UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Integrated Modelling of Functional Capabilities and Reliability Analysis of Outdoor Autonomous Vehicle Intelligence

Master's Dissertation

Brain N. Ndlovu
Supervisor: Dr. Michael Ayomoh

November 22, 2022

Abstract

The autonomous vehicles concept and development were founded in the 1980s, but they became more famous and advanced more than a decade ago. Autonomous vehicles were created due to the advancement of different technologies, and it was believed to portray the progress of the 21st century. This idea led people to think these autonomous vehicles might help reduce or mitigate road accidents. However, firstly, according to the *National Law Review*, early accidents were recorded, and some were deadly. Secondly, the African continent has been left behind concerning technological advancement; hence, it is currently not ready for so-called *smart cities*. Therefore, the problem this dissertation looked into is that there is an issue of *complexity* associated with autonomous vehicles (with independent levels 4 and 5). The study aimed to objectively understudy the reliability of the intelligent autonomous vehicle amidst inter- and intra-complexities associated with autonomous ground vehicle navigation requirements. Therefore, an appropriate methodology had to be selected to fulfil the aim. Thus, two research methodologies were considered for this dissertation, which is (1) *design science research* and (2) *systems thinking* methodologies.

Additionally, a unification of these two methods was established, and a framework was designed. An optimal physical structure was developed using the established framework and analysing autonomous vehicles' sensor fusions. Furthermore, the reliability analysis model was formulated. The use of systems and reliability engineering theories and applications were adopted to develop and model the optimal structure and reliability model. Finally, the reliability of the autonomous vehicles with respect to traffic rules was calculated. It was found that there is a 99.94% chance that the autonomous vehicle will fail at least one of the traffic rules in 20 minutes.

Contents

List of Figures

List of Tables

Acronyms

AI Artificial Intelligence

AV Autonomous Vehicle

DARPA Defence Advanced Research Projects Agency

FOSM First Order Second Moment

AFOSM Advanced First Order Second Moment

AFOSMC Advanced First Order Second Moment for Correlated variables

SORM Second-Order Reliability Methods

FOGSM First-Order Gaussian Second Moment Method

FOGAM First-Order Gaussian Approximation Method

4IR Fourth Industrial Revolution

GPS Global Positioning System

IMU Inertia Measurement Unit

FSM Finite State Machines

CNN Convolutional Neural Network

mAP Mean Average Precision

YOLO You Only Look Once

CAV Connected and Automated Vehicle

LIDAR Light Detection and Ranging

RADAR	Radio Detection and Ranging
GNSS	Global Navigation Satellite System
PPS	Pulses Per Second
PCD	Point Cloud Data
1D	One-Dimensional
2D	Two-Dimensional
3D	Three-Dimensional
MMW	Millimetre Wave
GHz	Giga-Hertz
KHz	Kilo-Hertz
ToF	Time of Flight
MOT	Ministry of Transport
SRR	Short-Range Radars
MRR	Medium-Range Radars
LRR	Long-Range Radars
DSR	Design Science Research
FOV	Field-of-View
SONAR	Sound Navigation Ranging
CDF	Cumulative Density Function

Chapter 1

Introduction

The Autonomous Vehicle (AV)'s concept and development were founded in the 1980s (Wikipedia contributors, 2022b). Still, it became more famous and advanced over a decade ago (Grigorescu et al., 2020). The advancement of the AV was due to the advances in deep learning and Artificial Intelligence (AI). The AVs were created due to the rise of different technologies. That idea led people to believe that AVs might help reduce or mitigate road accidents. Furthermore, it was believed that having AVs portrays the advancement of the 21^{st} century. However, early accidents were recorded according to the *National Law Review's* website (www.natlawreview.com). The first accident that led to death was in 2016, May 7^{th}. Furthermore, an Uber AV had about 37 crashes before having a collision that led to a pedestrian's death in 2018, November 20^{th}, and it was found that a human driver caused the error and the AV Uber could not react appropriately.

These results caused concerns to the public since there was already an *issue of reliability* towards the AVs. However, these fatal accidents do not rule out the need to produce reliable AVs and ultimately reduce road accidents or further portray the advancements of the 21^{st} century. According to Grigorescu et al. (2020) and Chy et al. (2022), AVs have been positively developing to improve how they predict situations and how they should react to complex conditions and unknown/unforeseen situations. The improvements were possible due to the method called *deep learning* or *smart deep learning* and the advancements of AI technology and its applications. Given these advancements, it can be noted that there is good progress in this field (AI environment). Therefore, there is a need to look at how these AVs are built from the *functional level*, *performance level*, and *physical design level* so that a critical analysis is conducted and consequently a reliability analysis is provided. Furthermore, the study focuses on the AV components with intelligence embedded in them. But there is little research regarding the reliability of AVs available. Therefore, this dissertation focused on providing more insight concerning the reliability of AVs related to traffic rules.

Regarding the analysis of AVs, it should be noted that there are six different levels of driving automation (Casado-Herráez, 2020; Singh and Saini, 2021), please see section 1.1. However, this dissertation focused on the last two levels, level 4 and level 5, since they both have the latest technology features, including the ones found in the lower levels (levels 0 to 3). The AV in levels 4 and 5 can be argued that they can produce reliable performance on the road. Therefore, an assessment on the AVs in levels 4 and 5 was conducted with respect to the traffic rules—this will was done to assess if the AVs in levels 4 and 5 do obey the traffic rules. The assumption is that if an AV obeys all the traffic rules, then it is reliable. The traffic rules used to assess the AVs are defined in Chapter 3.

The AVs assessed in this dissertation are from different automotive industry companies, such as Tesla, Waymo (by Google), Yandex, etc—i.e., the AV were not built in a laboratory for the purpose of this research.

It should be noted that some insight of how autonomy works was withdrawn from Defence Advanced Research Projects Agency (DARPA) grand challenge. The In the summer of 2002, DARPA officially announced the DARPA grand challenge (Behringer et al., 2004). The goal was to race fully autonomous road vehicles from Los Angeles to Las Vegas (at the United States of America) without controls or user intervention. In the first challenge, the distance to be covered was expected to be 400 km and completed in 10 hours. Unfortunately, in all of the AVs they tested, the maximum distance reached was 11.91 km (Behringer et al., 2004). The short distance was due to the short preparation time they had. What can be noted about DARPA grand challenge is that they already had the technology in the early 2000s. Therefore, the grand challenge provided solid background of what an AV is and what is expected to do.

1.1 Six different driving automation levels

The six different driving automation levels are grouped into two main categories, that is, *human driver monitors the road* and *automated driving system monitors the road* (see Figure 1).

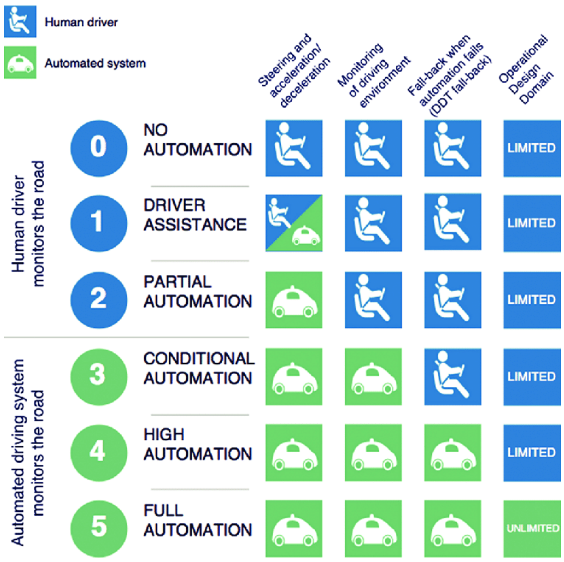


Figure 1: Six levels of driving automation (adopted from (Serban et al., 2020))

It can be seen in Figure 1 that these levels are based on the level of human involvement during the driving process. These levels are further described as follows (by Singh and Saini (2021)).

- *Level 0:* There is *no automation* of any sort, the driver performs all the tasks.
- *Level 1:* There are at least stand-alone vehicle components such as Automated Braking, here the *driver assists* on a lot of operations.
- *Level 2:* There is *partial automation* such that the vehicle is capable of steering and accelerating by itself to keep the vehicle accurately on the lane(s) and adaptively moving around other vehicles. However, the human driver should always be there to monitor the operation.
- *Level 3:* There is a *conditional automation* such that the human driver can take total control in certain complex situations, that is, the vehicle can drive itself in less complex situations until there is a need for human intervention.
- *Level 4:* There is *high automation* control in the vehicle, such that it can perform all needed driving functions by itself. Such vehicles might provide options for human intervention or might not provide it.
- *Level 5:* There is *full automation* such that the vehicle can perform all driving functionalities in any given situation (complex or easy) and condition.

Considering levels 4 and 5, it is crucial to look at how such systems (in terms of the functional capabilities that is linked to intelligence) are designed. With the focus on looking at how the levels 4 and 5 AVs are developed, the theory of *reliability engineering*, *systems engineering*, and *AI* were adopted to address the problem described in section 1.6. A further description of these theories is provided in sections 1.2, 1.3, and 1.4.

1.2 AV sensors

Sensors are tools that translate environmental events or changes into quantitative measurements that can then be processed further. Typically, sensors are divided into two categories based on their core principles of operation. Firstly, the *proprioceptive sensors*, also known as internal state sensors, record a dynamic system's state and internal values—these sensors relate to encoders, Inertia Measurement Unit (IMU), inertial sensors (gyroscopes and magnetometers), and positioning sensors (Global Navigation Satellite System (GNSS) receivers, such as Global Positioning System (GPS)). Secondly, the *exteroceptive sensors*, also known as *external state sensors*, they sense and gather data such as distance measurements or light intensity from the system's surroundings—these sensors relate to Cameras, ultrasonic sensors, Radio Detection and Ranging (RADAR), and Light Detection and Ranging (LIDAR). Furthermore, in consideration of AVs, sensors play a key role in the perception of the environment and localisation of the vehicles (the perception system or layer) for path planning and decision-making, all of which are necessary prerequisites for directing the motion of the vehicle. (Yeong et al., 2021)

The overview of where the sensors are installed in a vehicle to make it an AV can be seen in Figure 2.

It can be noted that Figure 2 gives brief explanations of each of the sensors shown. However, it is still necessary to give more insight into the sensors by providing more details to provide an understanding of what is meant by the individual sensors that describe the two kinds of sensors (the *proprioceptive sensors* and *exteroceptive sensors*). Therefore, further details on the sensors are provided as follows.

LIDAR

LIDAR is a distance sensing method based on the idea that laser or infrared light pulses are produced and reflect off the targets. The apparatus picks up these reflections, and distance estimation is possible between the light pulse's emission and receipt. Since its creation in the 1960s, LIDAR has been extensively used to map the topography for use in aviation and aerospace. Manufacturers of laser scanners produced and used

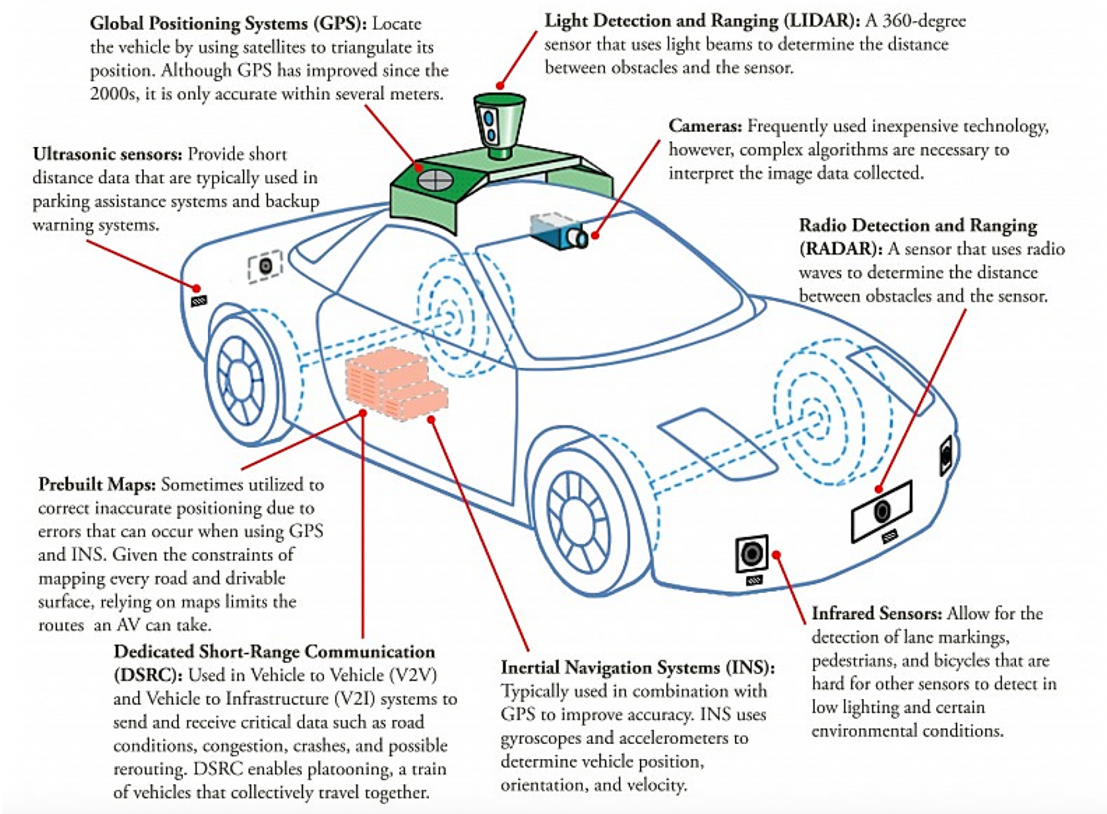


Figure 2: The AV sensor installation location overview (adapted from (Vargas et al., 2021))

the first commercial LIDARs with 2000 to 25,000 Pulses Per Second (PPS) for topographic mapping applications in the middle of the 1990s. In One-Dimensional (1D), Two-Dimensional (2D), and Three-Dimensional (3D) spaces, LIDAR sensors generate a point cloud data (also known as a Point Cloud Data (PCD)), which contain data on object intensity. (Ignatious et al., 2022; Yeong et al., 2021)

RADAR

In a wide range of military and civilian applications, such as aerial or terrestrial threat detection systems, shooting systems, airports, or meteorological systems, RADAR systems are utilised—which operate in wavelengths on the order of millimetres. This kind of technology usage in the automobile industry has been driven by the rise of smart vehicles and the need to improve road safety. The Millimetre Wave (MMW) RADAR, used in intelligent vehicle RADAR systems, operates at frequencies of 24/07/79 Giga-

Hertz (GHz). By computing the duration of flight of the sent signal and the returned echo, the RADAR determines the distance between the emitter and the object. The radars can correctly provide the direction and speed of the objects in addition to detecting the distance to multiple targets. (Rosique et al., 2019)

According to Ignatious et al. (2022), the *doppler effect*, often known as the *doppler shift*, illustrates how changes or modifications in wave frequency result from relative motion between a wave source and its targets. The frequency of the detected signal increases as the object moves in the direction of the RADAR system—which are the shorter waves. The general mathematical formula for the Doppler frequency shift of a RADAR may be expressed as follows.

$$f_D = \frac{2V_r f}{c} = \frac{2V_r}{\lambda}, \tag{1.1}$$

Where
f_D = Doppler frequency, in Hertz (Hz),
V_r = relative speed of the target,
f = frequency of the transmitted signal,
$c = 3 \times 10^8$ meters per second (m/s)—speed of light, and
λ = wavelength of the emitted energy.

Camera

Cameras are the main sensors for high-resolution tasks, including object classification, semantic picture segmentation, scene perception, and activities requiring color perception, like a traffic light or sign identification. The fundamental idea behind how cameras function is that they use a lens to detect light from things placed on photosensitive surfaces. The photosensitive surfaces measure the quantity of light they receive and transform that into electron motions. The cameras can locate any object. An object's location may be described as a vector in space with three components (x, y, z), which will be projected via a lens to a point on the picture plane (x, y). The projected points are transformed from the metric unit to pixels to use the image for additional processing and information extraction, such as item recognition, categorisation, and tracks in our autonomous vehicle's route. (Campbell et al., 2018; Shahian Jahromi et al., 2019; Yeong et al., 2021)

Ultrasonic sensor

Ultrasonic sensors determine the distance to an object by using sonic waves in the 20 Kilo-Hertz (KHz) to 40 KHz frequency range produced by a magnetoresistive membrane. It works by measuring the sonic wave's Time of Flight (ToF) from the moment it is released until the echoes are picked up. The distance is calculated as follows. (Rosique et al., 2019)

$$d = \frac{c}{2}(ToF), \tag{1.2}$$

where
$c =$ velocity of the wave, in m/s.

Furthermore, according to Vargas et al. (2021), ultrasonic sensors can find solid, liquid, granular, or powdered items. Finally, they argued that ultrasonic sensors rely on sonic transducers to emit sonic waves in the 40 to 70 KHz range for automotive applications. This is a significant difference from Rosique et al. (2019), and a logical reason would be that they probably commented on different types of ultrasonic sensors with different properties.

GNSS

The most popular technique for obtaining precise location data on the earth's surface is the GNSS. The GPS system, which offers positioning, navigation, and timing services to users, is the most well-known GNSS system. The GPS was created by the U.S. Department of Defense in early 1970 and consists of three elements—the space segment, the control segment, and the user segment. Furthermore, a minimum of 24 of the 31 active satellites in the space segment are available 95% of the time. Each of these satellites circles the planet twice daily in medium earth orbit at a height of 20,200 kilometres (km). This makes the AVs to require this kind of service as it is free and very efficient for positioning, navigation, and timing at any given time. (Vargas et al., 2021)

1.3 Reliability and analysis

Reliability is when a product (or an item or a system or a service) is capable of performing a given task following specified conditions in a given set period, and it can be expressed as *the number of failures over a certain period* (Misra, 2008; O'Connor and Kleyner, 2012). Consequently, a reliable product (or system or service) can perform its given tasks with certain conditions and in a specified period *without failure*. Given the reliability definition, it follows that *reliability analysis* is when a product (or an item or a system or a service) is tested using the terms and conditions outlined in the reliability definition. The test is conducted by calculating several commonly used measures of scale in reliability (see Chapter 3.1).

According to Denoël (2007), there are six different reliability analysis methods. These methods are described as follows.

First Order Second Moment (FOSM)

This method is rigorous in cases when the failure function is linear only (see Figure 3). The method focuses on replacing the actual failure function with a linearised relation, which means that the failure function remains unchanged when any transformation is applied (also known as *lack of invariance*). The method depends on the linearity of the failure conditions.

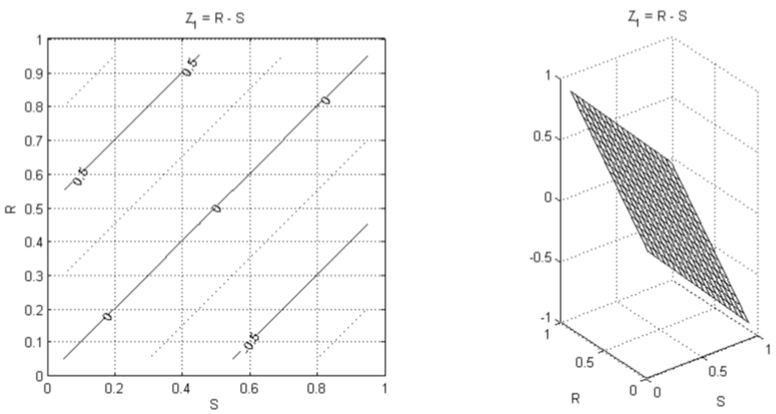


Figure 3: FOSM linear failure function example (adapted from (Denoël, 2007))

Please note that R and S (in Figure 3) are resistance and load, respectively, which are used to assess a failure condition of a product or system such that $S > R$ when the product or system is reliable.

Advanced First Order Second Moment (AFOSM)

This method was developed due to the lack of invariance in the FOSM method. The development made in this method is that it considers the beginning of the linearity of the failure function and does not depend on the linearity of the failure condition. This can be seen in Figure 4 below.

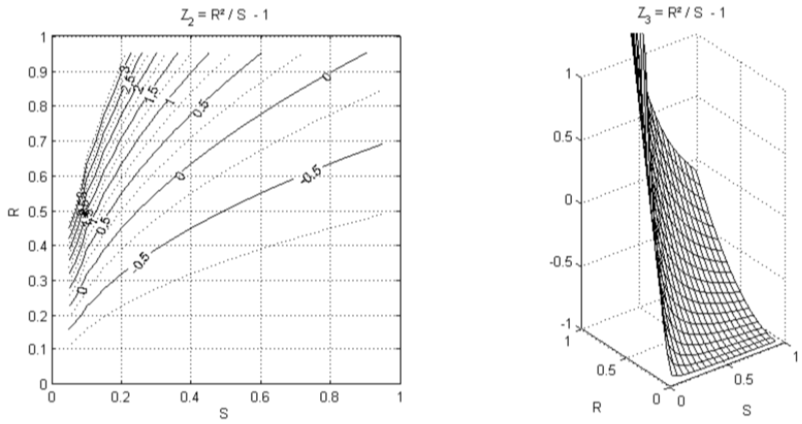


Figure 4: AFOSM linear failure function example (adapted from (Denoël, 2007))

Advanced First Order Second Moment for Correlated variables (AFOSMC)

This method is a modification of the AFOSM method, such that the involved variables are correlated; that is, the variables involved should have a mutual relationship or connection, and they affect each other.

Second-Order Reliability Methods (SORM)

This reliability method focuses on analysing products or systems with random variables interacting in a non-linear fashion.

First-Order Gaussian Second Moment Method (FOGSM)

This method deals with products or systems with Gaussian random variables. The Gaussian random variable is known to follow a normal distribution, and its shorthand definition can be written as

$$X - N(\mu, \sigma^2).$$

First-Order Gaussian Approximation Method (FOGAM)

This method was developed to improve the results on the FOGSM method by replacing the actual probability density functions with the equivalent Gaussian probability distribution.

Given the above outlined six different reliability analysis methods, the AFOSMC was adopted for this study. This method deals with variables with some relationship that will affect each other's performance or response. For this to be possible, another method is required, which is the *needs and requirement analysis* method, which was integrated with AFOSMC method (see Section 1.4).

1.4 Needs and requirement analysis

To provide a reliability analysis of the intelligence of AVs concerning traffic rules, the functional capabilities that are linked to the intelligence of the AV should be outlined first. Therefore, the *Systems Engineering* theory and method were utilised to identify the functional capabilities that speak to the intelligence of the AV.

Kossiakoff et al. (2011) describes how the systems engineering theory and method can be applied in *needs and requirements analysis*. *Needs and requirements analysis* is the first phase of the *origin of a new system* that is either driven by technological opportunity or unique needs. This study focuses on the AV that is caused by technological opportunities, that is, AI (this is further explained in the next section, Section 1.5). The *needs and requirements analysis* has two inputs, the operational deficiencies and technological opportunities, so that a system can be studied, its technology can be assessed, and its operation can be assessed. Finally, there are two outputs, that is, the *system operational effectiveness* and *system capabilities* (see Figure 5). These outputs were used to develop further a reliability model that calculates the reliabilities of individual components that speak to intelligence to provide the overall reliability of the level 4 to 5 automation vehicle.

The *needs and requirements analysis* have four activities that should be considered during its execution (Kossiakoff et al., 2011). These activities are briefly discussed as follows.

- *Operations Analysis.* This activity is also known as a *requirement analysis.* This activity results are knowing the *operational objectives* and identifying the *system capabilities.*

- *Functional Analysis.* This activity is also known as a *functional definition*; here, the *operational objective* are translated into functions and allocated into subsystems. The results of this activity are the list of *functional requirements* of the system/subsystem/components.

