
Radial distortion coefficients:
[0, 0]
Tangential distortion coefficients:
[0, 0]
Skew of the camera axes:
0
OK
Cancel
Help
Apply



B.2 Radar Detector Parameters



Appendix C

Extra Test Results

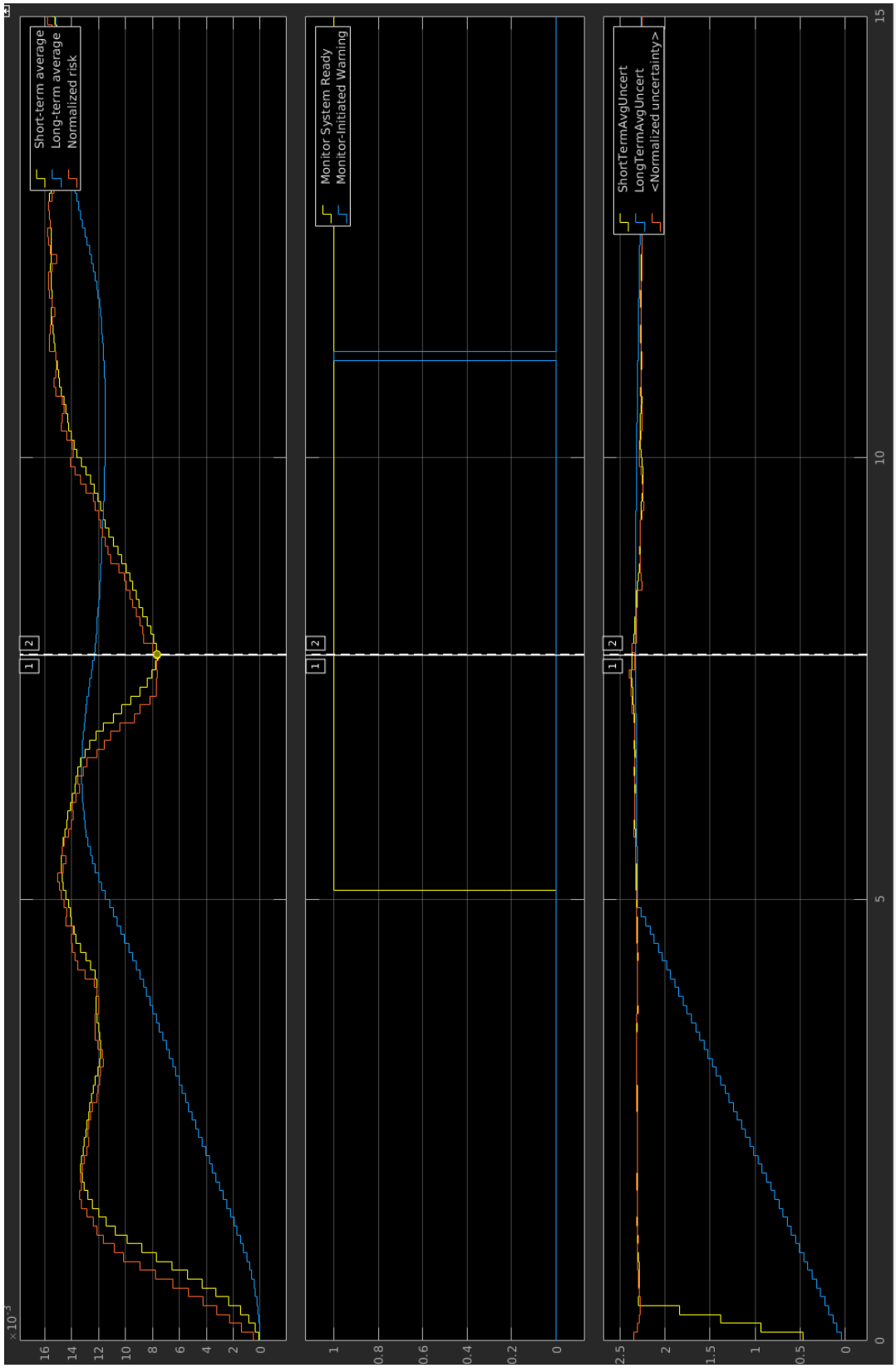


Figure C.1: Risk over time captured during testing of the AEB/FCW functionality. At around 5 seconds the total risk begins to drop quickly due to the first detections of the stopped vehicle by the main tracking algorithm. Interestingly, the accident occurs around 7.7 seconds in (at the location of the scope cursors), where the risk, interpreted as 'risk of false negatives' is at its lowest. Due to the low risk relative to the long term average, the early warning system, labelled 'monitor-initiated warning' in the plot, does not activate