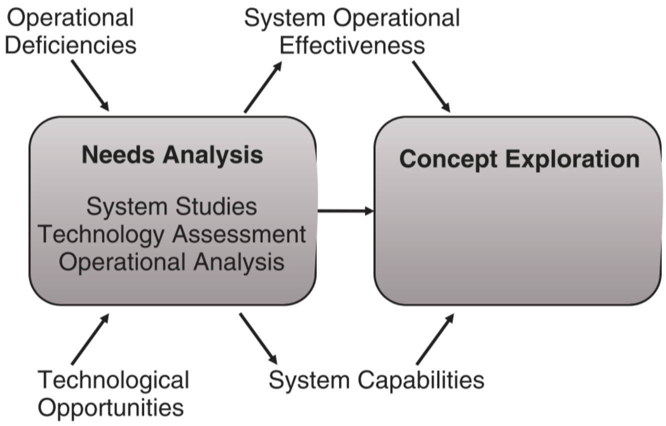


Figure 5: The needs and requirements analysis phase (adopted from (Kossiakoff et al., 2011))

- *Feasibility Definition*. This activity is also known as a *physical definition*; here, the physical nature of the subsystems is visualised to check if they can perform the required functions. Furthermore, a feasibility concept should be defined considering the costs and capabilities of the system/subsystem/component. The result of this activity is a list of initial physical *requirements*.
- *Needs Validation*. This activity is also known as a *design validation*. It is when a model or a validation criterion is designed or adapted to check the validity of the suggested solution(s).

These activities have been used to identify the operational objectives and system capabilities of the AVs. In addition, it is done to determine the functional and physical abilities of AVs to compute the reliability of an AV with driving automation between levels 4 and 5.

1.5 Problem context

The problem context of this study focused on the automotive industry, more specifically on the vehicles with AVs with levels 4 and 5 of autonomy (see section 1.1 for the descriptions of the levels). This focus relates to the reliability analysis of AVs. To accomplish the objectives stated in section 1.7, the theory of AI technology, *Systems Engineering*, and *Reliability Engineering* were incorporated into the study. Simulation modelling could have added value in evaluating the reliability of AVs. However, the data to conduct simulation modelling was unavailable; therefore, simulation was not utilised.

1.6 Problem statement

The problem is that there is an issue of *complexity* associated with autonomous vehicles (autonomous in levels 4 and 5). The associated complexities are a diversity of navigation requirements on the road, connectivity (inter- and intra-connectivity) of the intelligent system, intelligent interaction (inter- and intra-interaction), swift decision-making, and obedience to road rules. It should be noted that these complexities are viewed from the *Systems Engineering* point of view.

1.7 Research aim and objectives

This dissertation aims to objectively study the reliability of the intelligent autonomous vehicle amidst inter- and intra-complexities associated with autonomous ground vehicle navigation requirements. Given the research aim, three objectives were derived and outlined as follows.

Objective 1: Model and analyse the reliability of the intelligence of autonomous vehicles with respect to traffic rules.

Objective 2: Model and analyse the inherent complexity associated with autonomous vehicles.

Objective 3: Validate the reliability (analysis with respect to the traffic rules) model using a reliability engineering technique.

1.8 Research questions

Three research questions were formulated, which were derived from the research objectives outlined in section 1.7. The search questions are now described.

Research question 1: *What is the state-of-the-art of autonomous vehicles?*

Research question 2: *What is the reliability of the intelligence of autonomous vehicles with respect to traffic rules?*

Research question 3: *What is the inherent complexity associated with autonomous vehicles?*

1.9 Scope limitation, demarcation and assumptions

This study only focused on the automotive industry's driving automation of levels 4 and 5; from the outlined driving automation levels in section 1.1, levels 0 to 3 are excluded. The theories and methods adopted for this study are systems engineering methods in the needs analysis phase of the system life cycle and reliability engineering.

1.10 Significance and justification

The industrial revolution is essential, and it helped with the development of cities/provinces, countries, and continents. However, during the development of the industrial revolution (the Fourth Industrial Revolution (4IR)), Africa was left behind (Ndung'u and Signe, 2020). Since there is a new developing era of technology whereby cities need to be more intelligent, that is, the *era of globalisation* with smart cities (Komninos, 2008; Mishra, 2020), Africa is almost two industrial revolutions behind. It should be noted that industrial revolution is still the currently revolution around the world, however, the *era of globalisation* is emerging. Smart cities can be defined as a network of interconnected technologies that communicate with each other and transfer and analyse essential data so that urban operations are improved and maintained (Mishra, 2020). Moreover, smart cities include smart transportation, such as autonomous vehicles. Hence, this study is significant as it provides help to automotive industries in Africa that intend to adapt or start to (re)design autonomous vehicles.

1.11 Table of definitions

The definitions in Table 1 are not general definitions, they relate to how the terms have been utilised in this dissertation.

Table 1: The table of definitions.

Term	**Definition**
Inter-connectivity	Refers to the connectivity of components within an AV—the components that makes the AV intelligent.
Reliability	Refers to ability of a system to accomplish a task required under the conditions provided for a certain time interval
Reliability analysis	Refers to investigating the characteristics of measuring scales and the components that make up the scales.
Inherent complexity	The inherent complexity of an AV refers to what makes it difficult for the vehicle to understand, especially in its design, to comply with certain conditions in a certain given period. Furthermore, the inherent complexity of an autonomous vehicle also refers to how difficult it is for an autonomous vehicle to be deployed in a certain country, city, or province.
Intelligence	Refers to the capability for knowledge and skill acquired to be applied by a component.
Intelligent inter-interaction	Refers to interactions of intelligent components within an AV.
Intelligent intra-interaction	Refers to interactions of the AV—vehicle to vehicle communication.
Intelligent system	Refers to machines with cutting-edge technology that can perceive their surroundings and respond accordingly.
Swift decision-making	The intelligent system has to make a decision on the given conditions without unnecessary delay.

1.12 Document structure

The rest of the chapters are now discussed. First, Chapter 2 answers the research questions (research questions 1 and 3) outlined in section 1.8. Next, Chapter 3 defines the research methodology used in this study, provides an in-depth analysis of the problem described in section 1.6 and the requirements to provide a solution, and provides the solution development and design. Next, the results obtained after assessments and evaluations are provided in Chapter 4. Finally, this dissertation's conclusion and future work are provided in Chapter 5.

Chapter 2

Literature Review

The purpose of this chapter is to provide comprehensive answers to the research questions outlined in section 1.8. Firstly, section 2.1 answers the question about state-of-the-art autonomous vehicles (i.e., research question 1). Secondly, section 2.2 answers the question about the inherent complexity associated with autonomous vehicles (i.e., research question 3).

2.1 Autonomous vehicle state-of-the-art

The state-of-the-art refers to the most recent or highest developments due to a certain level of development according to Wikipedia contributors (2022c). Therefore, state-of-the-art of autonomous vehicles is the up-to-date developments an autonomous vehicle has reached. However, the state-of-the-art discussed in this section focuses on the *functional requirements*, *physicality (the design)*, and *performance* of the autonomous vehicle.

Literature research was conducted to address state-of-the-art of autonomous vehicles, and the results are now discussed.

2.1.1 Autonomous vehicle functional requirements

The functional requirements refer to what a system should do based on described tasks or activities it should perform during its operation (Kossiakoff et al., 2011). Consequently, Matthaei and Maurer (2015) conducted a study to present a functional system architecture for an autonomous vehicle. The study was developed in a top-down approach based on the functional requirements of autonomous vehicles, and these requirements are described as follows.

- *Operating:* The vehicle needs instructions (these refer to the mission of the vehicle), and usually, human beings write out these instructions.
- *Mission accomplishment:* Now that the mission has been described, the vehicle should be able to accomplish the mission or the instructions – these include behaviour, navigation, and the control of the actuators.
- *Map data:* This data is required for route planning.
- *Localisation:* The vehicle should know its location or position on a global scale for map data, such as navigation, and the purpose of communication in vehicle-to-vehicle or vehicle-to-infrastructure communication.
- *Environmental perception:* The vehicle should know its environment, whether it is stationary or moving, and it is expected to know the dynamics of the moveable elements.
- *Cooperation:* The vehicle is expected to respond as required in such a way that it reacts accordingly based on other traffic participants. The vehicle should also communicate its intentions to those other traffic participants.
- *Safety:* The vehicle is expected to cause no harm or danger to its environment.
- *Self-perception:* The vehicle is expected to know its state at all times, it should know its state in terms of its motion, functional capabilities, etc.

Furthermore, Vaicenavicius et al. (2020) conducted a study focused on a rigorous modular statistical approach for reasoning with the safety of an autonomous vehicle. The study aimed to explain the component level of the importance of the vehicle's safety. An iterative approach was adopted to accomplish their aim, as seen in Figure 6. In their problem development, a system specification was provided, highlighting one important functional requirement: the *ability to stop to avoid harm or danger*. This requirement sums up a couple of requirements. Matthaei and Maurer (2015) outlined (cooperation, environmental perception, safety, and operating – this requirement is implied since the vehicle may need to be halted if it was operational).

Sviatov et al. (2021) described a structural and functional model of an autonomous vehicle control system intending to generate several mathematical problems. Figure 7 shows the developed model, and this model is crucial as it provides the link between the structure and functions of the vehicle. In addition, the study identified the *control*

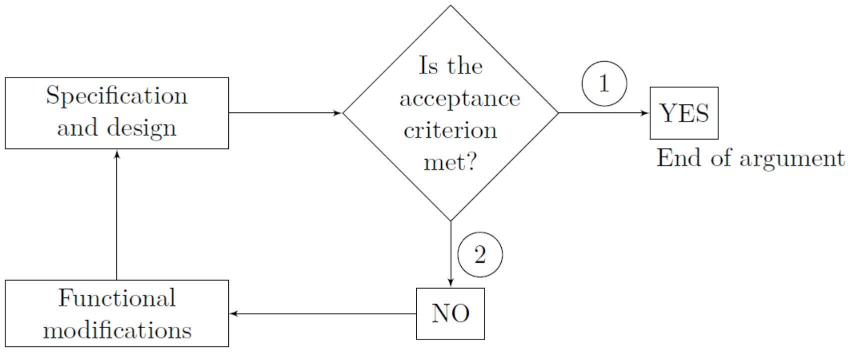


Figure 6: The iterative process approach (adopted from (Vaicenavicius et al., 2020)).

system of an autonomous vehicle as a functional requirement. This requirement refers to the vehicle's ability to control itself – hence Autonomous Vehicle (AV).

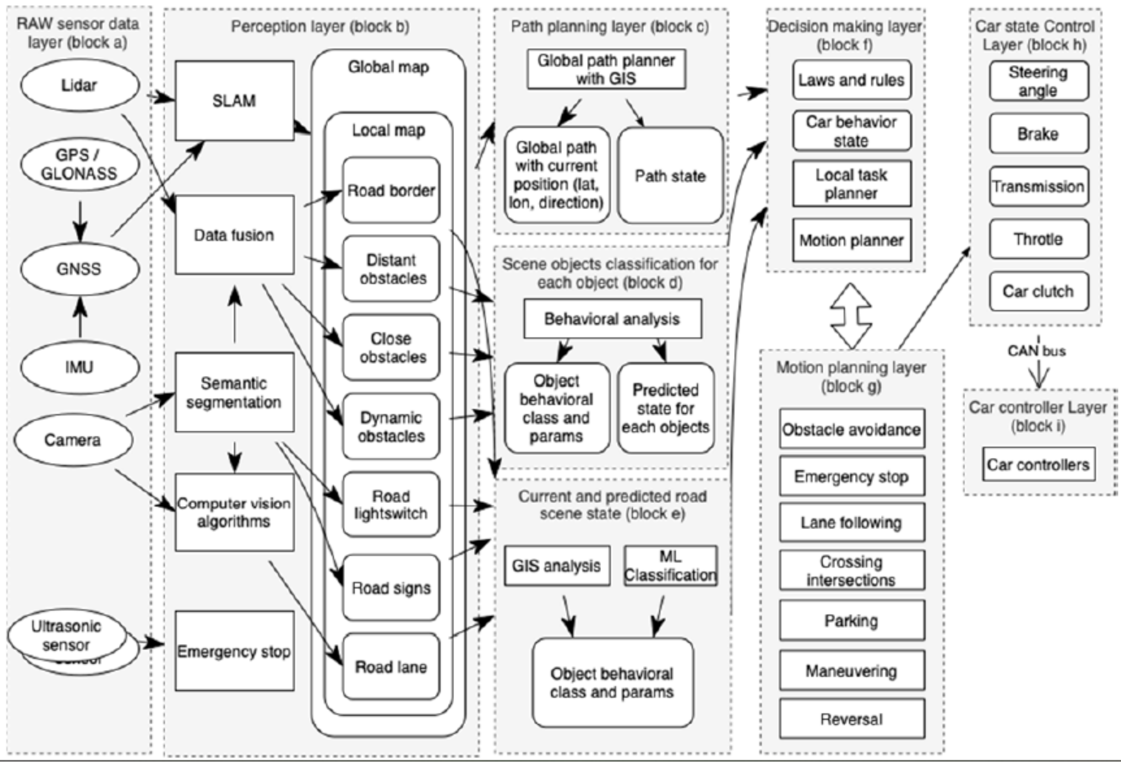


Figure 7: The structural and functional control system of an autonomous vehicle (adopted from (Sviatov et al., 2021)).

It can be noted that this control system covers six of the functional requirements outlined by Matthaei and Maurer (2015), leaving out the *operating* and *mission accom-*

plishment functional requirements. However, it can be deduced that these requirements are seen as primary, and without them, the control system will not accomplish its purpose – to control (conduct all necessary tasks) the vehicle from point A to point B.

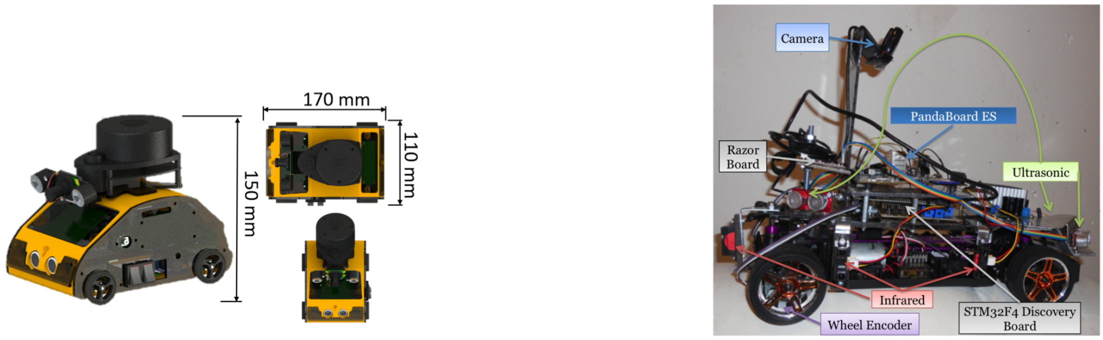
Badue et al. (2021) surveyed search on AVs that focused on vehicles with the autonomous driving capability of level 3 and above (section 1.1 describes the levels). The study identified two main functional requirement categories, that is, the *perception system* and the *decision-making system* of AV. The perception system of AV consists of the following functional requirements, (1) the vehicle should have different methods of localisation – these methods are *Light Detection and Ranging (LIDAR)-based localisation*, *LIDAR plus camera-based localization*, and *camera-based localization*; (2) the vehicle should be able to map obstacles offline – these can be accomplished by *regular spacing metric representation* and *varied spacing metric representation*; (3) should be able to conduct road mapping – ways to accomplish this are *metric representation* and *topological representation*; (4) should be able to track moving obstacles – the following can be done to track moving obstacles, *traditional-based Ministry of Transport (MOT)*, *model based MOT*, *stereo vision based MOT*, *grid map based MOT*, *sensor fusion based MOT*, and *deep learning based MOT*; and (5) should be able to detect and recognise traffic signalisation – these can be achieved by *traffic light detection and recognition*, *traffic sign detection and recognition*, and *pavement marking detection and recognition*.

Regarding the *decision making system* of an AV, Badue et al. (2021) described the following functional requirements, the vehicle should be able to (1) conduct a route planning – the vehicle can achieve this by using the following techniques *goal-directed*, *separator-based*, *hierarchical*, and *bounded-hop*, or combining any of the techniques; (2) select its expected behaviour – the techniques that can be adopted for this requirement are *Finite State Machines (FSM)-based* (Jo et al., 2015), *ontology-based* (Zhao et al., 2017, 2015), and *Markov decision process based* ; (3) plan its motion – the motion planning consists of *graph search based*, *sampling-based*, *interpolating curve based* such as clothoid curves (González et al., 2015), and *numerical optimisation based* techniques that can be adopted; (4) control its systems – the methods used for this are *direct hardware actuation control* and *path tracking*.

2.1.2 Autonomous vehicle design

This section focuses on the designs of autonomous vehicles – the architecture design that may contain system parts or components and/or subsystems' parts or components. This helps link the functional requirements and the physical design of autonomous vehicles in their design stage.

Berger (2014a) conducted a study focusing on an effective way to test AVs with minimal cost. It proved the need to use a *miniature* AV (miniature AV examples can be seen in Figure 8) instead of using simulation-based evaluation. In doing this, the study required a design of a AV to accomplish their objective. The hardware design of the miniature vehicle can be seen in Figure 9. Consequently, other studies were conducted utilising the miniature AVs (Deac et al., 2018; Kannapiran and Berman, 2020). The study by Kannapiran and Berman (2020) further demarcated a testing area to give perspective on how the testing area should look, as seen in Figure 10 – the area simulates the real-life kind of environment for vehicles to manoeuvre. It further provided the exploded vehicle with labelled parts (Figure11). The parts that have been provided in Figure 11 are similar to the parts provided in Figure 7 – which is a real-life tested control structure, and this proves that using miniature AVs to test how actual AVs might perform in real-life can be useful. Furthermore, the use of miniature AVs has proven useful in testing the development and improvement of self-driving (Al Mamun et al., 2014; Berger, 2014b; Zug et al., 2014).



(a) Miniature AV example 1 (adopted from (Kannapiran and Berman, 2020)).

(b) Miniature AV example 2 (adopted from (Berger, 2014a)).

Figure 8: The miniature AV examples

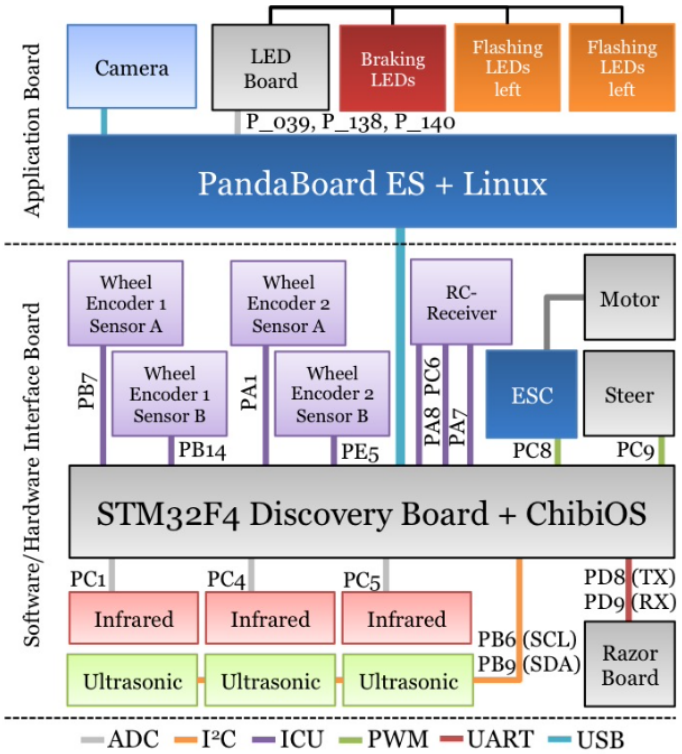
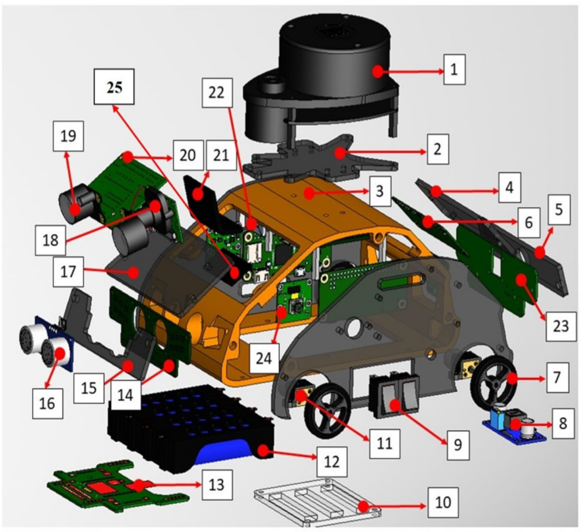


Figure 9: The hardware architectural design for the miniature vehicle (adopted from (Berger, 2014a)).



Serial No.	Component	Serial No.	Component
1	2D LiDAR	14	Front LED PCB
2	LiDAR and camera holder	15	Front light glass
3	Go-CHART main body	16	Ultrasonic sensor
4	Rear window glass	17	Front window glass
5	Rear light glass	18	Ultra-wide view camera (night vision)
6	RPi zero for rear and side-view cameras	19	UV blaster
7	Wheels	20	Power board
8	Buck converter	21	Camera holder
9	Rocker switches	22	RPi3B
10	Battery holder plate	23	Rear LED PCB
11	51.45:1 mini motor with encoder	24	RPi camera- rear and side
12	18650 battery	25	Teensy 3.2
13	Motor driver PCB		

Figure 11: The exploded miniature self-driving miniature vehicle that shows the important parts that make up the vehicle and its labels are provided on the right (adapted from (Kannapiran and Berman, 2020)).

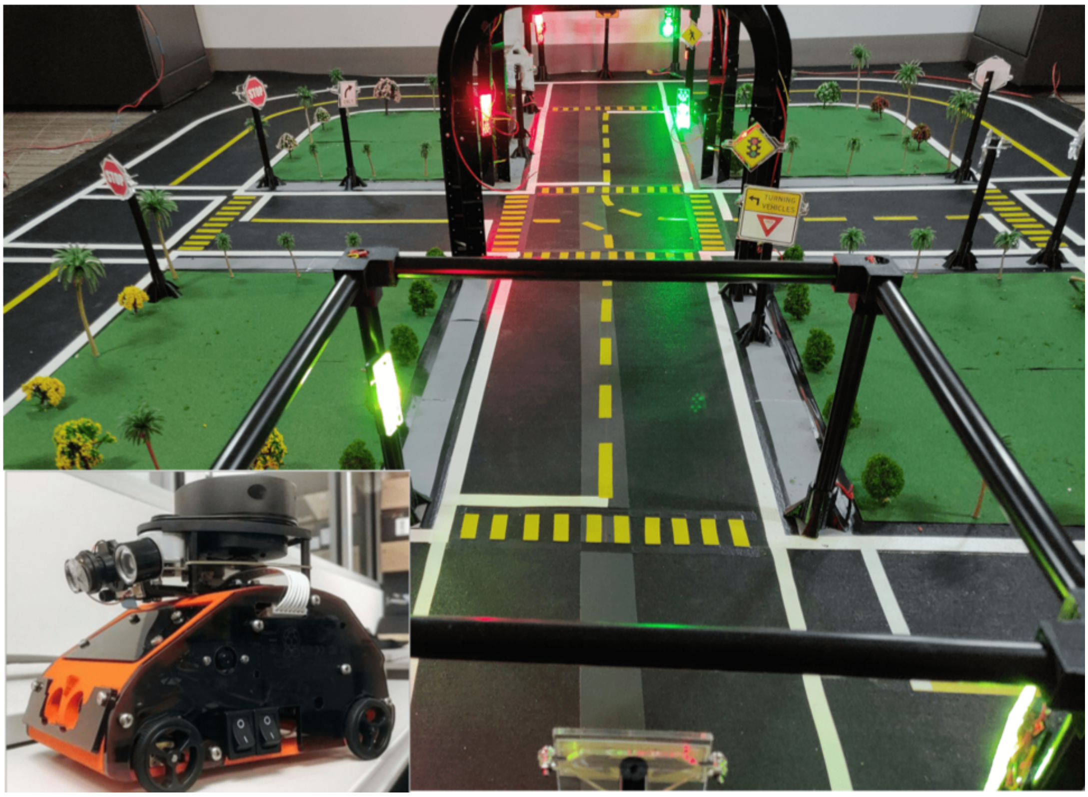


Figure 10: The demarcated miniature driving area for testing the self-driving miniature vehicles (adopted from (Kannapiran and Berman, 2020)).

It should be noted that the important parts in the design are the intelligent parts as they are also used the same way in AVs that carry human beings as passengers.

Guanetti et al. (2018) and Sell et al. (2018) introduced a control system highlighting the intelligent parts for the AVs. The control system designed by Guanetti et al. (2018) was also designed to be a planning architecture for an autonomous vehicle that led to optimised-based vehicle control. Furthermore, Sell et al. (2018) designed the control system for educational purposes. The provided architecture design can be seen in Figure 35 in Appendix A. Badue et al. (2021) provided a typical architecture of an autonomous system while conducting a survey on AVs, please see Figure 36 in Appendix A.

Autonomous vehicles need both hardware and software architecture, and they need to be in-sync so closely to avoid any miscommunication. Most architectural designs integrate both the software and hardware, but Sharma et al. (2021) provided a software architecture design for autonomous vehicles (Figure 12). However, it can be seen that there are some hardware parts labelled to support the design. This design provided important insight for this dissertation to assist in modelling a hardware structure that is directly and intentionally linked to a software structure of an autonomous vehicle during the planning and designing of the vehicle.

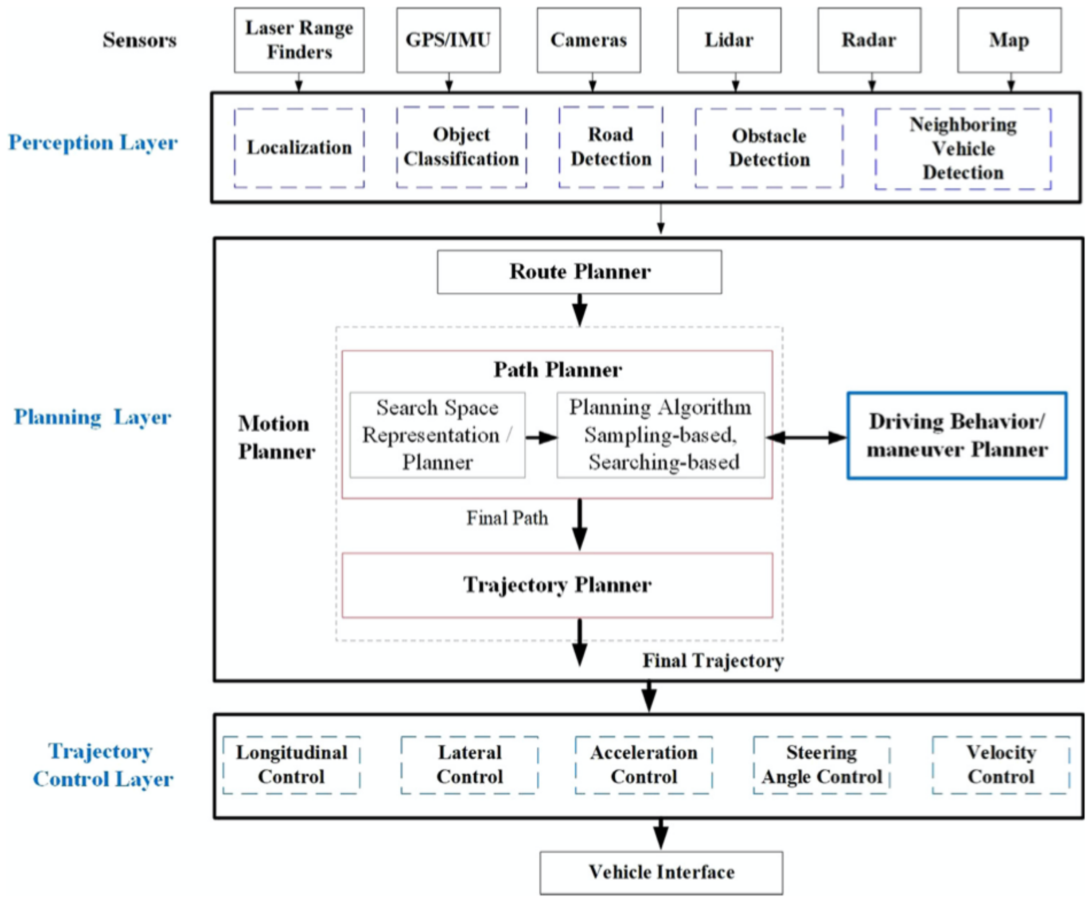


Figure 12: The architectural software design for an autonomous vehicle (adopted from (Sharma et al., 2021)).

2.1.3 Autonomous vehicle performance

A number of performances can be tested or reviewed since they can all be seen as important due to the expectations that come with autonomous vehicles. This section reviews performances that communicate the intelligence of the vehicles. The performances are now discussed.

Object detection

According to Shelhamer et al. (2016) and Wu (2017), the Convolutional Neural Network (CNN)-based approach is a promising real-time object detection method—CNN can be defined as a multilayer neural network which can also be referred to as deep learning architecture. It uses Artificial Intelligence (AI), inspired by the visual system of living beings and commonly used to analyse images (Chandana, 2020; Ghosh et al., 2020).

The use of CNN-based approach was reviewed by Hnewa and Radha (2020) (though other approaches were included in the study). It was found that the CNN-based approach performed well in clear weather conditions – that is, the object detection performed as expected. However, rainy weather conditions yield less accurate detection of objects than clear weather conditions; this does not mean the method does not work in rainy conditions or is poor. The problem that caused less accuracy in rainy conditions is the inability to detect and locate the objects as expected at some point, which is caused by the rain covering or obscuring the important details on the objects (Hnewa and Radha, 2020). The main performance metric used in the test was the Mean Average Precision (mAP)—which is said to be the most popular performance measure since 2012. The results obtained can be seen in the table provided in Figure 13. It can be seen that there was a noticeable change for mAP using the Faster R-CNN (also known as CNN) and You Only Look Once (YOLO) methods—this means the accuracy to detect the objects in rainy conditions decreases in both methods (YOLO and CNN).

	Faster R-CNN					YOLO-V3				
Mitigating Technique	**V-AP**	**P-AP**	**TL-AP**	**TS-AP**	**mAP**	**V-AP**	**P-AP**	**TL-AP**	**TS-AP**	**mAP**
None (clear conditions*)	72.61	40.99	26.07	38.12	44.45	76.57	37.12	46.22	50.56	52.62
None (rainy conditions**)	67.84	32.58	20.52	35.04	39	74.15	32.07	41.07	50.27	49.39
Deraining: DDN [14]	67	28.55	20.02	35.55	37.78	73.07	29.89	40.05	48.74	47.94
Deraining: DeRaindrop [15]	64.37	29.27	18.32	33.33	36.32	70.77	30.16	37.7	48.03	46.66
Deraining: PReNet [16]	63.69	24.39	17.4	31.68	34.29	70.83	27.36	35.49	43.78	44.36
Image translation: UNIT [32]	**68.47**	**32.76**	18.85	**36.2**	**39.07**	74.14	**34.19**	**41.18**	48.41	**49.48**
Domain adaptation [33]	67.36	**34.89**	19.24	**35.49**	**39.24**	Not applicable				

Figure 13: The object detection results obtained using the CNN and the YOLO methods for different mitigating techniques (more especially the mAP performance measure) to show the object detection differences between the clear and rainy conditions (adopted from (Hnewa and Radha, 2020)).

Some mitigating techniques, as in Figure 13, are defined as follows. V-AP stands for *vehicle average performance*, P-AP stands for *pedestrian average performance*, TL-AP stands for *traffic light average performance*, and TS-AP stands for *traffic sign average performance*.

Decision making

Autonomous vehicles can plan how to behave or react normally and sometimes in complex situations. However, complex situations are usually more difficult to handle, and hence mistakes occur. Complex situations in decision-making come with a dynamic environment. The dynamic environment brings about uncertainty in data acquisition handling – data acquisition helps in understanding an environment in real-time to plan appropriately to avoid dangerous situations. Data acquisition and analysis in real-time is still a challenge. (González et al., 2015)

The study conducted by Guanetti et al. (2018) (discussed in section 2.1.2) highlighted that decision-making and motion planning of Connected and Automated Vehicle (CAV) generated a reference trajectory for longitudinal and lateral motion. As a result, the trajectory is expected to follow traffic rules, be feasible for lower-level controllers, be comfortable for the passengers, and be capable of accurately following high-level directions (Guanetti et al., 2018; Paden et al., 2016).

The ultimate goal in decision-making is for the autonomous vehicle to move from point A to point B without accidents. However, this has been an issue to achieve (accidents still occur). Consequently, problem formulation related to decision-making had to be conducted to minimise the number of hazardous situations and ultimately reduce the number of accidents. Two problem planning were utilised, which are *path planning* and *trajectory planning*; these plannings were conducted using mathematical modelling. The number of accidents has been proven to have reduced, but they still occur, which is still a problem that needs great attention. (Badue et al., 2021; Paden et al., 2016)

When considering real-life situations in road driving, there are few simple situations – this means most situations in real-life autonomous vehicles will be involved in complex situations more than in easy situations. This triggers and requires autonomous vehicles to make quick, smart, and reliable decisions accurately and precisely. However, this is still a problem in real-life situations. Though it can be noted that the decision-making of autonomous vehicles depends on the method used to formulate the decisions for particular situations, there can be a generic way to follow in the formulation (Figure 14). (Schwarting et al., 2018)

Sensors

The sensors play one of the most important roles in the AV's performance—it provides data to the perception system and then to the rest of the other systems to make the

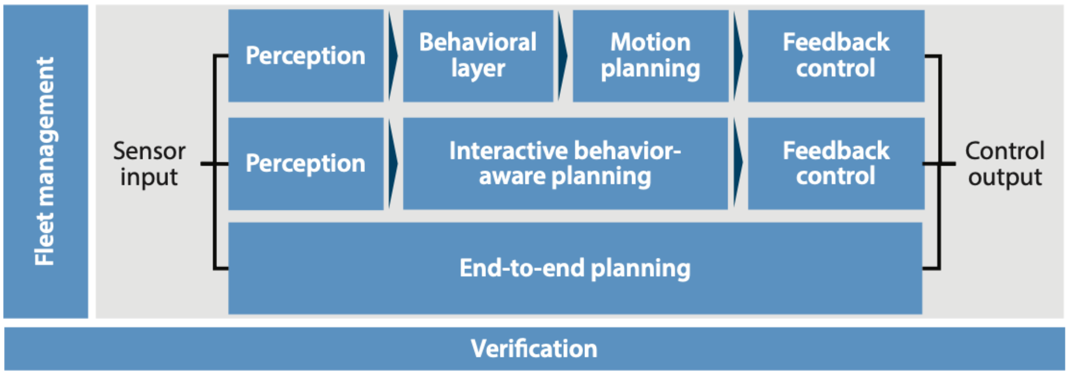


Figure 14: The generic decision-making formulation for autonomous vehicles (adopted from (Schwarting et al., 2018)).

vehicle act or react as expected. Therefore, the sensors' performance has to be very accurate to avoid any mistakes that would injure the passengers, pedestrians, and the environment. In Figure 15, it can be seen that these sensors use different electromagnetic spectrums in their operations; consequently, their performance is different, and they can be combined to do different things. The definitions of these sensors can be seen in section 1.2.

Yan et al. (2016) described the ranges of these sensors in their study that explored the reliability of the AV's *eyes* and the security of the sensors used by AVs. The ranges are discussed as follows.

- **Proximity** ($\approx 5m$). Ultrasonic proximity sensors are designed to find obstacles within a few meters of the AV's body. They are mostly intended for situations with low speeds.

- **Short Range** ($\approx 30m$). In this range, lane departure warning and traffic sign recognition are done with forward-looking cameras, and parking assistance is done with backward-looking cameras. Cross-traffic alert and blind spot identification are two functions of Short-Range Radars (SRR).

- **Medium range** (≈ 80 to $160m$). These ranges are associated with LIDAR and Medium-Range Radars (MRR) and aid in the identification of pedestrians and collision avoidance.

- **Long Range** ($\approx 250m$). These are Long-Range Radars (LRR) which were developed to support *adaptive cruise control* at high speed.

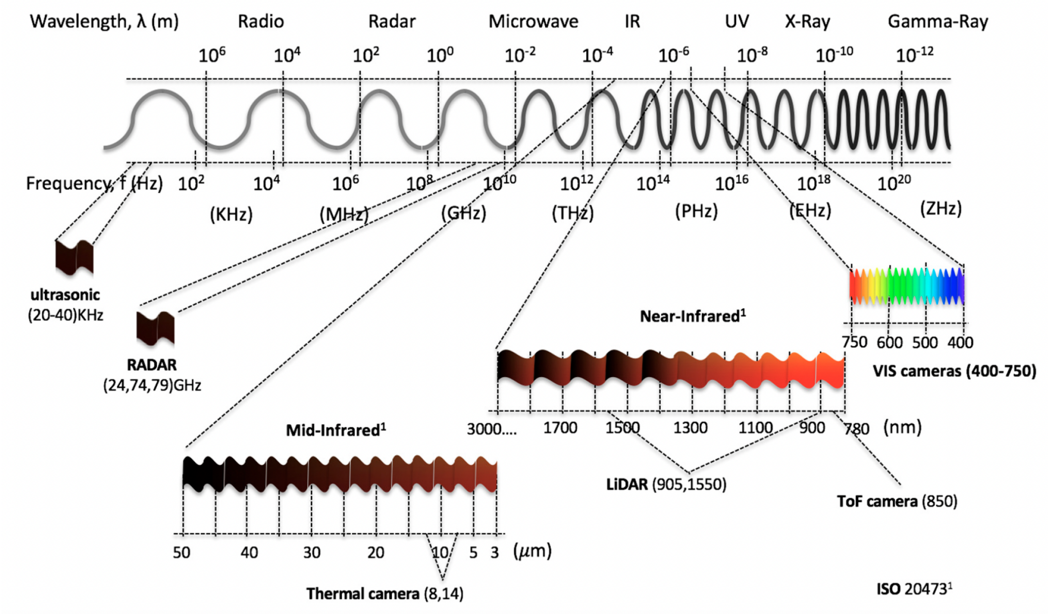


Figure 15: The electromagnetic spectrum of AV's sensors (adopted from (Vargas et al., 2021)).

Furthermore, the kind and placement of sensors that allow an autonomous vehicle to perceive its surroundings are crucial in the performance of the AV. The placement of the sensors and their expected distance range are shown in Figure 16. The red regions show the LIDAR coverage, the grey areas display the camera coverage surrounding the vehicle, the blue areas show the coverage of short-range and medium-range radars, and the green areas show the coverage of long-range radar.

With these sensor placements and range descriptions in place, it is necessary to determine what factors they can detect. Ignatious et al. (2022) and Yeong et al. (2021) provided a comparison of some of the sensors, which looked the same, except that Ignatious et al. (2022) did not specify if the fusion of the sensors concerning the range is possible or not. The comparison by Yeong et al. (2021) can be seen in Figure 17. Furthermore, before Ignatious et al. (2022) constructed their comparison table (provided in Appendix B), they firstly tabulated different types of sensors with different properties that can be used to select the best performing sensor(s) (these can be found in Appendix B).

In addition, Vargas et al. (2021) conducted a study that focused on the effects of various meteorological conditions (such as precipitation, fog, lightning, etc.) on the perception systems of AVs. The most popular AV sensor and communication kinds, includ-

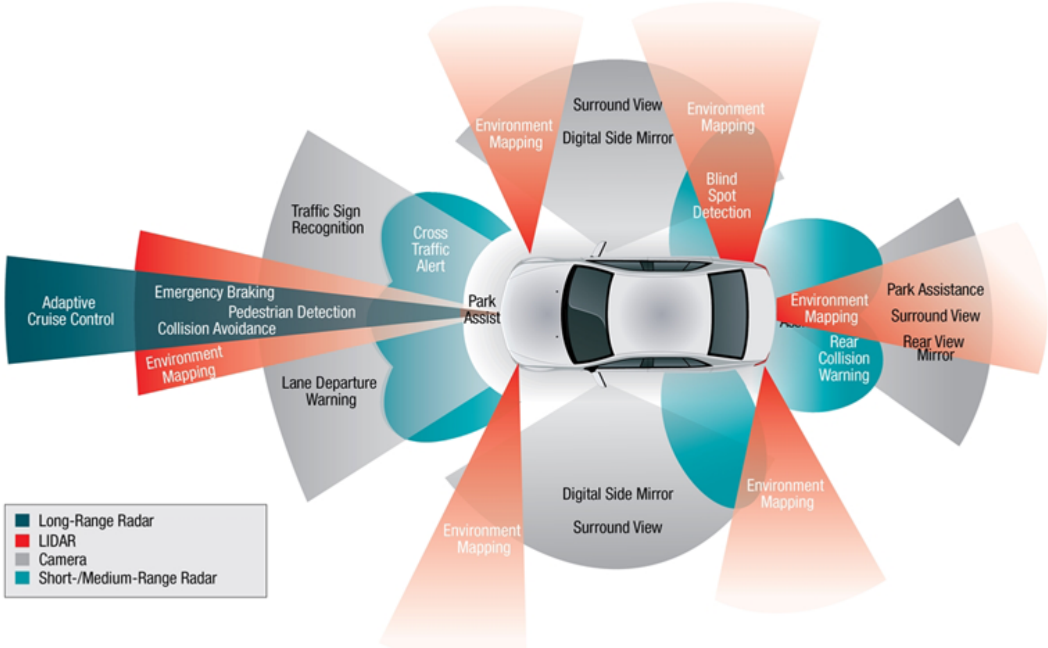


Figure 16: The sensor placement overview on AVs with regards to the range detection for perception system (adopted from (Ignatious et al., 2022; Yan et al., 2016; Yeong et al., 2021)).

ing Radio Detection and Ranging (RADAR), LIDAR, ultrasonic, camera, and Global Navigation Satellite System (GNSS), were the main focus. Therefore, a comparison of these sensors was conducted (excluding the GNSS); please see Figure 18.

In consideration of Figure 17, the ✓ symbol denotes that sensors operate completely under specific conditions, the ∼ symbol denotes that sensors perform reasonably well under specific conditions, and the ✗ symbol denotes that sensors do not operate well under the specific factor relative to other sensors.

There are different kinds of AV sensors (RADAR, LIDAR, ultrasonic, camera, GNSS, etc.), and there are different types of sensors of the same AV sensor kind. Therefore, different AV sensor kinds' performance depends on their types.

2.2 Inherent complexity of autonomous vehicles

The inherent complexity of an AV refers to what makes it difficult for the vehicle to understand, especially in its design, to comply with certain conditions in a certain given period. Furthermore, the inherent complexity of an autonomous vehicle also refers to

Factors	Camera	LiDAR	Radar	Fusion
Range	~	~	✓	✓
Resolution	✓	~	×	✓
Distance Accuracy	~	✓	✓	✓
Velocity	~	×	✓	✓
Color Perception, e.g., traffic lights	✓	×	×	✓
Object Detection	~	✓	✓	✓
Object Classification	✓	~	×	✓
Lane Detection	✓	×	×	✓
Obstacle Edge Detection	✓	✓	×	✓
Illumination Conditions	×	✓	✓	✓
Weather Conditions	×	~	✓	✓

Figure 17: The AV sensor comparison (adopted from (Yeong et al., 2021)).

Feature	LiDAR	RADAR	Camera	Ultrasonic
Primary Technology	Laser beam	Radio wave	Light	Sound wave
Range	~200 m	~250 m	~200 m	~5 m
Resolution	Good	Average	Very good	Poor
Affected by weather conditions	Yes	Yes	Yes	Yes
Affected by lighting conditions	No	No	Yes	No
Detects speed	Good	Very good	Poor	Poor
Detects distance	Good	Very good	Poor	Good
Interference susceptibility	Good	Poor	Very Good	Good
Size	Bulky	Small	Small	Small

Figure 18: The comparison of AV sensors based on their characteristics (adopted from (Vargas et al., 2021)).

how difficult it is for an autonomous vehicle to be deployed in a certain country, city, or province. This part of the literature review focuses on both aspects regarding the inherent complexity of an autonomous vehicle. Literature research was conducted on this subject matter, and the results are discussed in sections 2.2.1 through 2.2.3. Please note that this section answers *Research Question 3* outlined in section 1.8.

2.2.1 Technological competency

In consideration of Africa's technology, the Fourth Industrial Revolution (4IR) is not evident (Ndung'u and Signe, 2020), and this is not a good sign as it must be evident since it brings about the advanced lifestyle that every continent needs. Moreover, the emerging technology that comes with the autonomous vehicle is complex and has risks associated with it, that is, *geopolitical* and *socio-economic* risks (Tan and Taeihagh, 2021). With this in consideration, any continent, country, province, or city that wants to introduce autonomous vehicles will experience difficulties, especially the African continent since it lacks the 4IR technology, and the technology associated with the autonomous vehicle is getting more advanced than the 4IR technology. Therefore, technological incompetency in continents such as Africa is an inherent complexity that needs close attention.

2.2.2 Sensor functionality issues

Autonomous vehicles can self-drive since it has an embedded AI technology in them, specifically, the sensors that have been programmed to function in a certain way. All the decisions made by a fully autonomous vehicle are directly from the data gathered by the sensors and analysed. Therefore, the sensors must function as expected to avoid obvious disasters (accidents). Now, if any of the sensors in an autonomous vehicle fail or provide unclean data, then that will be a big problem, and that is one of the inherent complexity found in the autonomous vehicle – the sensors do gather dirty data when there is a bad or abnormal weather condition (such as snow, heavy rain, etc.), this can occur in any sensor category (see Table 2 for sensor categories). Human drivers also experience similar problems with bad or abnormal weather conditions. (Ma et al., 2020)

There are three categories that autonomous vehicle sensors fall under as described by Ma et al. (2020); please see Table 2, which provides these categories and their respective definitions.

There is a security problem with the sensors—sensors are being attached using a phenomenon called *physical channel* as stated by Yan et al. (2016). Using physical channels is the key distinction between sensor attacks and cyberattacks. Most of the time, sensor assaults use the same physical channels as the targeted sensor, which might tamper with or alter sensor results. Falsified readings could have unintended system effects since sensors are typically trusted and categorised as the lowest control system layers. Further, Sensor attacks are less advantageous than cyberattacks because they require more technology, have a longer exploitation time, and require a higher level of knowledge. Distinct sensors may rely on very diverse physical principles, which necessi-

Table 2: Autonomous vehicle sensor category definitions

Category	Definition
Self-sensing	Measures the current state of the geo-vehicle using proprioceptive sensors. These include the autonomous vehicle's acceleration, velocity, and steering angle. Note: *Proprioceptive information* is usually determined by units such as Inertia Measurement Unit (IMU)s and/or odometers, which are pre-installed.
Localisation	The determination of the autonomous vehicle's location on a global and local scale by using tools such as Global Positioning System (GPS) and dead reckoning by IMU readings
Surrounding sensing	Perceives road markings, road slopes, weather conditions, traffic signs, and the state of obstacle (such as other surrounding vehicles) by using exteroceptive sensors

tates using quite different strategies to combat them, resulting in low transplantability. (Yan et al., 2016)

Therefore, to combat this type of attack, Yan et al. (2016) examined some of the sensors by analysing *jamming* and *spoofing* attacks in their physical channels. In the *jamming attack*, the sensors are made to withstand environmental noise that occurs during typical working circumstances. In the *spoofing attack*, when sensors are positioned incorrectly, it is possible to get real physical signals from the incorrect source. All this was done so that the vehicle's sensors might result in crashes and jeopardise the security of AVs. Consequently, Yan et al. (2016) provided hardware and software countermeasures that may strengthen sensor resilience against the attacks to relieve the problems.

2.2.3 Accident accountability

Accidents are inevitable, especially when human beings are involved. Furthermore, accountability when it comes to accidents is an issue, and that is why legal framework and regulations are integrated – this is one of the most important requirements for autonomous vehicles' deployment (Singh and Saini, 2021). The main question in this situation is *who should be held liable for either fatal or none fatal accidents?* This question does not have a straightforward answer. According to Borenstein et al. (2019), if there was an accident that involves an autonomous vehicle, it does not make sense that the technology itself can be held responsible but the designers, car dealer(s), manufacturers, and/or other people that could be identified as guilty. This claim supports what Mackie (2018) stated, that is, human drivers should remain liable for the accident depending on the automation installed in the vehicle. For example, suppose the automation is in level 4 or 5 (highly or fully automated). In that case, the plaintiffs are responsible for identifying who should be held accountable, which could be the manufacturers, maintainers, or who contributed to the autonomous vehicle.