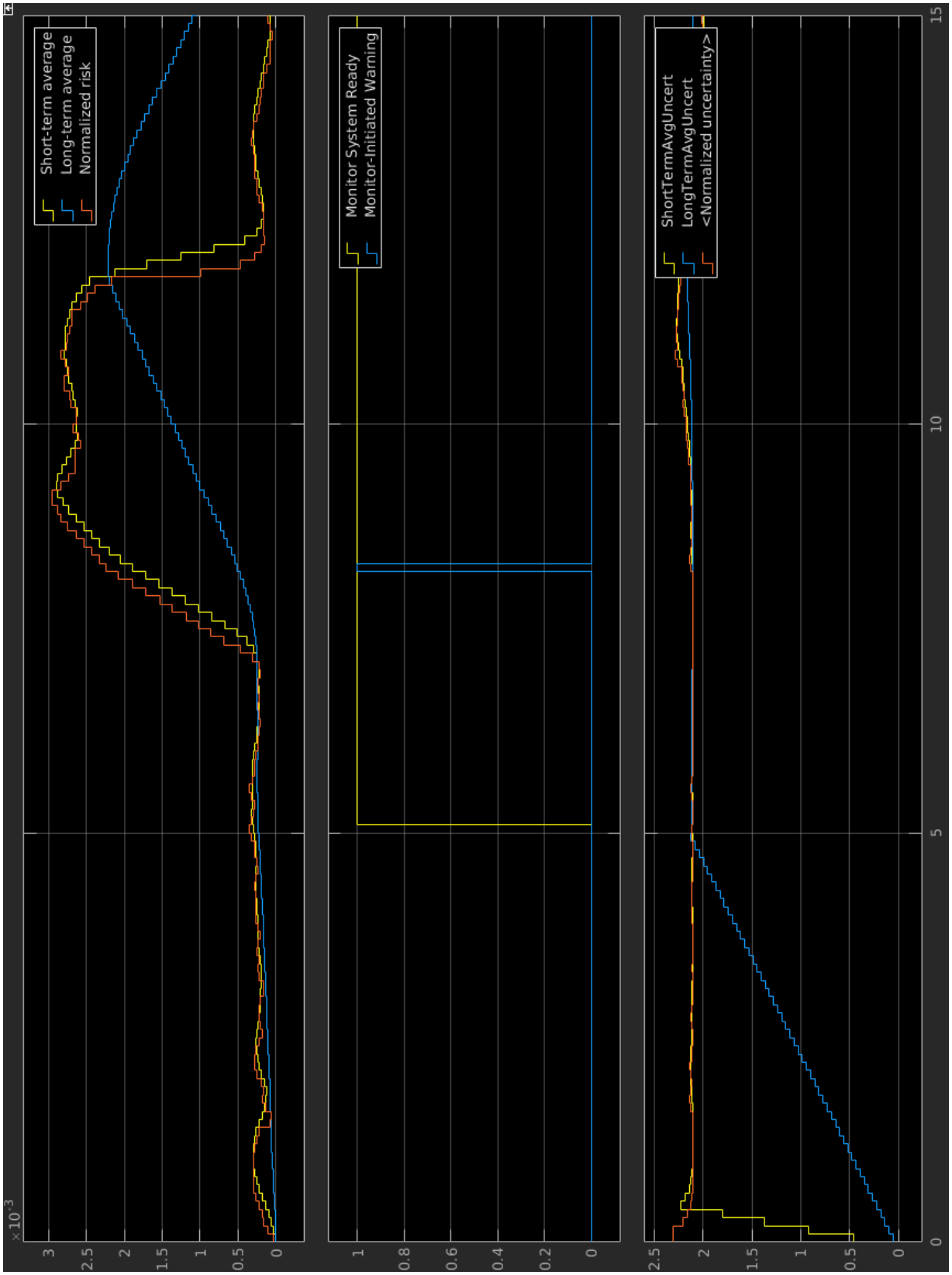


Figure C.2: The test results from running the simulated May 7th test case against the system with monitors. The top graph shows a large increase in short term system risk as the vehicle approaches the truck without it being detected by the main sensor fusion node. An early warning is issued, labelled as 'monitor-initiated warning' in the plot. The warning can be seen in the middle graph, triggered at around 8.5 seconds, with the actual collision taking place at 11.7s, shortly before the steep drop-off in detected risk. This drop-off is due to the fact that the vehicle has passed the obstacle in the simulation, and there are no longer any spurious sensor readings.

Bibliography

[1] P. L. Bogler. "Shafer-dempster reasoning with applications to multisensor target identification systems". In: *IEEE Transactions on Systems, Man, and Cybernetics* 17.6 (Nov. 1987), pp. 968–977. ISSN: 0018-9472. DOI: 10.1109/TSMC.1987.6499307.