2.3 Reliability analysis

To define what is meant by reliability analysis, the term *reliability* should be defined. *Reliability* is defined as the ability of a system to accomplish a task required under the conditions provided for a certain time interval. Although the term *reliability* can be used to refer to a product's or system's overall performance, in engineering disciplines, reliability refers to a specific measure that can be quantitatively evaluated. Since several reliabilities can be calculated for a systems or subsystem, it is crucial to identify the various jobs that make up the system or subsystem. Reliability can be calculated for each task independently because a subsystem may have many tasks. (Ahmadi et al., 2020; Bastidas-Arteaga and Soubra, 2014)

Numerous analyses used to assess and enhance the quality of goods, services, and systems are collectively referred to as reliability analysis. One could think of reliability as a potential for equipment failure. Data is therefore required to calculate equipment failure rates. The Time Between Failures (TBF) and the Time To Repair are the two most frequently required pieces of data to calculate dependability (TTR). According to the definitions given above, the probabilistic behaviour of an item's reliability can be summarised as follows (Ahmadi et al., 2020):

$$Pr(T \leq t) = \int_0^t f(t)\, dx = F(t)\, for\ t \geq 0 \tag{2.1}$$

While (t) is the probability density function of the time between failures, F(t) represents the likelihood that the item will fail from time zero to t. The reliability can be expressed as follows given that F(t) is a function of uncertainty:

$$R(t) = 1 - F(t) = \int_t^\infty f(t)\, dx R(t) = Pr(T \leq t). \tag{2.2}$$

The failure rate is a crucial component of reliability analysis because it shows the likelihood that a component will fail over the course of its lifespan, and it is defined as follows:

$$h(t) = \frac{f(t)}{R(t)}. \tag{2.3}$$

To perform an analysis of the dependability of engineering systems, numerous techniques have been developed. These techniques are especially helpful for analysing more intricate engineering configurations than those with standard reliability. The following are the most significant and widely utilised reliability analysis techniques (Ahmadi et al., 2020; Dhillon, 2009):

- Statistical analysis method—this method typically performs reliability assessment for both repairable and non-repairable systems using MTBF and MTTF, respectively.

- Failure mode and effects analysis method—this is a method for analysing data that leverages the pre-occurrence prevention law to find possible sources of failure. By preventing failures, this strategy aims to increase security.

- Markov method—The Markov model includes a list of the system's potential states, potential routes between them, and the parameter rates governing those transitions. Transition typically entails failures and repairs in reliability analysis. Each state is typically depicted as a bubble when a graphical Markov model is given, with arrows denoting the direction of transition between the states.

- Fault tree analysis—A top-down logical and graphical representation illustrating the failure and its causes is used in the fault tree analysis method. A set of signs and symbols are used to depict the connections between the failures and their causes in the fault tree analysis diagram, which displays all system, subsystem, and collection faults.

2.4 Chapter summary

Considering the discussed findings presented in this chapter, autonomous vehicle state-of-the-art highlighted functional requirements as the starting point when designing. Though there is a precise integration between functional requirements, physical design, and the related/expected performance of the vehicle, it is essential to start with the requirements, followed by design, and finally, the performance. When considering the vehicle's designs, they can vary, and this dissertation analysed all the provided designs so that an optimal and standard design is provided. The vehicle's performance is assessed when the vehicle's prototype has been fully designed and built. Additionally, this dissertation reviewed essential performances related to the vehicle's intelligence. Furthermore, though different kinds of sensors can have various performances, it is still necessary to identify and select sensor types that yield the best performance for more efficient sensor data gathering.

The complexities that can be found in autonomous vehicles can threaten people's safety, but they are controllable and manageable. The two main issues discussed in this chapter that poses threats are *technological inconsistency* and *sensor functionality issues*. Sensor functionality issues, for example, pose a significant threat if there is a functional error such that the decision made by the vehicle is incorrect due to the wrong readings by the sensor(s). Such error could cause a fatal accident – for example, if the autonomous vehicle detects a pedestrian as *not moving*. Still, they were moving, and the vehicle was supposed to stop at that moment.

Reliability analysis was discussed, there are a lot different kind of reliability analysis methods that can be utilised. This dissertation maintained the reliability definition provided in this chapter, and an analysis utilised was the *statistical analysis method* as discussed in Chapter 3.

Chapter 3

Research methodology

The purpose of this chapter is to provide an in-depth methodology that was followed during the cause of this study. Two research methodologies were adopted for this dissertation, which is (1) *Design Science Research (DSR)* and (2) *Systems Thinking* methodologies. Two methodologies were utilised in this dissertation because the theories and methods utilised needed both methodologies in terms of how to understand systems (using systems thinking methodology) and how to come up with an appropriate model (using DSR methodology). A unification of these methods was therefore established, as discussed in this chapter.

Looking closer first at DSR methodology, Goecks et al. (2021) and Aken (2004) defined DSR as an artifact-oriented discipline that seeks to either design a solution to an existing problem or a new problem. The artefact associated with this dissertation is the *reliability model* that was created to provide *explanatory solution* to an existing problem of complexity as described in section 1.6. The approach itself is best described in the book by Dresch et al. (2015) (see Figure 19 for details with regards to the actual steps with explanations).

Some steps shown in Figure 19 were utilised in this dissertation; that is, some of the steps were adjusted to suit this dissertation's actual methodology that was followed (see Figure 20). In addition, the following steps were omitted or modified to support the methodology in this dissertation.

- **Systematic literature review** (*modified*). This step was modified to *Literature review* because a non-systematic literature review was conducted in this dissertation.

- **Identification of the artefacts and configuration of the classes of prob-**

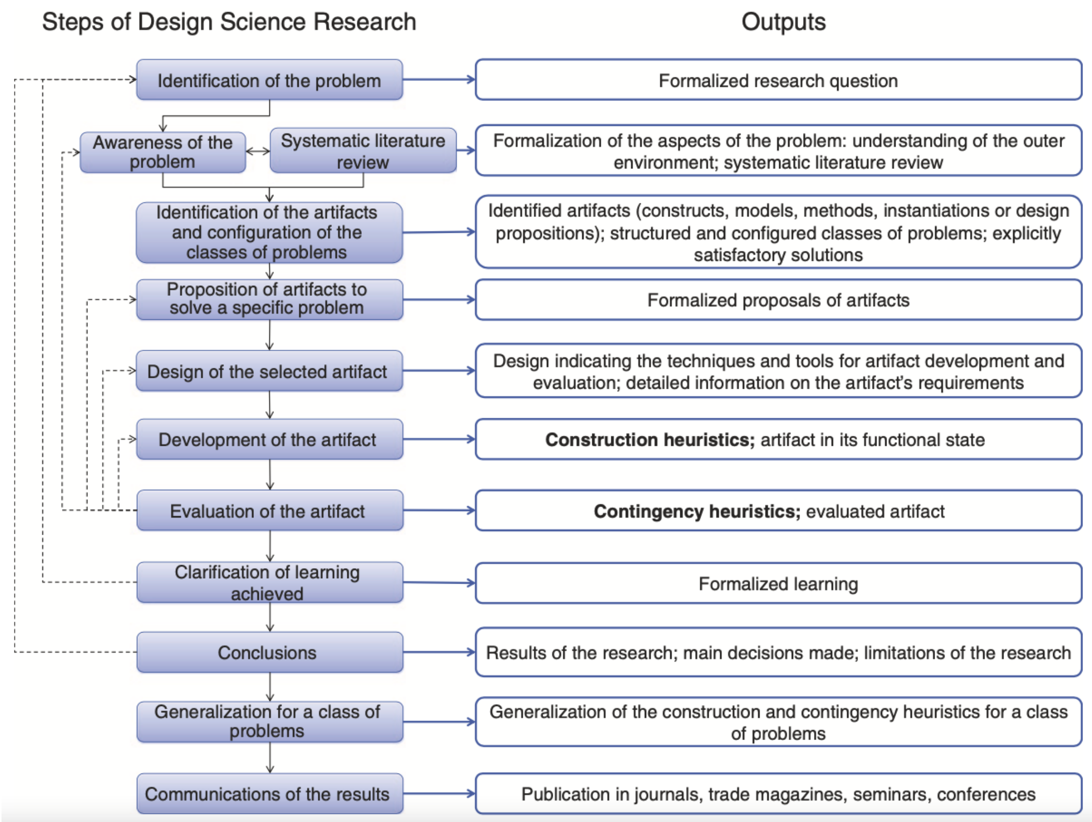


Figure 19: The DSR methodology with detailed steps (adapted from (Dresch et al., 2015)).

lems (*modified*). There is no multiple class of problems that were considered. Hence, renamed to *Identify the artefacts*.

- **Proposition of artefacts to solve a specific problem** (*omitted*). No multiple artefacts were proposed for this dissertation; hence, this step was irrelevant.

- **Generalisation for a class of problems** (*omitted*). There is only one class of problem considered in this dissertation.

Therefore, the adjusted DSR methodology followed by this dissertation with the above-mentioned changes can be seen in Figure 20. Furthermore, the added field (in Figure 20) named *outputs* shows the corresponding outputs provided by this document to address all the steps of the methodology.

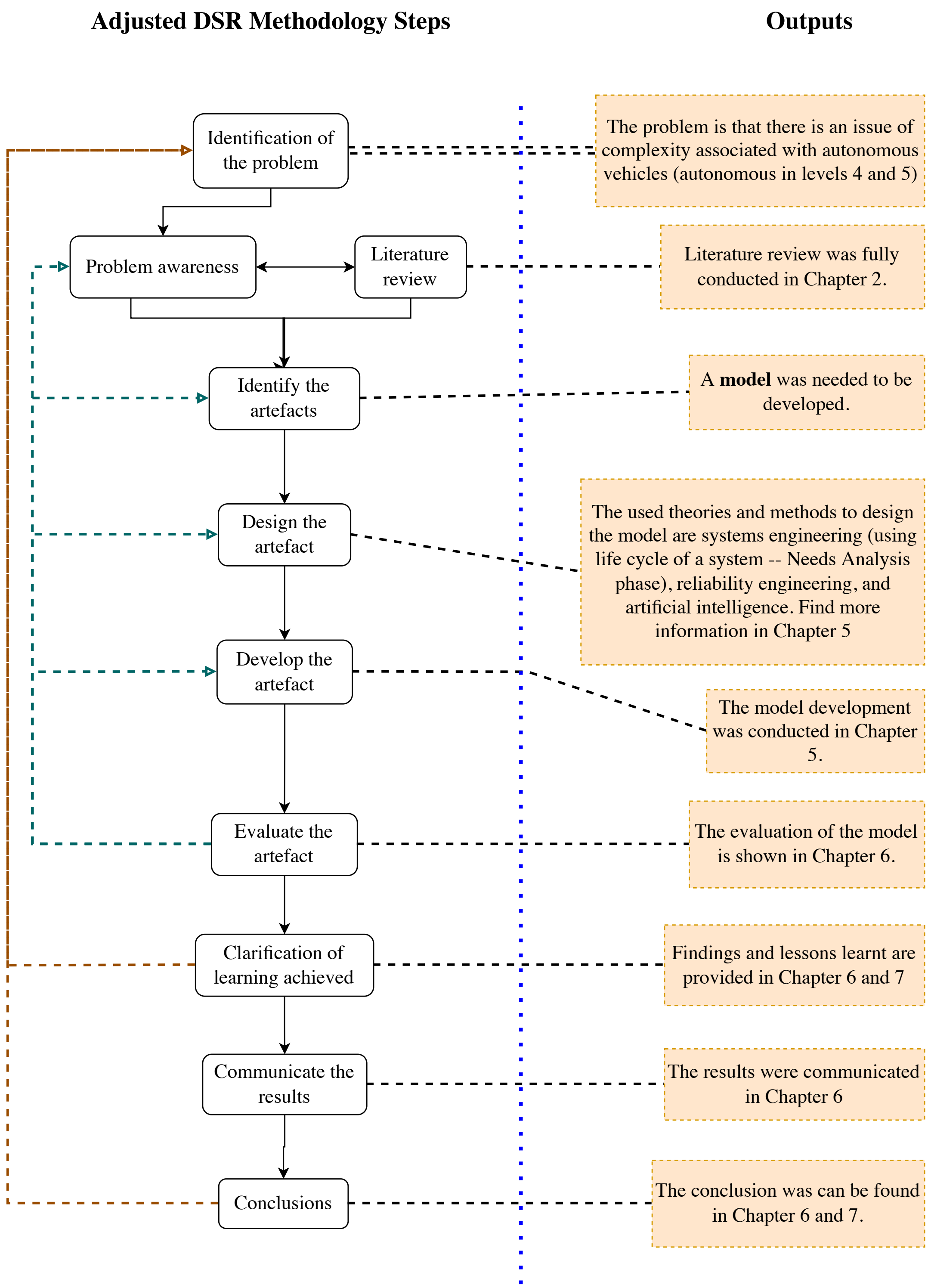


Figure 20: The adjusted DSR methodology followed by this dissertation.

It should be noted that the methodology depicted in Figure 20 is iterative. During the evaluation of the artefact, corrections can be made by going back to previous steps to modify them as needed. Furthermore, when conducting the results and conclusion of the research, they should be made sure to address the identified problem.

Looking at the *systems thinking* approach, the definition provided by Arnold and Wade (2015) is *systems thinking is a collection of complementary analytical skills that helps to get better at recognising and comprehending systems, forecasting their behaviours, and coming up with changes to them that will have the intended results and these abilities function as a system*. Systems thinking must have aspects such as *interconnectivity, feedback loops, stocks and flow, non-linearity relationship, modelling*, and it must be *system-related* (Figure 21) (Arnold and Wade, 2015).

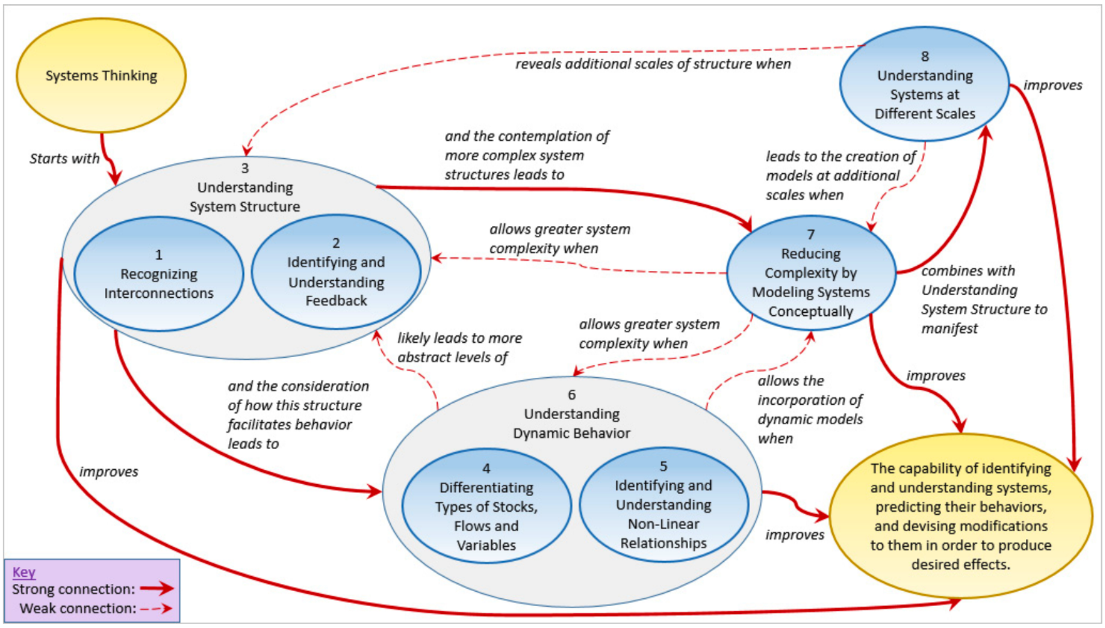


Figure 21: The systems thinking framework (adopted from (Arnold and Wade, 2015)).

Provided that the systems thinking approach is also a good approach to be utilised in the systems engineering discipline (this dissertation focuses mainly on systems engineering and Artificial Intelligence (AI) theories as discussed in the previous chapters – Chapter 1 and 2). Consequently, the integration (or unification) of this approach and DSR methodology was established. Furthermore, the integration was expanded by adding the framework (Figure 22) that was created using the application of systems engineering theory (adopted from Dresch et al. (2015))—which was developed to fulfil the steps in Figure 19 that are as labelled *Design of the selected artefact*, *development* *of the artefact*, and *Evaluation of the artefact.*

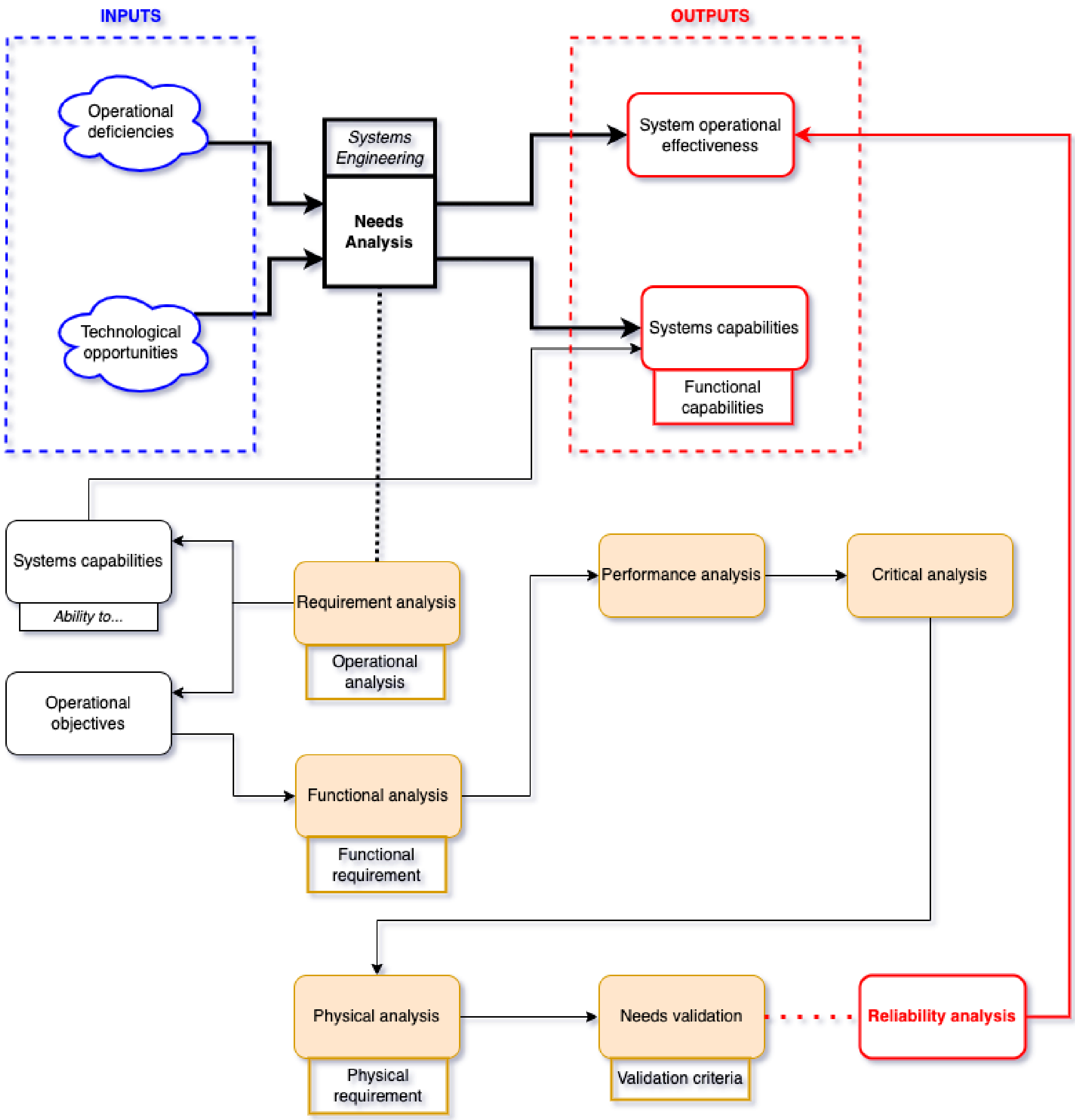


Figure 22: The solution approach framework using the systems engineering theories (application of Reliability Engineering, AI/Robotics, and Needs analysis – the life-cycle of a system).

In Figure 22, it can be noted that the first part of the framework (the top part with inputs and outputs) is the *Needs analysis* phase of the system engineering life cycle of a system (discussed in depth in section 1.4). To apply this part of the framework (mainly the inputs), systems thinking methods had to be adopted. Then the internal analysis in the needs analysis was created (the second part of the framework). The link between the two parts is initially established with the *black-dotted line*; therefore, the first step to the analysis is the *requirement analysis*. The last step is the *needs validation* which is directly linked to the application of the *reliability analysis*. The complete analysis of these phases was done in Chapter 3.1.

Furthermore, the integration of the DSR methodology, systems thinking approach, and framework can be seen in Figure 23.

The purple dotted lines represent the *indirect and strong link* between the steps conducted in the DSR methodology. The relationship can be interpreted as follow:

- [PURPLE DOTTED LINKS] The DSR steps that have this type of link require that particular *systems thinking* stage to be completed first before the step can be fulfilled. For instance, the step labelled *develop the artefact* from the DSR steps is indirectly completed by conducting *modelling the system conceptually* (it can be noted that this DSR step is also connected to a solid green line, which is directly completed by conducting *reliability analysis (the evaluation of the model)* and providing *system operational effectiveness*, the output of conducting *needs analysis*).

The solid green lines represent a strong and direct relationship between the solution approach framework and the DSR methodology steps. The relationship can be interpreted as follows.

- [GREEN SOLID LINKS] The DSR steps are directly fulfilled or completed by the linked outcomes from the solution approach framework. For instance, the step labelled *evaluate the artefact* from the DSR steps is directly completed by providing *systems' capabilities* written out starting with the words *the ability to*, which is the output of conducting *needs analysis*.

The main methodology is the *adjusted DSR*; however, the integration was necessary for completing the dissertation, as explained earlier in the chapter. Therefore, the application of the integrated methodology (Figure 23) as it appears in this dissertation is outlined as follows.

- STEP 1: **Identification of the problem.** The identified problem can be seen in section 1.6—the problem of Autonomous Vehicle (AV) complexity.
- STEP 2: **Literature review (and problem awareness).** The literature review was conducted in Chapter 2.

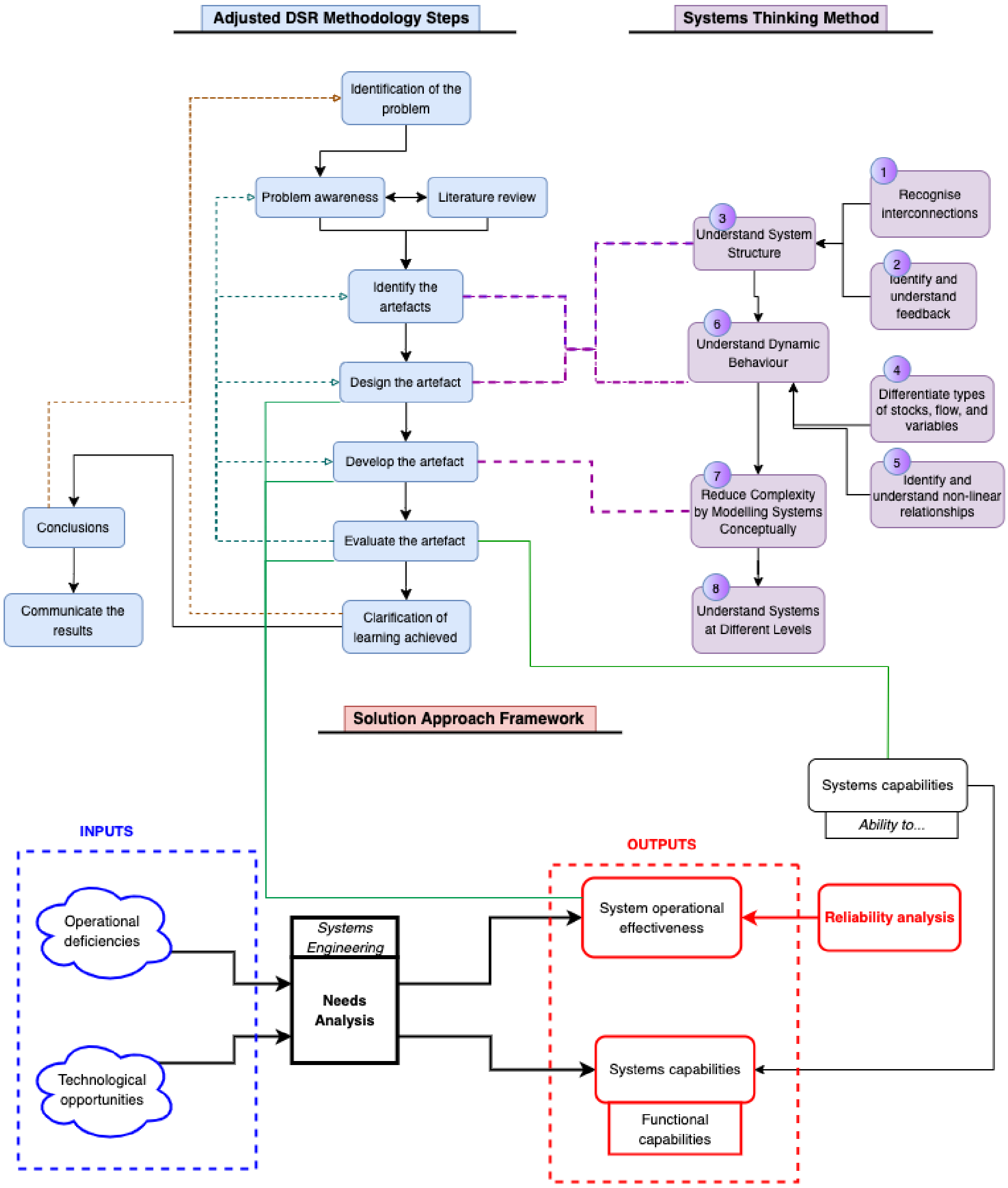


Figure 23: The integration of DSR methodology, systems thinking approach, and the solution approach framework.