[2] Subhash Challa and Don Koks. "Bayesian and Dempster-Shafer fusion". en. In: *Sadhana* 29.2 (Apr. 2004), pp. 145–174. ISSN: 0973-7677. DOI: 10.1007/BF02703729. URL: https://doi.org/10.1007/BF02703729 (visited on 06/17/2019).

[3] Erik Coelingh et al. "COLLISION WARNING WITH AUTO BRAKE - A REAL-LIFE SAFETY PERSPECTIVE". en. In: (2007), p. 10.

[4] B.V. Dasarathy. "Sensor fusion potential exploitation-innovative architectures and illustrative applications". In: *Proceedings of the IEEE* 85.1 (Jan. 1997), pp. 24–38. ISSN: 1558-2256. DOI: 10.1109/5.554206.

[5] D.L. Hall and J. Llinas. "An introduction to multisensor data fusion". In: *Proceedings of the IEEE* 85.1 (Jan. 1997), pp. 6–23. ISSN: 0018-9219, 1558-2256. DOI: 10.1109/5.554205.

[6] Brian Heater. *Mobileye says Tesla auto braking tech wasn't designed for scenario behind fatal crash.* en-US. Tech Blog. July 2016. URL:

https://social.techcrunch.com/2016/07/01/mobileye-tesla/ (visited on 04/13/2020).

[7] Sean Hollister. *Tesla and Mobileye disagree on why Autopilot didn't apply brakes*. en. URL: https://www.cnet.com/roadshow/news/tesla-autopilot-mobileye-statement-ltap/ (visited on 04/13/2020).

[8] *J3016B: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles - SAE International*. 2018. URL: https://www.sae.org/standards/content/j3016_201806/.

[9] Bahador Khaleghi et al. "Multisensor data fusion: A review of the state-of-the-art". en. In: *Information Fusion* 14.1 (Jan. 2013), pp. 28–44. ISSN: 1566-2535. DOI: 10.1016/j.inffus.2011.08.001. URL: http://www.sciencedirect.com/science/article/pii/S1566253511000558 (visited on 11/14/2019).

[10] E. Lefevre, O. Colot, and P. Vannoorenberghe. "Belief function combination and conflict management". en. In: *Information Fusion* 3.2 (June 2002), pp. 149–162. ISSN: 1566-2535. DOI: 10.1016/s1566-2535(02)00053-2.

[11] Herschel W. Leibowitz, D.Alfred Owens, and Richard A. Tyrrell. "The assured clear distance ahead rule: Implications for nighttime traffic safety and the law". en. In: *Accident Analysis & Prevention* 30.1 (Jan. 1998), pp. 93–99. ISSN: 00014575. DOI: 10.1016/S0001-4575(97)00067-5. URL: https://linkinghub.elsevier.com/retrieve/pii/S0001457597000675 (visited on 05/02/2019).

[12] Nancy Leveson and JOHN Thomas. "STPA handbook". In: *NANCY LEVESON AND JOHN THOMAS* 3 (2018).

[13] The Mathworks. *Autonomous Emergency Braking with Sensor Fusion*. Matlab official Documentation and Examples. May 2020. URL: https://www.mathworks.com/help/driving/examples/autonomous-emergency-braking-with-sensor-fusion.html.

[14] NHTSA. *NHTSA Investigation PE 16-007*. ODI Resume. NHTSA - National Highway Traffic Safety Administration, Jan. 2017. (Visited on 02/08/2020).

[15] NTSB. *Collision Between a Car Operating With Automated Vehicle Control Systems and a Tractor-Semitrailer Truck Near Williston, Florida, May 7, 2016*. en. Highway Accident Report NTSB/HAR-17/02 PB2017-102600. Sept. 2017, p. 63.

[16] NTSB. *Collision Between Vehicle Controlled by Developmental Automated Driving System and Pedestrian, Tempe, Arizona, March 18, 2018*. en. Highway Accident Report NTSB/HAR-19/03 PB2019-101402. Nov. 2019, p. 78.

[17] NTSB. *Rear-End Collision Between a Car Operating with Advanced Driver Assistance Systems and a Stationary Fire Truck, Culver City, California, January 22, 2018*. en. Highway Accident Brief HAB1907. 2018, p. 14.

[18] Glenn Shafer. *A Mathematical Theory of Evidence*. en. Google-Books-ID: wug9DwAAQBAJ. Princeton University Press, Apr. 1976. ISBN: 978-0-691-10042-5.

[19] Syed Asim Shah. "Safe-AV: A Fault Tolerant Safety Architecture for Autonomous Vehicles". en. Thesis. 2019. URL: https://macsphere.mcmaster.ca/handle/11375/24205.

[20] E-GAS Workgroup. *Standardized E-Gas Monitoring Concept for Gasoline and Diesel Engine Control Units. T.* Tech. rep. 6.0. July 2015.