- Step 3: **Identify the artefact.** The identified artefact in Chapter 1—the reliability analysis (and functional analysis) model. Systems thinking was integrated into this step by following steps 1 to 6 of the systems thinking method in Figure

20.

- STEP 4: **Design the artefact.** The artefact was designed in Chapter 3.1 through Chapter 3.1.2. Systems thinking was integrated into this step by following steps 1 to 6 of the systems thinking method in Figure 20.

- STEP 5: **Develop the artefact.** The artefact was developed in Chapter 3.1.2. Systems thinking was integrated into this step by following step 7 of the systems thinking method in Figure 20.

- STEP 6: **Evaluate the artefact.** The evaluation of the artefact was conducted using RStudio and Anaconda (Python) programming platform, which can be seen in Chapter 4.

- STEP 7: **Clarification of learning achieved.** The lessons learnt from the evaluation are in Chapter 4.

- STEP 8: **Communicate the results.** The results were communicated in Chapter 4.

- STEP 9: **Conclusions.** The conclusions related to lessons learned that were drawn are in Chapter 4.

The problem and requirements analysis whereby the basis and foundation of developing the model using the described methodology can be seen in section 3.1.

3.1 Research approach

The purpose of this section was to provide the approach taken to address the objectives outlined in section 1.7. Furthermore, the related models (functional and physical) were developed and discussed.

3.1.1 Functional and physical elements modelling

This section focuses on the system (the Autonomous Vehicle (AV)'s system) requirements, that is, the functional and non-functional requirements of the system. *Functional requirements* are the tasks or processes that the system must do or perform, and the *physical requirements* are the behavioural properties that a system must exhibit. Such as physical and technical operating environment, performance, legal requirements, and security (Kossiakoff et al., 2011). The solution approach framework (Figure 22, in Chapter

3) consists of the processes that were deployed in this chapter to provide the functional and physical requirements. Figure 24 presents these processes (shaded in peach colour).

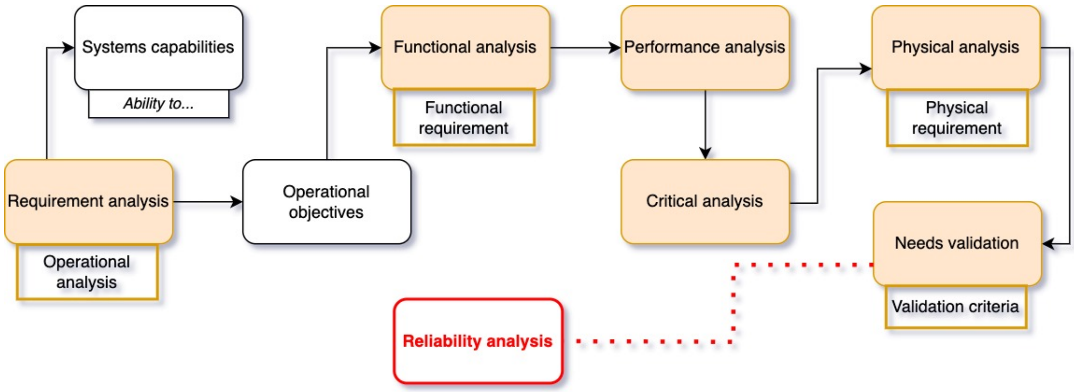


Figure 24: The functional and non-functional requirement processes.

The processes in Figure 24 are now discussed relating them to the AV's state-of-the-art design and reviews (as discussed in section 2.1). The definitions for *performance* and *critical* analysis processes were described in this chapter, but the other four processes were described in section 1.4.

3.1.1.1 Requirement analysis

The requirement analysis produces two sets of outcomes, the *systems capabilities* and the *operational objectives*.

Firstly, the *systems capabilities* (the capacity of a system to carry out a specific action or produce a desired result under a specific set of circumstances or conditions) identified are related to the intelligence of the AV system. The conditions that a level 4 or 5 AV should meet using the intelligent components or subsystems are: (1) move the vehicle from point A to B; (2) control the vehicle (steer, slow down, accelerate, etc.)—this condition is the most complicated as it also involves obeying traffic rules and introduces ethics when an accident occurs; (3) keep passengers safe; and (4) arrive at the destination on time. For these conditions to be fulfilled, the following intelligent capabilities were identified.

1. Ability to combine a range of sensors, including Global Positioning System (GPS), odometry, radar, Light Detection and Ranging (LIDAR), Sound Navigation Ranging (SONAR), thermographic cameras, and inertial measurement units, to sense their environment.

2. Ability to control systems and analyse sensory data to determine the best routes to take, as well as barriers and essential signage in a more advanced manner.
3. Ability to detect lanes using a camera system to read the marking on the road and keep the vehicle within its right (or safe) lane.
4. Ability to make safe decisions based on how other vehicles surrounding the AV are behaving using a *vehicle to vehicle communication technique*. Whereby the AV must be aware of the position, velocity, and trajectory of any close vehicles.
5. Ability to use a *decision-making system* built into it (the AV) to make informed decisions, such as reacting when other vehicles behave abnormally, to prevent accidents.

Secondly, the *operational objectives* focuses on the goals of a technology-driven development (in this case) which must include an operational concept. Therefore, the operational objectives for the AV should be in-line with the following aspects.

1. The final state of the operational environment or scenario should be addressed in the objectives; this concentrates on what the system will achieve in a broad sense.
2. The system's goal and what constitutes a need's satisfaction should be covered in the objectives.
3. When taken as a whole, the objectives explains the *why* for the system's necessity (why is this system needed).
4. The verb *provide*, which is the infinitive in most objectives, is commonly used but it is not a necessity.

With these four aspects outlined, the operational objectives associated with AV (gathered from the literature provided in Chapter 2) are as follows.

- **Operational objective 1** *Provide advancement to vehicles to bring about the new era of smart cities and integration of Artificial Intelligence (AI) to vehicles.*
- **Operational objective 2** *Transport human beings from one point (location) to another without any harm and on time.*

Literature does not explicitly provide these operational objectives, they are a result of the requirement analysis on the literature provided in Chapter 2. Now, with the operational objectives in place, the functional analysis was conducted and the functions needed to accomplish these objectives were provided.

3.1.1.2 Functional analysis

The product of conducting this analysis are the functional requirements that describes the actions that should be taken to fulfil the system-wide operational objectives. These requirements were extracted from the literature discussed in Chapter 2, mainly from Matthaei and Maurer (2015), Vaicenavicius et al. (2020), and Badue et al. (2021). The functional requirements identified and discussed in this dissertation focused on two type of systems that consists of the intelligence that is integrated in the AVs. These systems are, the *perception system*—the vehicle should *always* know how to identify its environment, and the *decision-making system*—a system that deals with what the AV should do and how it should do it. The functional requirements are outlined in Table 3.

Table 3: The AV system-wide functional requirements related to the intelligent components and subsystems.

System type	**Functional requirement**
Perception	1. The system should be able to perform localisation by using different methods. 2. The system should be able to map obstacles offline using metric representation methods or any other available method. 3. The system should be able to conduct road mapping using methods such as metric representation and topological representation. 4. The system should be able to track moving obstacles—several methods are used to accomplish this task. 5. The system should be able to detect and recognise traffic signs—traffic light detection and recognition, traffic sign detection and recognition, and pavement marking detection and recognition are the commonly used methods.
Decision-making	1.The vehicle should be able to plan route using goal-directed, separator-based, hierarchical, bounded-hop, or any combination of the abovementioned techniques. It needs a lot of data and data mapping might have to be executed. 2. The vehicle should be able to select its expected behaviour when faced by a situation, regardless of the complexity of the situation. 3. The vehicle should be able to plan its motion. 4. The system should control the vehicle using direct hardware actuation control and path tracking. 5. The vehicle should be able to know its state at all times (self-perception) so that it reacts appropriately. 6. The system should be able to learn from previous encounters so that it can react more quickly and naturally in complex situations—this triggers the need for extensively integrating *machine learning*.

It can be argued that a *control system* is crucial and is needed in AV, which is correct.

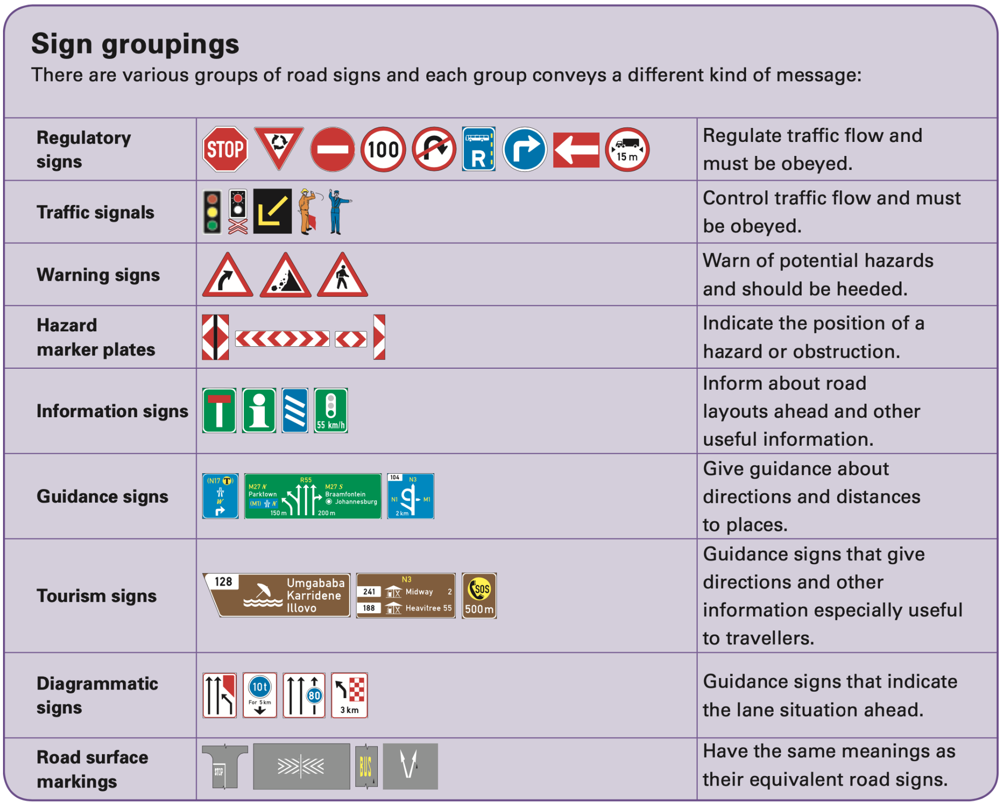
However, in this dissertation the control system was concluded to be a subsystem of the decision-making system since *the control system's actions are only executed whenever a certain decision has been made* by the decision-making system.

With the functional requirements outlined, the physical requirements of AV can also be provided, however, it is more effective to conduct performance and critical analysis beforehand. This will allow the recommended physical requirements to be the best selected requirements based on how and if the functional requirements were met. It can be noted that this approach would still work the same if it was applied to a non-existing system or a newly created system. This is due to the fact the newly created system would need to have parts or components that best fit the new system and will provide optimal system performance. This means components or parts will have to be evaluated to test their performance before they are utilised.

3.1.1.3 Performance and critical analysis

The performance this dissertation focused on is directly linked to both *operational objectives* discussed above in this chapter. Therefore, if the AVs obey traffic rules, that means the vehicle is capable of meeting the operational objectives that were discussed in the *operational objectives'* phase. This is due to the fact that if an AV is capable of transporting a passenger from point A to B without any harm to anyone or anything (successfully protecting both its passenger(s) and its external environment), and doing that *consistently*, it can be deduced that it obeys all the traffic rules. This will allow AVs to be commercialised to the cities that are capable of providing what these vehicles require to function as desired. Further, this provides the opportunity for cities to have more advanced vehicles and hence moving to or closer to a *smart city era*, depending on the technological state of that particular city.

Consequently, the *reliability* of intelligence of AVs was conducted focusing on some of the vehicle brands that have worked on the AV concept, manufactured, and tested them. The analysis of these brands based on the categories in Table 4 is provided in section 3.1.2.1. It should be noted that Table 4 provides two types of K53 traffic rules as described by Hoole (2013), though there are three types of traffic rules described by Hoole (2013). This dissertation focused only on two types as the third rule type focuses on *knowing the controls* of a vehicle and an AV is assumed to know what controls it has and how to use them.



Sign groupings

There are various groups of road signs and each group conveys a different kind of message:

Regulatory signs		Regulate traffic flow and must be obeyed.
Traffic signals		Control traffic flow and must be obeyed.
Warning signs		Warn of potential hazards and should be heeded.
Hazard marker plates		Indicate the position of a hazard or obstruction.
Information signs		Inform about road layouts ahead and other useful information.
Guidance signs		Give guidance about directions and distances to places.
Tourism signs		Guidance signs that give directions and other information especially useful to travellers.
Diagrammatic signs		Guidance signs that indicate the lane situation ahead.
Road surface markings		Have the same meanings as their equivalent road signs.

Figure 25: The summary of road signs, signals, and markings (adopted from Hoole (2013)).

Table 4: Category descriptions that focus on checking the reliability of AV with respect to traffic rules

Category type	Traffic rule
Road signs, signals, and markings—the purpose is to safely regulate traffic flow, warn drivers or motorists of the circumstances on the road ahead, provide useful and necessary information, and provide guidance on routes and destinations.	Figure 25 provides the summary of the symbols and short descriptions of each symbol type. The following is the summary of what the AV should do or is expected to do regarding the types of symbols or *sign groupings*. **Rule 1**. Regulatory signs—*must obey.* **Rule 2**. Traffic signals—*must obey.* **Rule 3**. Warning signs—*must heed to avoid potential danger.*

	Rule 4. Hazard marker plates—*must heed to avoid potential danger.* **Rule 5**. Information signs—*must understand to react appropriately.* **Rule 6**. Guidance signs—*must be built in the AV*, for instance, using a GPS. **Rule 7**. Tourism signs—*not important* simply because AVs must have built-in GPS which they can use to navigate to a desired tourist's destination. **Rule 8**. Diagrammatic signs—*must heed to select an appropriate lane.* **Rule 9**. Road surface markings—*must obey.* **Rule 10**. Hand signals—*must obey* if it is traffic officer, and *must heed* if it is other motorists.
Rules of the road—the purpose of the rules of the road is to *control traffic*, *provide safety*, and *safeguard each individual's rights to use the road.* Speed restrictions, lane discipline, parking, and lighting all have regulations that must be adhered to. Following the traffic laws is required, and doing so will significantly lower the likelihood of accidents, injuries, and fatalities on the roadways.	The following *must be obeyed.* **Rule 1**. The vehicle must drive on the correct side (left or right) of a two-way road. **Rule 2**. The vehicle can travel on the right or left side of a one-way road if it is safe to do so. **Rule 3**. The vehicle must prioritise a traffic officer's instructions above the rules of the road and road signs. **Rule 4**. The vehicle must keep a following distance that is appropriate and prudent, taking into consideration the speed of the vehicle being followed, the amount of traffic, and the state of the road. **Rule 5**. Speed limit (in km per hour) of 60, 100, and 120 for when the vehicle is on urban area, outside urban area, and on freeway, respectively.

Rule 6. The vehicle should not cross over the solid driving marking (yellow or white in colour).

Rule 7. The vehicle should drive over to the left lane and not accelerate when it is being overtaken.

Rule 8. The vehicle should always signal its intentions in time before it executes it, and it should execute only when it is safe to do so.

Rule 9. The vehicle should not stop on the road, unless an accident had to be avoided, a traffic officer or road sign(s) has instructed, or it was caused by an unavoidable cause (such as a mechanical problems).

Rule 10. At a roundabout or mini-circle, the vehicle must give way to other vehicle that approach from the right (the other vehicle(s) should be already approaching from the right or it stopped on the yield sign first). The vehicle should know when to yield in other intersections as well (such as four-way, three-way, etc.).

Rule 11. The vehicle may not enter a traffic lane or cross it if it is likely to cause a dangerous situation or disrupt the flow of traffic.

Rule 12. The vehicle should not turn if it will obstruct or cause danger to other traffic. Therefore, before turning, the vehicle must move to the right lane, indicating necessary intentions and turn when it is safe to do so

Rule 13. The vehicle should never park on the side-walk or the verge, therefore, it should park within a designated parking space (see more parking rules in Figure 26).

	Rule 14. The vehicle should always give way to the emergency vehicles, rescue vehicles, traffic officers vehicles, etc., when they signal with the siren. **Rule 15**. The vehicle must stop for pedestrians on, or about to enter, a pedestrian crossing on its side of the road, or if it is involved in an accident. **Rule 16**. The vehicle must make use of hooters for safety reasons only, and the hooter must be audible enough for a distance of at least 90 meters. Furthermore, the tone of the pitch should not vary for any reason **Rule 17**. The vehicle must have headlights that are white, they should be switched on between sunset and sunrise, and they should be switched on if visibility is not clear in a distance of greater or equals to 150 meters. **Rule 18**. The vehicle may not drive in a way that endangers the lives of other drivers, pedestrians (the vehicle will always be liable if it hits a pedestrian regardless of who had the right to way in the road), or damage any property. **Rule 19**. The vehicle should ensure that the passenger(s) fasten the seatbelts before it starts moving. **Rule 20**. The vehicle should stop immediately after an accident and if someone died or got injured in the event then the vehicle should not move without a traffic officer's authorisation.

The traffic rules that are provided in Table 4 were carefully selected to assess the reliability of the AVs that have already been manufactured. However, these rules can also

Some no-stopping and no-parking distances

NO STOPPING IN, ON OR CLOSER		NO PARKING CLOSER THAN:		
9 m from	**6 m from**	**5 m from**	**1.5 m from**	**1 m from**
⏩ a pedestrian crossing	⏩ a tunnel ⏩ a subway ⏩ a bridge ⏩ where the road has been constricted	⏩ an intersection	⏩ a fire hydrant, on either side of it	⏩ the road edge outside an urban area, unless in a demarcated parking bay

Figure 26: Some of the parking rules the vehicle should always adhere to (adopted from Hoole (2013)).

be used for AVs that are not yet manufactured (whereby the assessment of these rules for non-manufactured vehicles can be done through simulation modelling, or prototypes—like miniature vehicles, etc.). With these rules in place, section 3.1.2.1 provides the analysis indicating which rule(s) were passed, failed, and/or not tested during the testing of the AVs.

Furthermore, looking at the quality of the AV's performance, firstly, the sensors of the vehicles should be chosen in such a way that optimal performance is achieved. Secondly, an appropriate and optimal fusion of sensors should be established. Thirdly, the data (provided by the sensors) should be interpreted correctly. Fourthly, the system should plan on what should be done on what the sensors have read in. Finally, the correct manoeuvre should be executed. With all this in place, it can be noted that all other stages or operations of the intelligence of AV depends on the sensors and the quality of data they provided. Therefore, sensors that were specified by Ignatious et al. (2022), Vargas et al. (2021), and Yeong et al. (2021) were used to select the best three for camera and LIDAR, and best two for Radio Detection and Ranging (RADAR) (Table 5, 6, and 7).

Table 5: The top three best performing LIDAR sensor in terms of vertical Field-of-View (FOV), horizontal FOV and Range.

LIDAR sensor	**vertical FOV (°)**	**horizontal FOV (°)**	**Range (m)**
Velodyne Alpha Prime	40	360	245
Velodyne VLP-32C	40	360	200
Velodyne RoboSense	40	360	200

Table 6: The top three best performing Camera sensor in terms of lens' baseline–it provides optimal view range, Range, and lenses resolution.

Camera sensor	**Baseline (mm)**	**Range (m)**	**Resolution (MP)**
Intel D15	55	10	3
RealSense D435	50	10	3
Framos D435e	55	0.2–10	2

Table 7: The top three best performing RADAR sensor in terms the overall frequency.

RADAR sensor	**Overall frequency (Giga-Hertz (GHz))**
Smartmicro UMRR-96 T-153	79 (usually in 77 to 81)
Continental ARS 408-21	76 to 77

Additionally, the fusion of sensors were analysed to identify the best fusion option of the LIDAR, camera, and RADAR sensors. A *sensor fusion* relates to sensors being fitted into one AV for optimal results in terms of performance (Vargas et al., 2021; Yeong et al., 2021). Yeong et al. (2021) and Vargas et al. (2021) already provided analysis of these sensors, however, in this dissertation the fusion of lesser sensors (fusion of only two sensors) was also analysed. This was necessary since these sensors are expensive (especially the LIDAR and RADAR). According to the Neuvition website, Velodyne 64-line LIDAR is $80,000 (≈R1.5 million). The Smartmicro RADAR sensor is £2,725.00 to £2,995.00 (≈ R55,712.92 to R61,233.10) according to Level Five Suppliers website. The Continental ARS 408 is between R729.06 and R13,155.87 according to AliExpress website. Finally, the Intel D415 costs $317.95 (≈ R5,750.39), RealSense D435 costs $317.50 (≈ R5,742.25), and Framos D435e costs €945.10 (≈ R16,861.39) according to Sparkfun, B & H Photo Video Audio, Mouser Electronics websites. It can be argued that the camera sensor is necessary since it is the only sensor with the highest image resolution and can see colour, the RADAR is also a necessary sensor as it provides the longest range detection. Therefore, the analysis of two sensor fusion (camera and RADAR) and three sensor fusion (camera, LIDAR and RADAR) was conducted as seen in Table 8 and 9. The goal was to check if the two sensor fusion would meet the minimum requirement of fusing all factors or features of each sensor so that they produce optimal results.

Considering Figure 17 and 18, the sensor fusion analysis was conducted. Sensor fusion aims to provide insight into which sensors can be exploited for automotive industries still new in developing AVs. To perform the analysis, the symbols seen in Figure 17 were redefined as follows.

$$\begin{aligned} \text{✗} &: \quad 0 \\ \sim &: \quad 0.5 \\ \checkmark &: \quad 1 \end{aligned}$$

Table 8: The comparison of AV sensor fusion based on the comparison provided by Yeong et al. (2021).

Factors	**Camera**	**LIDAR**	**RADAR**	**2-Fusion**	**3-Fusion**
Range	0.5	0.5	1	1.5	2
Resolution	1	0.5	0	1	1.5
Distance Accuracy	0.5	1	1	1.5	2.5
Velocity	0.5	0	1	1.5	1.5
Colour Perception (traffic lights etc.)	1	0	0	1	1
Object Detection	0.5	1	1	1.5	1.5
Object Classification	1	0.5	0	1	1.5
Lane Detection	1	0	0	1	1
Object Edge Detection	1	1	0	1	2
Illumination Conditions	0	1	1	1	2
Weather Conditions	0	0.5	1	1	1.5
Total				**13**	**19**
Good Fusion (≥ 11)				**Yes**	**Yes**

It can be noted that the last row in Table 8 assessed if the fusion of two or three sensors is good or not. A criteria of ≥ 11 was used on the ground that there are 11 factors that were assessed *and* all values in blocks representing *2-Fusion* and *3-Fusion* are ≥ 1. Furthermore, to further conduct the analysis of the comparison seen in Figure 18, the keywords (such as *Good*, *Yes*, etc.) were redefined as follows.

$$\begin{aligned} \text{Poor, Yes:} &\quad 0 \\ \text{Average:} &\quad 0.25 \\ \text{Good, 200m:} &\quad 0.5 \\ \text{Very good, No, 250m:} &\quad 1 \end{aligned}$$

The criteria used in the last row in Table 9 is the same as the one used for Table 8. However, a value of ≥ 6 was used instead of ≥ 7 since there are seven rows, this is due to fact that the third factor tested consisted of zeros. Therefore, both sensor fusions will always result in a zero (sensor fusion is always poor).

Table 9: The comparison of AV sensor fusion based of the comparison provided by Vargas et al. (2021).

Factors	**Camera**	**LIDAR**	**RADAR**	**2-Fusion**	**3-Fusion**
Range	0.5	0.5	1	1.5	2
Resolution	1	0.5	0.25	1.25	1.75
Affected by weather conditions	0	0	0	0	0
Affected by lighting conditions	0	1	1	1	2
Detects speed	0	0.5	1	1	1.5
Detects distance	0	0.5	1	1	1.5
Interference susceptibility	1	0.5	0	1	1.5
Total				**6.75**	**10.25**
Good Fusion (≥ 6)				**Yes**	**Yes**

Looking at the results in Tables 8 and 9, automotive industries in Africa should opt for two sensor fusions. One LIDAR sensor is more expensive compared to RADAR and camera sensors combined. Therefore, considering a two-sensor fusion would be more beneficial as a start.

3.1.1.4 Physical analysis

This phase focuses on the *physical definition* of the needs and requirements analysis as defined in section 1.4. When considering the nature of the AVs, more specifically on what makes it autonomous, the primary aspect is the components that gathers data—the sensors. Furthermore, when we consider the AV architectures or structural designs in Figures 12 and 35—the only physical components (that are directly linked to automation and the intelligence of the AVs) are the sensors, the rest of the structures are the softwares that trigger controls.

Consequently, the analysis of sensors provided in the *Performance and critical analysis* phase was utilised, and a recommendation of an efficient AV structural and functional design was developed (Figure 27).

3.1.1.5 Needs validation

In this phase, typically the aim is to assess the fundamental validity of the argument put forth regarding the existence of a need for a new system as well as the viability of

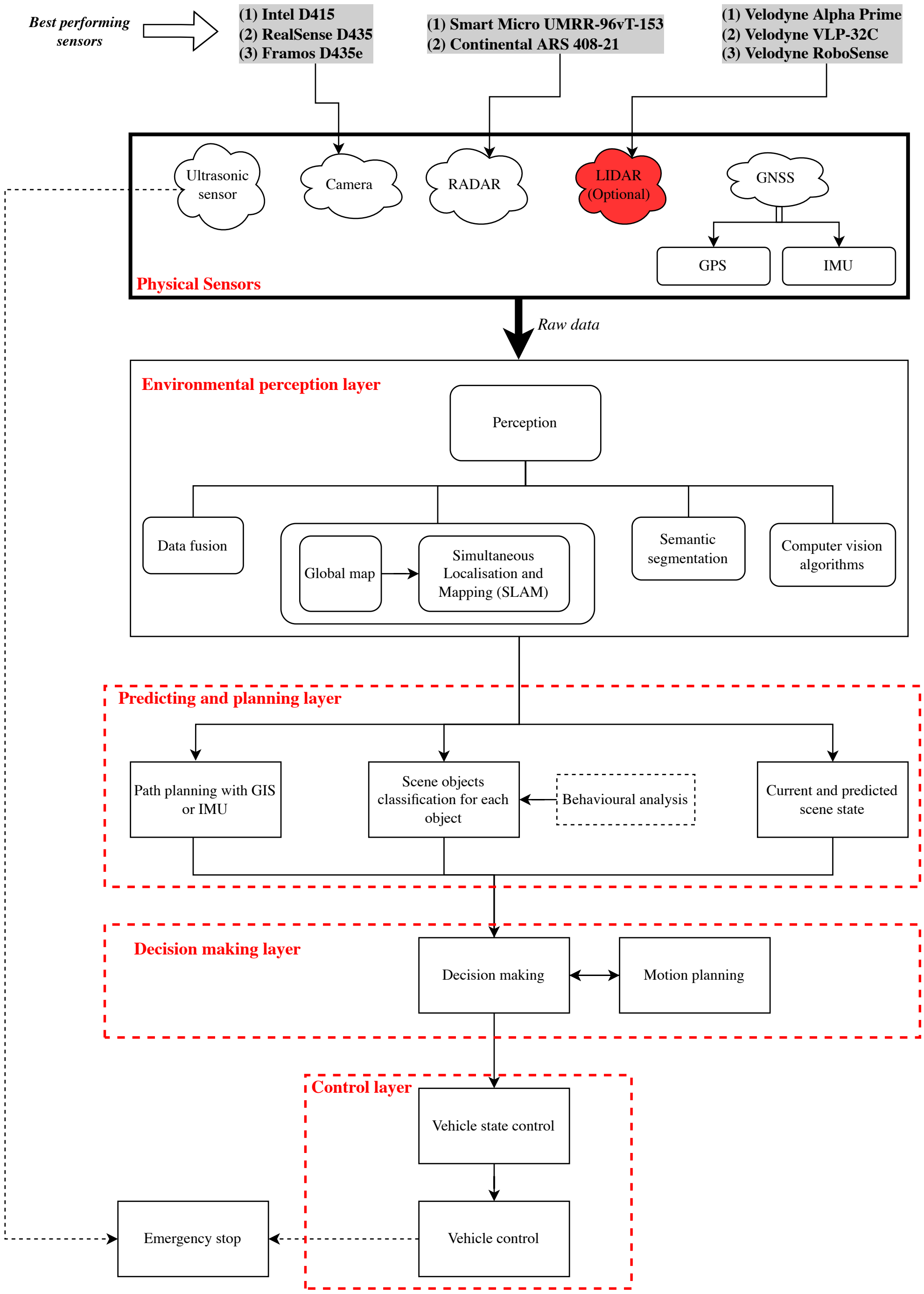


Figure 27: The recommended AV structural and functional design ((re)designed from Sviatov et al. (2021)'s design).

meeting this need at a cost that is both feasible and risk acceptable (Kossiakoff et al., 2011). In consideration of this dissertation, this phase focused on validating the reliability of AVs obeying traffic rules. The details associated with this phase can be seen in section 3.1.2.

Considering Figure 27 and the results in Tables 8 and 9, automotive industries in Africa should opt for two sensor fusions. One LIDAR sensor is more expensive compared to RADAR and camera sensors in the development of the AV. Hence LIDAR is highlighted in red and marked as optional in Figure 27. Therefore, considering a two-sensor fusion would be more beneficial as a start.

3.1.2 Reliability design data and analysis

This section focused on providing details regarding the Needs validation by providing the reliability analysis of Autonomous Vehicle (AV)s. The *Reliability Engineering* and *statistics* theories were utilised to achieve this objective (see section 3.1.2.1). Furthermore, the theory of Artificial Intelligence (AI) was also a major contribution since it was necessary to understand how *machine learning* works when the AVs were analysed in section 3.1.2.1.

3.1.2.1 Reliability design data

The purpose of this section is to provide the *reliability of AV concerning traffic rules* by conducting a reliability analysis using *Reliability Engineering* and *statistics* theories. Furthermore, this analysis addresses the *complexity* of the AVs to obey road rules and swift decision-making when necessary. Two main analyses were conducted to provide the reliability of AV. Firstly, videos of AVs were analysed—these videos showed different AVs of different brands and companies. In addition, these videos showed how the AVs are currently performing, and the video links can be seen in Table 10.

The purpose of using videos to conduct the assessment is that there is no raw data on the performance of the AV available in the published literature. In addition, Africa does not have AV automotive industry yet, and there was no access to an AV to do the tests. However, some people already did the tests and posted the videos online on YouTube (accessible for free). Therefore, it was more efficient to access those videos and assess whether the AVs obeyed traffic rules. Furthermore, it can be noted in the videos provided that the reviewers focused on how the AVS is performing without hiding the mistakes of the vehicle. Therefore, this provided good quality data for a reliable analysis of AV performance.

Table 10: Video links related to AV

Vehicle brand	Video link
Tesla models	Video Test 1
	Video Test 2
	Video Test 3
	Video Test 4
	Video Test 5
	Video Test 6
	Video Test 7
Deeproute	Video Test 1
Cruise	Video Test 1
Waymo	Video Test 1
	Video Test 2
	Video Test 3
	Video Test 4
AutoX	Video Test 1
	Video Test 2
	Video Test 3
	Video Test 4
Pony AI	Video Test 1
	Video Test 2
	Video Test 3
	Video Test 4
Yandex	Video Test 1
	Video Test 2
	Video Test 3

The video links provided in Table 10 can be accessed with a simple click. These videos were treated as tests from time 0 (t_0) to time n (t_n) and observed which rule was tested (further assessed if the rule was passed or failed) and if the rule was not tested. If the rule was not tested, it was given a value of zero since there's a 50% chance that the rule will pass. All rules are supposed to be 100% passed, meaning, in a given period, a particular rule should be passed in any situation the AV encounters. Two tables were created relating to the outcome of accessing the videos in Table 10 (see Chapter 4).

The reliability of AVs concerning the traffic-rules formula was estimated using a reliability engineering plotting method. To provide the estimated formula, a mathematical reliability model was formulated using *Weibull distribution* and its application. However, before achieving this goal, the *time-stamps* of when the traffic rules were disobeyed from the analysed videos in Table 10 were recorded—these time-stamps represent *time-to-failure* of AVs concerning the traffic-rules. This was done so that further analysis would be conducted; therefore, the *time-to-failure* data was generated (see Table 11). The analysis of *time-to-failure* data can be seen in section 3.1.2.2.

3.1.2.2 Time-to-failure data analysis

This section provides the *time-to-failure* data analysis related to the AVs as they fail to obey some of the traffic rules. To conduct the analysis, some assumptions had to be put in place, and these assumption are outlined as follows.

1. All AVs of different brands are considered to be just one AV society. This is because this dissertation focused on providing the reliability of AVs jointly. Therefore, if one of the AV brands fails one traffic rule, all AV brands are affected.

2. The AVs are tested for 20 minutes, though some videos are shorter than 20 minutes, and some are barely longer than 20 minutes.

3. The tests conducted in each AV brand represent one of the tests of AV as a whole—that is, video one represents test one, video two represents test two, and so on, disregarding the AV's brand.

To analyse the data, two programming environments or platforms and languages were utilised, which are *RStudio* platform using with *R programming language*, and *Anaconda* platform using *Python programming language*. These are freely available platforms to download, write, and run codes.

To formulate the mathematical reliability model, it is necessary to know which statistical distribution the *time-to-failure* data *closely* follow. The word *closely* emphasises that it can be challenging to tell which distribution a certain data *accurately* follows. Furthermore, uncertainty is associated with how the AVs operate, especially in difficult conditions. Therefore, the way they fail is associated with uncertainty; hence, the probability of the AVs failing at a certain time-stamp should be calculated —this serves as an estimate of how reliable these AVs in public. The following lines of code (R language) were used to check the statistical distribution of the *time-to-failure* dataset.

Table 11: Time-to-failure of AVs observed in the analysed videos

Number of observation (n)	**Time-to-failure (t_i, in minutes)**
1	0.19
2	0.32
3	0.32
4	0.40
5	1.04
6	1.14
7	1.50
8	1.54
9	2.08
10	2.10
11	2.13
12	2.15
13	2.21
14	2.29
15	2.46
16	2.52
17	2.56
18	2.59
19	3.14
20	3.37
21	3.47
22	4.41
23	5.03
24	5.29
25	5.47
26	5.57
27	6.24
28	6.55
29	7.22
30	7.39
31	11.04
32	13.37
33	17.07

```
time_to_failure <- read.csv(file = "TimeToFailure_Data.csv")
y <- time_to_failure$TimeToFailure_Minutes
```

```
time_to_failure_hist <- hist(y, breaks = 20, col = gray(0.001), main = "",
    xlab = "Time to failure (minutes)", xlim=range(0, 20), ylim = range(0,
    15))

abline(v = mean(y), col = "red", lwd = 3) # Add line for mean
text(x = mean(y) * 1.6, y = mean(y) * 3, paste("Mean =", round(lambda,
    digits = 3)), col = "red", cex = 1) # Add text for mean
```

After running the above code, the distribution produced can be seen in Figure 28. A *poisson distribution* can be fitted with a lambda (λ) value of at least one traffic-rule failure by an AV every 4.066 minutes. The Poisson distribution can be fitted as there is close relation in terms of how it appears and the characteristics of the Poisson distribution. The Poisson distribution can be seen in Figure 29, and it can be noted how the shape changes as the λ value changes. The AV time-to-failure distribution seems to have some bins that are far from each other compared to the ones in Figure 29. This is because the number of observations relating to the time at which AVs failed traffic rules is small ($n = 33$). Though $n = 33$ is statistically acceptable since $n \geq 30$, they are good enough to assess the current reliability of AVs with respect to the traffic rules.

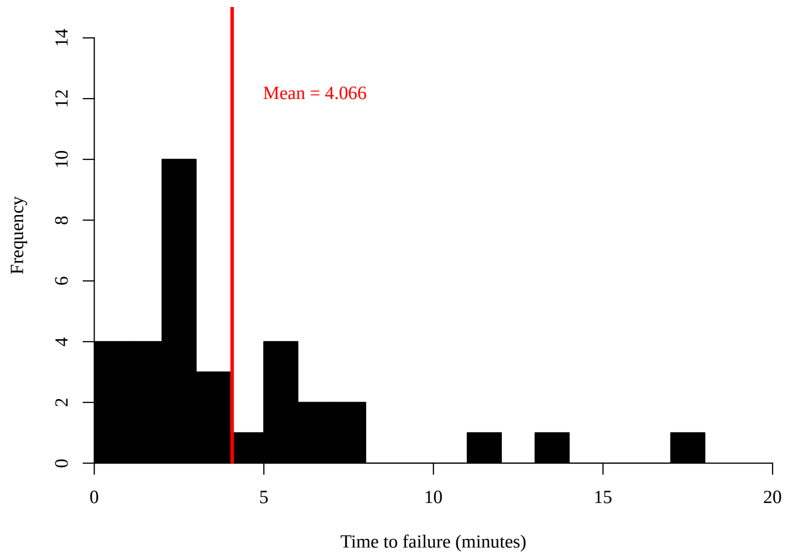


Figure 28: Time-to-failure distribution

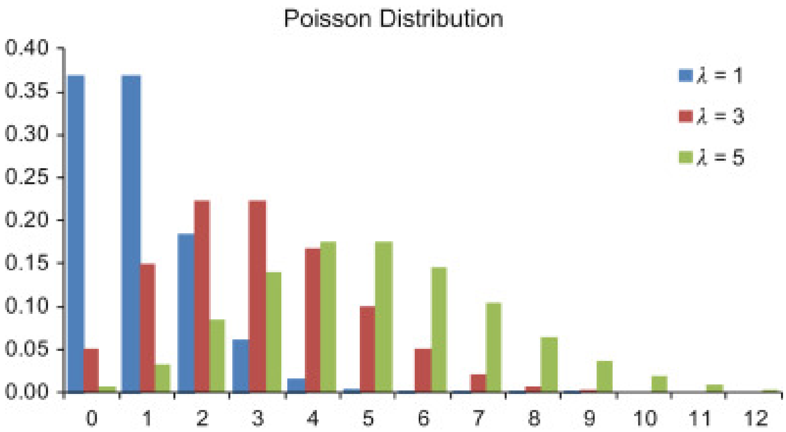


Figure 29: Poisson distribution (adopted from Kissell and Poserina (2017))

When comparing Figure 28 and 29, it can be seen that the AV time-to-failure distribution is closely related to the orange poisson distribution (with $\lambda = 3$). However, it can be noted that the time-to-failure distribution is also closely related to a *right skewed normal distribution* (see Figure 30).

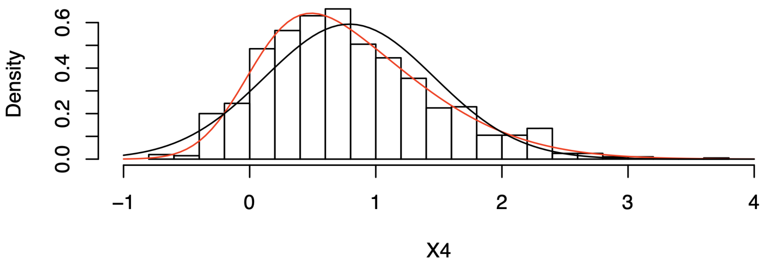


Figure 30: Right skewed normal distribution (adopted from Figueiredo and Gomes (2013))

Furthermore, we can observe how the probability density function (red fitted line) of the right skewed normal distribution closely look like the *Weibull distribution* (see Figure 31).

These distributions look similar, but it is a must to select a distribution that will help construct a model that gives a good representation of how the AVs are reliable in traffic rules. With this in mind, it is necessary to shift focus to look at which distribution(s) mostly deals with time-to-failure data to provide an estimate of a system or component's reliability.

Since the data used to produce the distribution in Figure 28 is directly linked to time-to-failure of AVs concerning traffic rules, it is more appropriate to utilise and fit

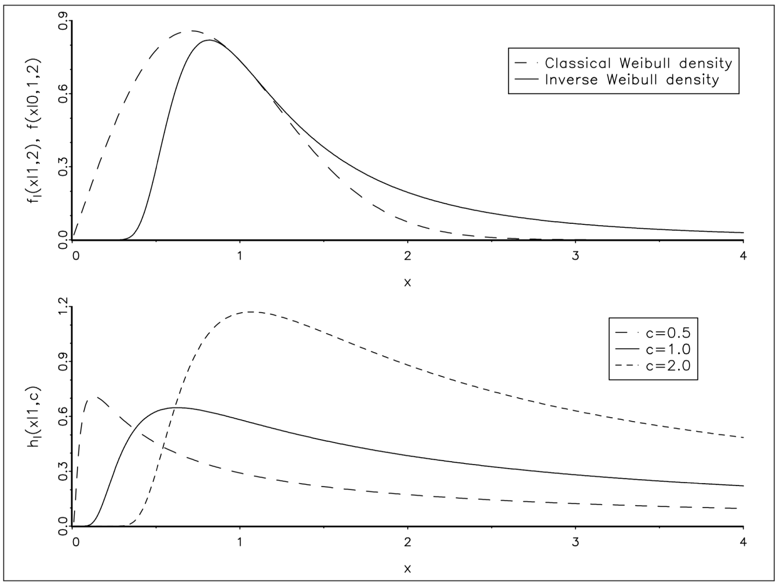


Figure 31: Probability density function of Weibull distribution (adopted from Rinne (2008))

the *Weibull distribution*. This is because it is one of the most well-known lifetime distributions and accurately describes observed failures of a wide range of components and occurrences (Lai et al., 2006). Furthermore, the Weibull distribution can adapt to many different conditions and take various distributions as its parameters change; hence it is fit for the type of distribution seen in Figure 28. This dissertation focused on the occurrence (which is the probability) at which the AV will fail to adhere to one of the traffic rules. Consequently, a model was formulated using the *Weibull distribution's application* (a *reliability engineering* method); this can be seen in section 3.1.2.3.

3.1.2.3 Reliability analysis model

The Weibull distribution has been found to appear in five different forms, *three-parameter* and *two-parameter* being the two common forms (Hallinan Jr, 1993; Lai et al., 2006). The three-parameter has the τ, β, and α (or η) parameters. When the $\tau = 0$, the Weibull distribution is a two-parameter.

The three-parameter Weibull distribution was chosen for this dissertation since the parameter τ was found to be useful as the vehicle cannot fail one of the traffic rules at zero minutes (i.e., $t_0 \neq 0$, this means the graph of the time-to-failure's values do not start from zero or the origin). τ is known as a *location* or a *threshold* parameter; in this case, the *threshold* applies. The reliability model was created using the probability plotting approach as it is the least mathematically demanding method for parameter estimation (it is easy to understand). The following are three simple ideas involved in conducting the probability plotting approach.

- A visual representation of the data is produced on a specialised probability plotting paper (different for each statistical distribution).
- Utilise a probability plotting paper with transformed axes to ensure that a genuine Cumulative Density Function (CDF) plots as a straight line (linearisation).
- The data is deemed to suit the appropriate distribution if a straight line can fit the plotted data —this can be interpreted as an *assumption*.

To implement the method, the failure times' data should be obtained so that it can be analysed and linearised by calculating the *median ranks* of the data, and this is one of the methods. The median rank is the cumulative percentage of a population of a given data sample with a 50% confidence level. To calculate the median ranks, Bernard's approximation was utilised. The second method uses the median ranks and the actual failure times without being linearised. The two methods are now discussed. (Firdos et al., 2020; Lai et al., 2006)

Method 1: The linearised approach

$$\text{Median rank } (r_{ti}) = 100\left(\frac{i_{ti} - 0.3}{n + 0.4}\right) \tag{3.1}$$

Where:

$$i_{ti} = i_{ti-1} + Nti \tag{3.2}$$

$$N_{ti} = \frac{(n+1) - i_{ti}}{1 + (n - \text{number of preceding items})} \tag{3.3}$$

i = order number of failed items, $0 > i \leq n$
n = sample size
It should be noted that $i = ti$ when conducting the calculations.

The values produced by r_{ti} (see Equation 3.1) are in %, which are further used to calculate the y-axis of the Weibull probability plot. The x- and y-axis of the Weibull probability plot are formulated in Equation 3.7 and 3.8. Before that, Equation 3.4 is the reliability function for a two-parameter Weibull distribution, which was manipulated to derive a linear function (Equation 3.6).

$$F(t) = Q(t) = 1 - e^{-\left(\frac{t}{\eta}\right)^{\beta}} \tag{3.4}$$

To linearise Equation 3.6, the double natural logarithms were taken to produce the following equation.

$$ln\left(ln\left(\frac{1}{1-Q(t)}\right)\right) = \beta ln(t) - \beta ln(\eta), \tag{3.5}$$

but, $y = ln\left(ln\left(\frac{1}{1-Q(t_i)}\right)\right)$ and $x = ln(t)$, therefore,

$$y = \beta x - \beta ln(\eta). \tag{3.6}$$

When conducting calculation for every time-to-failure observed (t_1 to t_n), the corresponding x_i and y_i are calculated as follows.

$$x_i = ln(t_i) \tag{3.7}$$

and

$$y_i = ln(-ln(1 - r_{ti})) \tag{3.8}$$

Given Equation 3.7 and 3.8, one may use the specialised Weibull probability plotting graph or sheet (this can be achieved by manually plotting or using computer software tools such as Microsoft Excel, RStudio, Python, etc.). It should be noted that Equation 3.8 was derived from a two-parameter Weibull distribution by manipulating Equation 3.4.

The problem is that Equation 3.8 utilised the two-parameter distribution function, but the three-parameter Weibull distribution is the desired approach. Therefore, the correct equation should be used, which is similar to Equation 3.4 with one additional parameter, the threshold (τ), see Equation 3.9 and 3.10.

$$F(t) = Q(t) = 1 - e^{-\left(\frac{t-\tau}{\eta}\right)^{\beta}},\ t > \tau \tag{3.9}$$

and given that $\lambda = \eta^{-\beta}$, the simplified equation is

$$F(t) = 1 - e^{-\lambda(t-\tau)^{\beta}},\ t > \tau. \tag{3.10}$$

To linearise Equation 3.10, the double natural logarithms were taken to produce the following equation (Equation 3.11).

$$y = \beta ln(\lambda) + \beta ln(t - \tau), \tag{3.11}$$

where

$$\lambda = \frac{1}{mean}. \tag{3.12}$$

The value of τ is the value that cuts the x-axis after plotting the calculated linearised values. The value of η is the x-axis value that cuts through the plotted graph when plotted with the value $ln(-ln(1 - 0.6320))$. The value of τ can be seen as t_0. The value of β is the slope of the fitted straight line. The value of λ in a Weibull distribution is interpreted as the *failure rate* and calculated as seen in Equation 3.12. Figure 28 shows that $mean = 4.066$ and the $\lambda = 1/4.066$. The question is, *what is the confidence level that the mean value is 4.066 and can be utilised in further calculations?* To address this question, a *Bootstrap* approach was conducted to, firstly, re-calculate the mean value, and secondly, the lower and higher values that the mean falls within were calculated, with 95% level of confidence.

A *bootstrap* method is when one generates a large number of *phantom samples* known as *bootstrap samples* by *re-sampling* (with replacement) from the sample data at hand. The sample summaries for each bootstrap sample are then calculated (usually a few thousand or thousands). Finally, the *bootstrap distribution* of the statistic refers to a histogram of the collection of these computed values. There are several ways to conduct a bootstrap method. At times it depends on what needs to be corrected or validated in the sample, such as the mean values, standard errors, rate parameters, distributions, etc.

For this dissertation, the time-to-failure distribution and the λ value had to be verified for three reasons. First, the data sample has few observed sample numbers ($n = 33$). Second, to conduct reliability analysis, a distribution has to be known, and the resulting formula depends on the λ parameter. Lastly, the distribution looked like a few other statistical distributions (as discussed earlier in this chapter). The bootstrap was conducted in *RStudio* and *Anaconda* software applications using *R* and *Python* programming languages, respectively.

To verify the time-to-failure distribution and the mean value, firstly, in *Python* environment, a function called `bootstrap()` was adopted to calculate where the mean value falls in with a 95% confidence interval. Secondly, in *R*, two approaches were utilised; (1) functions `matrix()` and `sample()` were used simultaneously to re-sample m number of times and store all the re-sampled data into a matrix—m is a discrete number that defines how many times one would re-sample from the same population ($m > 0$); (2) a function called `boot()` was used, its format is outlined as follows.

```
boot(data, statistic, R, sim = "ordinary", stype = c("i", "f", "w"),
  strata = rep(1,n), L = NULL, m = 0, weights = NULL,
  ran.gen = function(d, p) d, mle = NULL, simple = FALSE, ...,
  parallel = c("no", "multicore", "snow"),
  ncpus = getOption("boot.ncpus", 1L), cl = NULL)
```

Several arguments need to be provided to utilise `boot()` function. These arguments are well explained within *RStudio* software. Running the `?boot` on the console provides all the details needed.

Method 2: Auto-linearisation method

In this method, the first few steps (of calculating the median ranks r_i) described in the previous method apply. However, the values of time-to-failure (t_i) are used to plot. However, most steps should be computed using software to generate linearised values of β, τ, and η, since these values are supposed to be utilised in a linear equation (Equation 3.11). In this dissertation, this method was adopted, and RStudio software was used (using an R programming language). The following formulae were used to plot and fit the three-parameter Weibull linear probability graph. The formulae take the time-to-failure values (t_i), calculate the median ranks internally, then plot and fit the linearised data (the results can be seen in Chapter 4). It is important to manually calculate the median ranks to verify against the values calculated by the function.

```
wblr(x, s=NULL, interval=NULL,...)
wblr.fit(x, modify.by.t0=FALSE, ...)
```

With the model formulated and discussed, the results generated in different software platforms are all outlined and discussed in Chapter 4.

3.2 Chapter summary

In this chapter, two research methodologies (the *Design Science Research (DSR)* and *Systems Thinking*) were outlined, modified, and discussed. The methods were unified to meet the objectives of this dissertation. The research approach was conducted using the *systems engineering* method, specifically the *needs analysis* method. The five functional capabilities of AVs were outlined, and 11 functional requirements are outlined in Table 3. Furthermore, to analyse the AV, 30 traffic rules were carefully selected from a K53-book, and they were used to analyse the videos in Table 10.

Sensors were analysed by conducting a sensor fusion in Tables 8 and 9. In addition, a structural and functional design was re-designed based on sensor-fusion results (see Figure 27.

The reliability design was then developed by firstly providing the time-to-failure data (in Table 11) and analysing the data. The data analysis selected a *three-parameter Weibull* distribution. Consequently, two (linearised method and non-linearised method—a computerised method) analysis methods were outlined and explained.

Chapter 4

Results and discussion

The purpose of this chapter was to provide detailed results that were produced using all the techniques discussed in section 3.1.2. Furthermore, a comprehensive discussion of the results was provided. The results and related discussions are now looked into.

Table 12: The reliability analysis of Autonomous Vehicle (AV) with respect to traffic rules—part A (road signs, signals, and markings rules)

Vehicle brand	**Road signs, signals, and markings traffic rule**						
	Tesla Model 3	**AutoX (robo-taxi)**	**Waymo (by Google)**	**Deeproute**	**Yandex**	**Pony AI**	**Cruise**
Rule 1	1	1	1	1	1	1	1
Rule 2	1	1	1	1	1	1	1
Rule 3	0	0	0	0	0	0	0
Rule 4	0	0	0	0	0	0	0
Rule 5	1	1	1	1	1	1	1
Rule 6	1	1	1	1	1	1	1
Rule 7	0	0	0	0	0	0	0
Rule 8	0	1	1	1	1	1	1
Rule 9	0	1	1	1	1	1	1
Rule 10	0	0	0	0	0	0	0
Passed	**4**	**6**	**6**	**6**	**5**	**6**	**6**

In consideration of the videos in Table 10, two tables were drawn to provide the results of the analysis; see Table 12 and Table 13. These two tables represent the assessment outputs of AVs obeying the traffic rules. The values (red zeros, black zeros, and green ones) in Table 12 are described as follows (which applies to Table 13 related to the rules of the road).

1: Rule tested and *passed*.
0: Rule tested and *failed*.
0: Rule *not tested*.

Table 13: The reliability analysis of AV with respect to traffic rules—part B (rules of the road)

Vehicle brand	**Road signs, signals, and markings traffic rule**						
	Tesla Model 3	**AutoX (robo-taxi)**	**Waymo (by Google)**	**Deeproute**	**Yandex**	**Pony AI**	**Cruise**
Rule 1	1	1	1	1	1	1	1
Rule 2	1	1	1	1	1	1	1
Rule 3	0	0	0	0	0	0	0
Rule 4	1	1	1	0	1	1	1
Rule 5	0	1	1	0	0	0	0
Rule 6	0	1	1	1	0	1	1
Rule 7	0	0	0	1	0	1	1
Rule 8	1	1	1	0	1	1	1
Rule 9	0	1	1	1	0	1	1
Rule 10	0	1	1	1	0	1	1
Rule 11	0	1	1	1	1	1	1
Rule 12	0	1	1	1	1	1	1
Rule 13	1	0	0	0	0	0	0
Rule 14	0	0	1	0	0	0	0
Rule 15	0	1	0	1	0	1	1
Rule 16	0	0	0	0	0	0	0
Rule 17	0	0	0	0	0	0	0
Rule 18	0	1	0	1	0	1	1
Rule 19	1	0	0	0	0	0	0
Rule 20	0	0	0	0	0	0	0
Passed	**6**	**12**	**11**	**10**	**6**	**12**	**12**

The totals passed number of rules (combining the totals in Table 12 and 13) for each vehicle brand are illustrated in Table 14.

Table 14: Total passed traffic rules by different AV

Vehicle brand	Tesla Model 3	AutoX (robotaxi)	Waymo (by Google)	Deeproute	Yandex	Pony AI	Cruise
Total passed	10	18	17	16	11	18	18
%	0.3333	0.6000	0.5667	0.5333	0.3667	0.6000	0.6000

It can be noted that the least performing vehicle is the Tesla Model 3 (passing the rules 33% of the time). Furthermore, this vehicle cannot be categorised as a level 4 or 5 AV, its autonomous level is less or equal to three. The highest performing vehicles are AutoX (robotaxi), Pony AI, and Cruise (passing the rules at least 60% of the time).Therefore, a more in-depth analysis had to be conducted using the reliability analysis—the evaluation of the reliability of AV with respect to traffic-rules. This analysis considered only the vehicles that have autonomous level of 4 or 5 (excludes the Tesla Model 3).

Therefore, considering the time-to-failure reliability analysis method selected—the Weibull distribution and application, the linearised method was followed (automated with functions) and the results are now discussed.

The first step in applying the linearised Weibull distribution is to calculate the median rank. Though the weibull function used in R programming was able to calculate the ranks from the data provided to it, it was necessary to calculate it manually to verify. Therefore, the following manually created function in Anaconda environment (using Python language) was used to calculated the median ranks of the data provided in Table 11.

```
def median_rank( data ):
  median_rank_results = []
  n = len( data )
  i_prev = 0

  for i in range (0 , len( data ) , 1):
    N = ( ( n +1) - i_prev) / (1 + (n - i) )
    i_prev = i_prev + N
    r = round ( ((i_prev - 0.3) / (n + 0.4)) *100 , 2 )
    median_rank_results.append( r )

  return median_rank_results
```

This function utilises Equation 3.1 equivalent to `median_rank_results`, Equation 3.2 equivalent to `i_prev`, and Equation 3.3 equivalent to `N` in the **for** loop. To call the function defined above in order to view the results, the following code was created and ran. The results are in Table 15

```
import numpy as np
data = [0.19, 0.32, 0.32, 0.40, 1.04,1.14, 1.50, 1.54, 2.08, 2.10, 2.13,
   2.15, 2.21, 2.29, 2.46, 2.52, 2.56, 2.59, 3.14, 3.37, 3.47, 4.41,
   5.03,5.29, 5.47, 5.57, 6.24, 6.55, 7.22, 7.39, 11.04, 13.37, 17.07]

median_rank_results = median_rank(data)
print("Median ranks:\n", median_rank_results,"\n")
```

The produced data in Table 15 can be used to further plot a linearised Weibull graph by calculating the x- and y-axis values using Equation 3.7 and 3.8. It should be noted that for a three-parameter distribution x_i is $ln(t_i - \tau)$. After the plot is complete, the values of β and η will be read to formulate the linear equation in Equation 3.11. However, as discussed in section 3.1.2.1, the method followed to create the graph is the *auto-linearised* approach whereby an R language function was utilised, programmed as follows.

```
 df <- data.frame(time = failure_rate_times, event = rep(1, 33))
 weibl <- wblr(df , col="Red", label="Censored dataset", dist = "weibull3p",
          ylab = "Unreliability")

# Plot the data with the fitted line.
weibl_fit <- wblr.fit(weibl, col = "Blue", method.fit = "rr")
plot(weibl_fit) # Plot the data
```

The data-frame created from utilising `failure_rate_time` holds the values of time-to-failure which are in Table 11, and the function `wblr()` used that data frame with one most important parameter called *dist*. In this case because the three-parameter distribution was used, `dist` was assigned `weibull3p`, this signifies that the three-parameter weibull distribution was fitted. If a two-parameter weibull was to be fitted, then `dist` would be `weibull2p`. The sub-function (this type of function depends on the upper level function, in this case its `weibl()`) called `wblr.fit()` was used to further fit the *rank regression* which appears as `rr` assigned to the parameter called `method.fit`. This allows the function to calculate the the rank regression and fit it to the linearised plot as a straight line. The graph produced can be seen in Figure 32.

Table 15: Time-to-failure median ranks

Rank number (n)	**Median ranks (r_{ti}, in %)**
1	2.10
2	5.09
3	8.08
4	11.08
5	14.07
6	17.07
7	20.06
8	23.05
9	26.05
10	29.04
11	32.04
12	35.03
13	38.02
14	41.02
15	44.01
16	47.01
17	50.00
18	52.99
19	55.99
20	58.98
21	61.98
22	64.97
23	67.96
24	70.96
25	73.95
26	76.95
27	79.94
28	82.93
29	85.93
30	88.92
31	91.92
32	94.91
33	97.90

From Figure 32 it can be noted that the section with the title *Censored dataset* provides the most important details of the graph—the three parameters of the three-

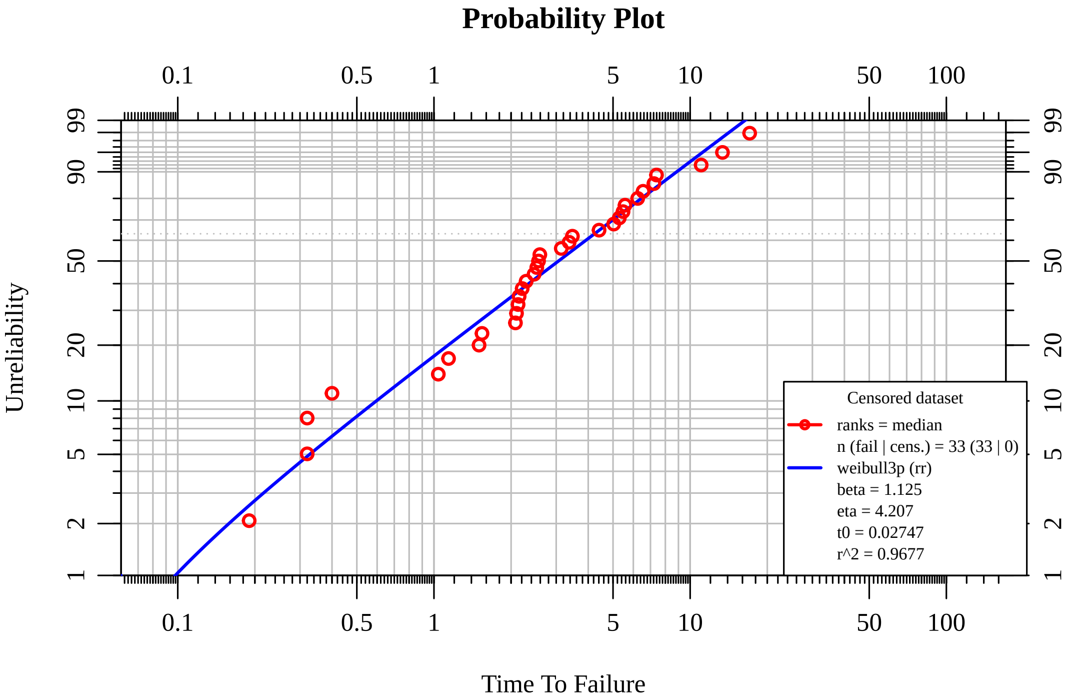


Figure 32: The linearised fitted Weibull distribution plot

parameter Weibull distribution (depicted in Table 16). Further, it can be noted that the censored data confirms that the rank utilised are the median ranks.

Table 16: The calculated three parameters of the three-parameter Weibull distribution from the AV time-to-failure data

Shape parameter (β, minutes)	Scale parameter (η, minutes)	Location parameter ($t_0 = \tau$, minutes)
1.155	4.24	0.9675

These parameters are plugged into Equation 3.9, however the simplified version of it (Equation 3.10 does not necessarily need η as there is the λ value in it. The mean value of the time-to-failure data can be seen in Figure 28—$mean = 4.066$ minutes. However, to decide whether or not to utilise that mean value it was necessary to recalculate it after bootstrapping and then get the confidence level interval for the mean value after bootstrapping. Bootstrap was prioritised to make sure after resampling m number of times ($m \geq 1$) from the same population (time-to-failure) if the value of the mean is still close the the initially calculated value. The recalculated mean value from the bootstrap method provides a more accurate estimate when thousands of resampling are conducted, therefore, the bootstrap method code was created in RStudio platform (using R programming language) which is outlined in Listing 4.1.

Listing 4.1: Time-to-failure bootstrapping code

```
y <- time_to_failure$TimeToFailure_Minutes

set.seed(1) # set seed (default value) to reproduce the results
Sample <- function(data, i) {
  sample(data[i], size = n, replace = TRUE)
}
results <- boot(y, Sample, R=10000) # Performing 10 000 replications
    with boot

# Plotting the output
hist(results$t, breaks = 20, col = gray(0.001), main = "",
  xlab = "Time to failure (minutes)",
  xlim=range(0, 20), ylim = range(0, 120000))

abline(v = mean(results$t), col = "red", lwd = 3) # Add line for mean
text(x = mean(y)*1.7, y = mean(y)*25000,
  paste("Mean =", round(mean(results$t), digits = 3)),
  col = "red", cex = 1) # Add text for mean

boot_lambda_value <- mean(results$t) # Lambda value after bootstrapping
```

Regarding the code above, the time-to-failure data was assigned to variable y, and *fixed seed* value was set to its default value of one to be able to reproduce the results when the bootstrapping is performed. The bootstrap was performed using the function `boot()`, and the number of resampling (m) was set to be 10,000. The distribution of the bootstrapped values of time-to-failure was plotted (see Figure 33) and the mean value can be seen in the plot, i.e., $mean_{new} = 4.058$ minutes.

This value ($mean_{new} = 4.058$) is close enough to the initial value calculated (mean = 4.066). Therefore, a 95% confidence interval of the mean value was calculated. The calculation was done in Anaconda platform (using Python programming language) and Listing 4.2 represents the created code of the 95% confidence interval values.

Listing 4.2: Time-to-failure bootstrapping code

```
import numpy as np
```

```
from scipy.stats import bootstrap

# time-to-failure data
data = [0.19, 0.32, 0.32, 0.40, 1.04, 1.14, 1.50, 1.54, 2.08, 2.10, 2.13,
      2.15, 2.21, 2.29, 2.46, 2.52, 2.56, 2.59, 3.14, 3.37, 3.47, 4.41,
      5.03, 5.29, 5.47, 5.57, 6.24, 6.55, 7.22, 7.39, 11.04, 13.37, 17.07]

#convert array to sequence
data = (data,)

#calculate 95% bootstrapped confidence interval for median
bootstrap_ci = bootstrap(data, np.mean, confidence_level=0.95,
    random_state=1, method='percentile')

#assign the 95% boostrapped confidence interval
conf_interval = bootstrap_ci.confidence_interval
```

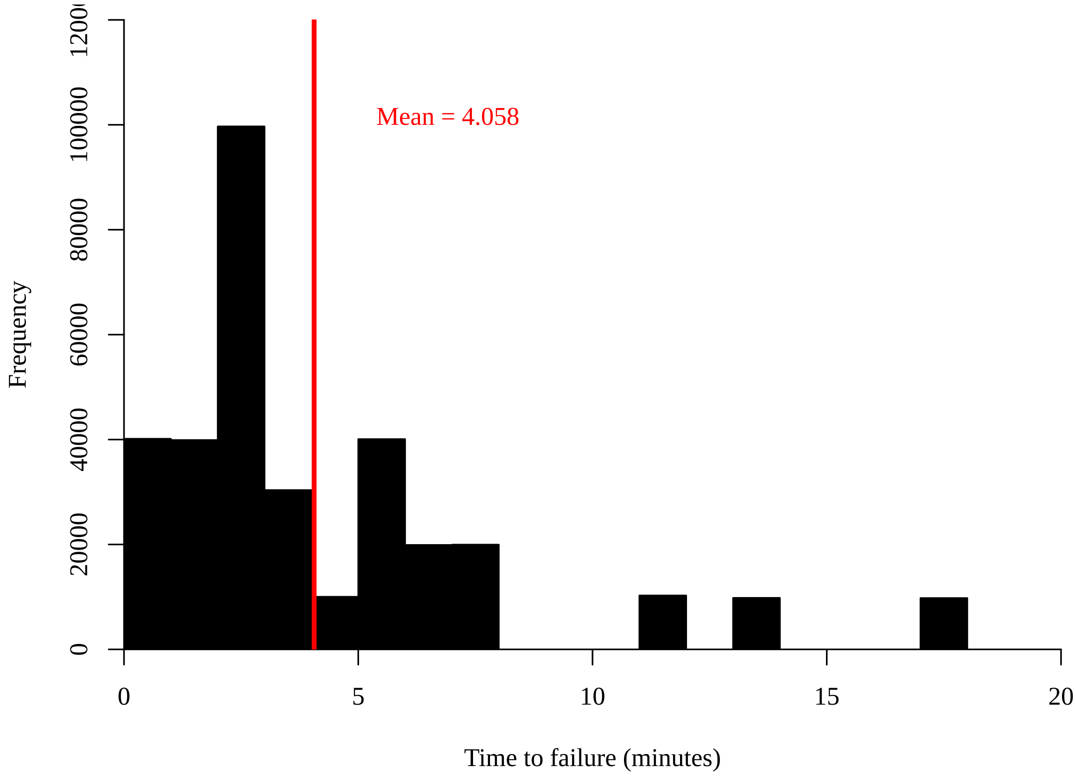


Figure 33: The time-to-failure distribution after bootstrapping by resampling 10,000 time

The most important parts of the code in Listing 4.2 are the assignments of the arguments in the bootstrap function and the assignment of the confidence interval using a dot(.) method. In the assignment of the arguments in the bootstrap function, the 95% confidence level had to be specified, the type of confidence value needed, in this case the *mean* (λ), and the method of calculation. The assignment of the lower and upper confidence level values, that is, the lowest and highest λ values where the calculated λ value of the time-to-failure values should found, used an in-build `.confidence_interval` method of the `bootstrap()` function. Consequently, the confidence interval of the λ value (in minutes) is

$$mean \in [2.87, 5.45].$$

Therefore, the use of $mean_{new} = 4.058$ (with $\lambda = 1/4.058 = 0.2464$) is the best approach since it was calculated using the bootstrap method and it falls within a 95% confidence interval calculated above. With all these parameters in place, both Equation 3.10 and Equation 3.11 are fully complete as seen in Equation 4.1 and Equation 4.2.

$$F(t) = 1 - e^{-0.2464(t-0.9675)^{1.155}} \tag{4.1}$$

$$y = 1.155ln(0.2464) + 1.155ln(t - 0.9675) \tag{4.2}$$

Let $x = ln(t - \tau)$, Equation 4.2 can be re-written as follows

$$\begin{aligned} y &= 1.155ln(0.2464) + 1.155x. \\ \therefore y &= 1.155x - 1.6179. \end{aligned} \tag{4.3}$$

To answer the question, *what is the reliability of an AV with respect to traffic-rules*, Equation 4.1 was utilised. But first, the assumption in section 3.1.2.1 were applied, more specifically to assumption 3, i.e., the AVs are tested for 20 minutes. Therefore, the reliability of the AV at 20 minutes is calculated as follows

$$
\begin{aligned}
R(t) &= 1 - \left(1 - e^{-0.2464(20.00-0.9675)^{1.155}}\right) \\
&= e^{-0.2464(20.00-0.9675)^{1.155}} \\
&= 6.089 \times 10^{-4} \\
&= 0.06089\% \text{ reliable.}
\end{aligned}
$$

This means there is $1 - R(t) = 99.94\%$ chance that the AV will fail at least one of the rule in 20 minutes. This is because of the increasing *hazard rate* ($h(t)$). the hazard rate is increasing when $\beta > 1$ (in this case $\beta = 1.155$). Equation 4.4 is used to calculate the hazard rate.

$$
h(t) = \left(\frac{\beta}{\eta^{\beta}}\right)\left(t^{\beta-1}\right) \tag{4.4}
$$

The goal is to achieve a decreasing hazard rate (with $\beta \leq 1$), then the reliability will increase. When $\beta = 1$,

$$
h(t) = \left(\frac{1}{\eta}\right) = \left(\frac{1}{4.24}\right) = \frac{25}{106},
$$

this represents a constant hazard rate, there is no impact of time, therefore, at any given time the value of the hazard rate will remain the same. To further explain the reliability of the AV with respect to the traffic-rules, a method that utilises a *Bathtub Curve* was adopted. The curve can be seen in Figure 34.

Though the hazard rate of AVs failing at least one of the rules was calculated to be increasing since β is greater than one, the failure rate is categorised as *constant*. This is due to the fact that AVs are established, but they still have failures that are observable or constant (random) failures as seen in the bathtub curve in Figure 34.

4.1 Chapter summary

This chapter started by providing two tables that consist of the results related to the reliability of AVs with respect to traffic rules. Table 14 contains a summary of total passed traffic rules (in %). The linearised Weibull distribution plot was provided by applying the techniques described in Chapter 3. The shape, scale, and location parameters were extracted, but to continue with the analysis, a bootstrap method had to be applied. After the bootstrapping had been applied, a similar time-to-failure graph was

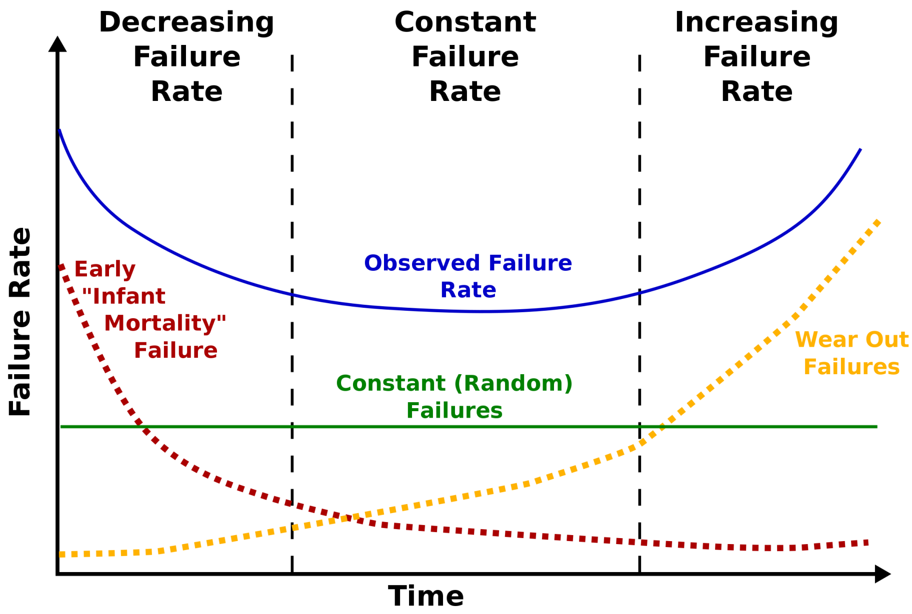


Figure 34: The Bathtub curve (adopted from Wikipedia contributors (2022a)).

observed and a mean of 4.058 minutes was calculated—the initial mean calculated was 4.066 minutes. Then, the reliability was calculated—it was concluded that within 20 minutes an AV will fail at least one of the traffic rules.

Chapter 5

Conclusion and recommendations

This chapter serves as the conclusion of this dissertation and recommendations regarding what was learned from the research conducted and its usefulness. These are now discussed.

The main problem is the complexity that the Autonomous Vehicle (AV)s (with autonomous levels of 4 and 5) possess—in terms of the diversity of navigation requirements on the road, connectivity of the system's intelligence, intelligent interaction, swift decision making, and obedience of road rules. It is nearly impossible to address all these causes at once; therefore, it was necessary to first address the complexity issue in terms of providing the functional capabilities of the AVs. Secondly, conduct a reliability analysis of the AVs concerning the traffic rules. Five functional capabilities are outlined, and the reliability analysis highlighted that the AVs are still not yet perfected. They are still being improved mainly in terms of performance so that they can make the correct decision at the right time.

The AVs are in a stage where they are useful such that they can fulfil their primary goal, which is to move from point A to B with minimal environmental harm. However, this only applies to some companies that have had good progress in developing the vehicles' intelligence. The vehicles still behave like non-advanced drivers in terms of performance and decision-making. Though some companies are doing a good job in developing the intelligence of the vehicles, the vehicle still needs monitoring. In the case of the Tesla AVs, a driver must be in front of the steering wheel should the vehicle fail to make the correct decision and execute it correctly, which has proved inevitable. Other companies do not need a driver in front of the steering wheel, but they can remotely control the vehicles, which is why they are not sold to individuals. The results outlined and discussed in Chapter 4 have proved this. According to the results, the reliability of the AVs, ,is not close to being perfect. The AVs are expected to fail at least one of the 20 rules within 20 minutes of drive. Additionally, because there are more than 20 rules the AVs should obey, they are not doing well. There is still a lot of work to be done to increase the reliability of AVs.

The reliability of AVs in terms of traffic rules is not available in the current literature. Furthermore, according to the literature, the application of systems and reliability engineering has not been applied to AVs. Therefore, the results in this dissertation have addressed the mentioned points, which closes the gap in the available literature regarding the reliability of AVs and the application of systems and reliability engineering. Furthermore, this dissertation will assist African automotive companies planning to build AVs, and this will assist Africa in becoming more technologicallytechnologically advanced.

In terms of fulfilling the objectives outlined for this dissertation (see section 1.7), Table 17 outlines where the objectives were addressed and met. The first objective was achieved by assessing the AV's reliability and secondly by modelling the reliability analysis model to evaluate the reliability of AVs. The second objective was achieved by assessing the sensors' fusion and re-designing the functional structure of AVs. The last objective was achieved by evaluating the reliability of the AV concerning traffic rules. Furthermore, the research findings and the recommendations related to the future work of the study are provided in section 5.1 and section 5.2, respectively.

Table 17: The objectives and their related sections where they were addressed and fulfilled.

Objective	**Section**	**Page**
1. Model and analyse the reliability of the intelligence of autonomous vehicles with respect to the traffic rules.	3.1.2.1	62 – 64
2. Model and analyse the inherent complexity associated with autonomous vehicles.	3.1.1	42 – 53
3. Validate the reliability (analysis with respect to the traffic rules) model using a reliability engineering technique.	4	67 – 76

5.1 Research findings

The focus of this section is to provide findings discovered from the research conducted in this dissertation and future work of the research. There are four research findings discovered, and they are outlined as follows:

Findings

1. **AV's reliability.** The reliability of an AV depends on many factors, such as the performance of the sensors, the performance of the systems (perception, decision-making, and control), and weather conditions.
   - The sensors' performance depends firstly on the type of sensors used (the good ones are usually expensive) and secondly on the placement (the area and angles around the vehicle). Finally, the sensors' performance depends on their fusion.
   - The systems of the AV are responsible for analysing the data and making the vehicle react in every possible way. Therefore, the programming and decisions made in those systems contribute a lot to an AV's performance.
   - The weather condition has proved to be a problem in all AVs. A bad weather condition affects the sensors, dirty data is read in, and unreliable decisions are made by the vehicle. Therefore, the performance is affected.
2. **Reality simulation.** It has proved to be a problem to simulate reality perfectly. That is because it is hard to simulate all possible human behaviours when they drive or walk. There is a lot of freedom associated with reality, and hence, it would help to test AVs in an environment where there is more other AV to assess their behaviour.
3. **AV's capability limitations.** From the current tested AVs, and they seem not to have reached complete (level 5) automation since these vehicles can be controlled remotely when they misbehave. They seem unable to fix themselves sometimes, making it not quite easy to trust them.
4. **AV's decision-making system.** Though autonomous in AVs depends on different aspects (such as the sensors and the built-in functional systems), there is one aspect that needs more attention. The *decision-making* system needs a lot of attention since it should be able to make a reliable decision still when the read-in data is clean or not. The AV can read clean data and understand its environment and situation correctly, but if its decision-making system is faulty, people will be in danger.

Future work

1. Though Defence Advanced Research Projects Agency (DARPA) grand challenge provided solid insight of what is meant by the AVs' performance, it is necessary to further research about their competitions and assess or review them.

2. The analysis of the reliability of signal transmission between the AVs transmission and receiver subsystems is another important aspect of AVs' performance. The signals are received from the external world with the aid of designated internal devices. These devices can be visual sensors, proximity sensors, e.t.c. which are capable of reading road signs, sign posts, sensing or sighting of other vehicles, or objects. Based on this, the AV is meant to respond by reacting. The efficiency and effectiveness of this interaction is quite significant, though it was slightly assessed in the traffic rules, it needs more research and assessment attention.

3. The effect of partial failure of a functional element would be assessed and analysed in a future research work. Fuzzy logic would be deployed to achieve this. The current research has focused on the reliability analysis of a functional element from the Boolean logic point of view i.e. fully operational or non-operational functional elements.

5.2 Recommendations

The presented results depend on the data gathered through analysing different AVs from other companies (videos online). That made the analysis less biased since some reviewers showed off the exciting news related to the existence of AVs. Therefore, it was possible to identify the mistakes the AVs would exhibit. However, the videos analysed could have been edited to make them shorter, to look professional, or maybe comply with the rules of the AV companies. Hence, it is recommended to test one or more AV of different companies in person and gather data from in-person experience. Therefore, two recommendations were deduced relating to the *future work* of this dissertation and are outlined as follows.

Recommendation 1 When looking at the fusion of sensors, it is recommended to use the camera and Radio Detection and Ranging (RADAR) since their fusion was proved to theoretically satisfy the AV's perception data gathering. Excluding the Light Detection and Ranging (LIDAR) sensor in the AVs saves a long of money (saving about R1.5 million per vehicle, as discussed in Chapter 3). It should be noted, though, that a comparison of theoretically excluding LIDAR will not drastically affect the performance of the AV since excluding LIDAR could mean other different sensors are needed. Therefore, this theory has to be verified, and the best way to very it without spending money on manufacturing and other related things is to utilise *simulation modelling*.

Recommendation 2 Regarding the reliability of the AVs, it is recommended that the companies should aim to make their AVs obey all of the traffic rules. Again, using simulation modelling can help different test scenarios on how the vehicle(s) should perform. Furthermore, the use of systems and reliability engineering will further help construct the correct functional capabilities of the vehicles and calculate their reliability. Hence, it is recommended to test the AVs in a simulation construct with correct functional capabilities, and calculate their reliability before deploying them to the public.

Bibliography

Ahmadi, S., Hajihassani, M., Moosazadeh, S., and Moomivand, H. (2020). An overview of the reliability analysis methods of tunneling equipment. *The Open Construction & Building Technology Journal*, 14(1).

Aken, J. E. v. (2004). Management research based on the paradigm of the design sciences: the quest for field-tested and grounded technological rules. *Journal of management studies*, 41(2):219–246.

Al Mamun, M. A., Berger, C., and Hansson, J. (2014). Explicating, understanding, and managing technical debt from self-driving miniature car projects. In *2014 Sixth International Workshop on Managing Technical Debt*, pages 11–18. IEEE.

Arnold, R. D. and Wade, J. P. (2015). A definition of systems thinking: A systems approach. *Procedia computer science*, 44:669–678.

Badue, C., Guidolini, R., Carneiro, R. V., Azevedo, P., Cardoso, V. B., Forechi, A., Jesus, L., Berriel, R., Paixao, T. M., Mutz, F., et al. (2021). Self-driving cars: A survey. *Expert Systems with Applications*, 165:113816.

Bastidas-Arteaga, E. and Soubra, A.-H. (2014). Reliability analysis methods.

Behringer, R., Sundareswaran, S., Gregory, B., Elsley, R., Addison, B., Guthmiller, W., Daily, R., and Bevly, D. (2004). The darpa grand challenge-development of an autonomous vehicle. In *IEEE Intelligent Vehicles Symposium, 2004*, pages 226–231. IEEE.

Berger, C. (2014a). From a competition for self-driving miniature cars to a standardized experimental platform: concept, models, architecture, and evaluation. *arXiv preprint arXiv:1406.7768*.

Berger, C. (2014b). Towards continuous integration for cyber-physical systems on the example of self-driving miniature cars. In *Continuous Software Engineering*, pages 117–126. Springer.

Borenstein, J., Herkert, J. R., and Miller, K. W. (2019). Self-driving cars and engineering ethics: The need for a system level analysis. *Science and engineering ethics*, 25(2):383–398.

Campbell, S., O'Mahony, N., Krpalcova, L., Riordan, D., Walsh, J., Murphy, A., and Ryan, C. (2018). Sensor technology in autonomous vehicles: A review. In *2018 29th Irish Signals and Systems Conference (ISSC)*, pages 1–4. IEEE.

Casado-Herráez, D. (2020). Self-driving car autonomous system overview-industrial electronics engineering-bachelors' thesis.

Chandana, P. H. I. M. A. (2020). Obtaining a complete assessment of angiography results with convolutional neural network. *Journal of Natural Remedies*, 21(4):105–111.

Chy, M., Amzad, K., Masum, A. K. M., Sayeed, K. A. M., and Uddin, M. Z. (2022). Delicar: a smart deep learning based self driving product delivery car in perspective of bangladesh. *Sensors*, 22(1):126.

Deac, M.-A., Al-doori, R. W. Y., Negru, M., Dănescu, R., et al. (2018). Miniature autonomous vehicle development on raspberry pi. In *2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 229–236. IEEE.

Denoël, V. (2007). An introduction to reliability analysis.

Dhillon, B. S. (2009). Mining equipment safety: a review, analysis methods and improvement strategies. *International Journal of Mining, Reclamation and Environment*, 23(3):168–179.

Dresch, A., Lacerda, D. P., and Antunes, J. A. V. (2015). Design science research. In *Design science research*, pages 67–102. Springer.

Figueiredo, F. and Gomes, M. I. (2013). The skew-normal distribution in spc. *REVSTAT-Statistical Journal*, 11(1):83–104.

Firdos, M., Abbas, K., Ahmed Abbasi, S., and Yousaf Abbasi, N. (2020). Modeling and comparison of maximum likelihood and median rank regression methods with fréchet distribution. *Proceedings on Engineering Sciences*, 2(2):159–168.

Ghosh, A., Sufian, A., Sultana, F., Chakrabarti, A., and De, D. (2020). Fundamental concepts of convolutional neural network. In *Recent Trends and Advances in Artificial Intelligence and Internet of Things*, pages 519–567. Springer.

Goecks, L. S., Souza, M. d., Librelato, T. P., and Trento, L. R. (2021). Design science research in practice: review of applications in industrial engineering. *Gestão & Produção*, 28.

González, D., Pérez, J., Milanés, V., and Nashashibi, F. (2015). A review of motion planning techniques for automated vehicles. *IEEE Transactions on intelligent transportation systems*, 17(4):1135–1145.

Grigorescu, S., Trasnea, B., Cocias, T., and Macesanu, G. (2020). A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386.

Guanetti, J., Kim, Y., and Borrelli, F. (2018). Control of connected and automated vehicles: State of the art and future challenges. *Annual reviews in control*, 45:18–40.

Hallinan Jr, A. J. (1993). A review of the weibull distribution. *Journal of Quality Technology*, 25(2):85–93.

Hnewa, M. and Radha, H. (2020). Object detection under rainy conditions for autonomous vehicles: a review of state-of-the-art and emerging techniques. *IEEE Signal Processing Magazine*, 38(1):53–67.

Hoole, G. (2013). *The new official k53 manual: Motorcycles, light and heavy vehicles.* Penguin Random House South Africa.

Ignatious, H. A., Khan, M., et al. (2022). An overview of sensors in autonomous vehicles. *Procedia Computer Science*, 198:736–741.

Jo, K., Kim, J., Kim, D., Jang, C., and Sunwoo, M. (2015). Development of autonomous car—part ii: A case study on the implementation of an autonomous driving system based on distributed architecture. *IEEE Transactions on Industrial Electronics*, 62(8):5119–5132.

Kannapiran, S. and Berman, S. (2020). Go-chart: A miniature remotely accessible self-driving car robot. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2265–2272. IEEE.

Kissell, R. L. and Poserina, J. (2017). *Optimal sports math, statistics, and fantasy.* Academic Press.

Komninos, N. (2008). *Intelligent cities and globalisation of innovation networks*. Routledge.

Kossiakoff, A., Sweet, W. N., Seymour, S. J., and Biemer, S. M. (2011). *Systems engineering principles and practice*, volume 83. John Wiley & Sons.

Lai, C.-D., Murthy, D. N., and Xie, M. (2006). Weibull distributions and their applications. In *Springer Handbooks*, pages 63–78. Springer.

Ma, Y., Wang, Z., Yang, H., and Yang, L. (2020). Artificial intelligence applications in the development of autonomous vehicles: a survey. *IEEE/CAA Journal of Automatica Sinica*, 7(2):315–329.

Mackie, T. (2018). Proving liability for highly and fully automated vehicle accidents in australia. *Computer Law & Security Review*, 34(6):1314–1332.

Matthaei, R. and Maurer, M. (2015). Autonomous driving–a top-down-approach. *at-Automatisierungstechnik*, 63(3):155–167.

Mishra, M. K. (2020). Smart cities in the era of globalisation.

Misra, K. B. (2008). Reliability engineering: A perspective. In *Handbook of Performability Engineering*, pages 253–289. Springer.

Ndung'u, N. and Signe, L. (2020). The fourth industrial revolution and digitization will transform africa into a global powerhouse. *Foresight Africa Report*.

O'Connor, P. and Kleyner, A. (2012). *Practical reliability engineering*. John Wiley & Sons.

Paden, B., Čáp, M., Yong, S. Z., Yershov, D., and Frazzoli, E. (2016). A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on intelligent vehicles*, 1(1):33–55.

Rinne, H. (2008). *The weibull distribution: a handbook*. Chapman and Hall/CRC.

Rosique, F., Navarro, P. J., Fernández, C., and Padilla, A. (2019). A systematic review of perception system and simulators for autonomous vehicles research. *Sensors*, 19(3):648.

Schwarting, W., Alonso-Mora, J., and Rus, D. (2018). Planning and decision-making for autonomous vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:187–210.

Sell, R., Leier, M., Rassõlkin, A., and Ernits, J.-P. (2018). Self-driving car iseauto for research and education. In *2018 19th International Conference on Research and Education in Mechatronics (REM)*, pages 111–116. IEEE.

Serban, A., Poll, E., and Visser, J. (2020). A standard driven software architecture for fully autonomous vehicles. *Journal of Automotive Software Engineering*, 1.

Shahian Jahromi, B., Tulabandhula, T., and Cetin, S. (2019). Real-time hybrid multi-sensor fusion framework for perception in autonomous vehicles. *Sensors*, 19(20):4357.

Sharma, O., Sahoo, N. C., and Puhan, N. B. (2021). Recent advances in motion and behavior planning techniques for software architecture of autonomous vehicles: A state-of-the-art survey. *Engineering applications of artificial intelligence*, 101:104211.

Shelhamer, E., Long, J., and Darrell, T. (2016). Fully convolutional networks for semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):640–651.

Singh, S. and Saini, B. S. (2021). Autonomous cars: Recent developments, challenges, and possible solutions. In *IOP Conference Series: Materials Science and Engineering*, volume 1022, page 012028. IOP Publishing.

Sviatov, K., Yarushkina, N., Kanin, D., Rubtcov, I., Jitkov, R., Mikhailov, V., and Kanin, P. (2021). Functional model of a self-driving car control system. *Technologies*, 9(4):100.

Tan, S. Y. and Taeihagh, A. (2021). Adaptive governance of autonomous vehicles: Accelerating the adoption of disruptive technologies in singapore. *Government Information Quarterly*, 38(2):101546.

Vaicenavicius, J., Wiklund, T., Grigaitė, A., Kalkauskas, A., Vysniauskas, I., and Keen, S. (2020). Self-driving car safety quantification via component-level analysis. *arXiv preprint arXiv:2009.01119*.

Vargas, J., Alsweiss, S., Toker, O., Razdan, R., and Santos, J. (2021). An overview of autonomous vehicles sensors and their vulnerability to weather conditions. *Sensors*, 21(16):5397.

Wikipedia contributors (2022a). Bathtub curve — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Bathtub_curve&oldid=1118902532. [Online; accessed 19-November-2022].

Wikipedia contributors (2022b). History of self-driving cars — Wikipedia, the free encyclopedia. [Online; accessed 9-February-2022].

Wikipedia contributors (2022c). State of the art — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=State_of_the_art&oldid=1077324612. [Online; accessed 31-March-2022].

Wu, J. (2017). Introduction to convolutional neural networks. *National Key Lab for Novel Software Technology. Nanjing University. China*, 5(23):495.

Yan, C., Xu, W., and Liu, J. (2016). Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle. *Def Con*, 24(8):109.

Yeong, D. J., Velasco-Hernandez, G., Barry, J., and Walsh, J. (2021). Sensor and sensor fusion technology in autonomous vehicles: A review. *Sensors*, 21(6):2140.

Zhao, L., Ichise, R., Liu, Z., Mita, S., and Sasaki, Y. (2017). Ontology-based driving decision making: A feasibility study at uncontrolled intersections. *IEICE TRANSACTIONS on Information and Systems*, 100(7):1425–1439.

Zhao, L., Ichise, R., Yoshikawa, T., Naito, T., Kakinami, T., and Sasaki, Y. (2015). Ontology-based decision making on uncontrolled intersections and narrow roads. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 83–88.

Zug, S., Steup, C., Scholle, J.-B., Berger, C., Landsiedel, O., Schuldt, F., Rieken, J., Matthaei, R., and Form, T. (2014). Technical evaluation of the carolo-cup 2014: A competition for self-driving miniature cars. In *2014 IEEE International Symposium on Robotic and Sensors Environments (ROSE) Proceedings*, pages 100–105. IEEE.

Appendix A

Architecture design

The design in Figure 35 represents the architectural design for a Connected and Automated Vehicle (CAV).

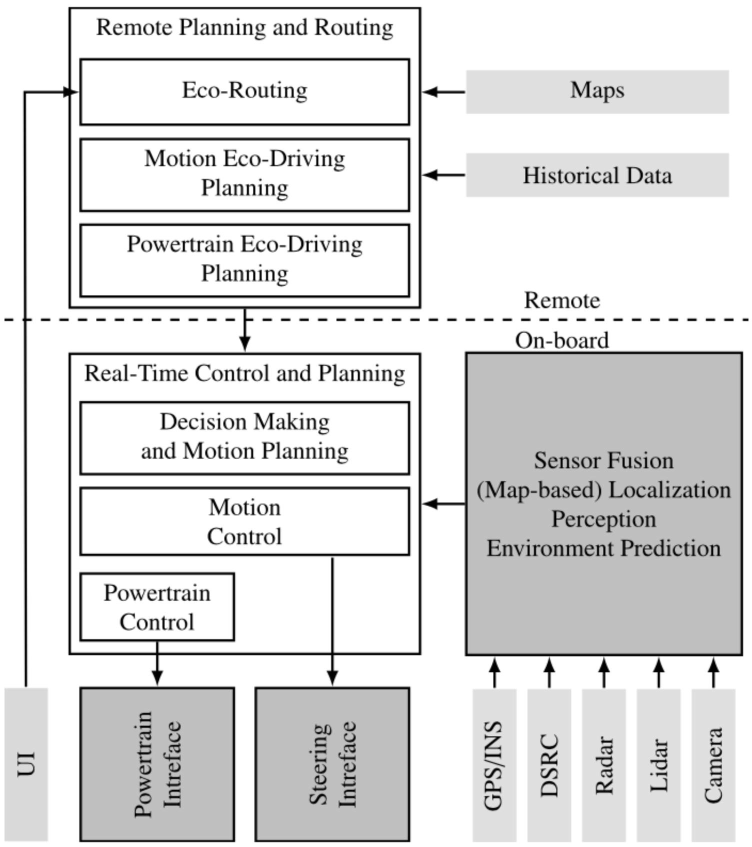


Figure 35: The architectural design for CAV (adopted from (Guanetti et al., 2018)).

The typical architecture of an autonomous system of a AV can be seen in Figure 36 as designed by Badue et al. (2021).

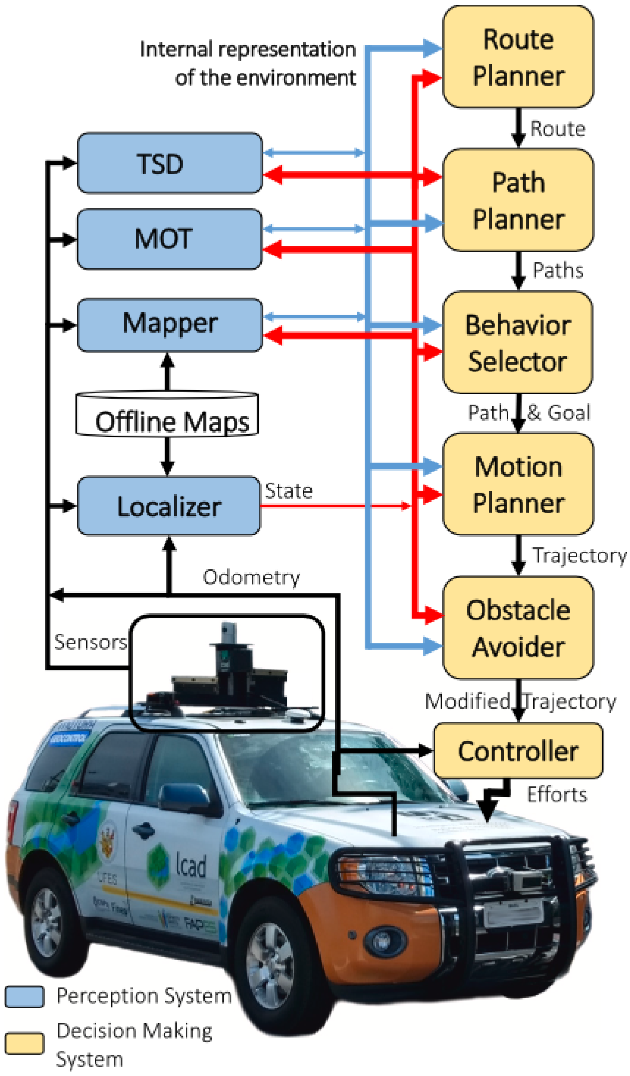


Figure 36: The typical architectural design for a AV (adopted from (Badue et al., 2021)).

Appendix B

AV sensor comparison

This chapter focused on different kind of sensors that were compared against their different type of sensors that have unique characteristics and have different level of performance. Ignatious et al. (2022) constructed three tables of three different kind of sensors (camera, LIDAR, and RADAR) that are related. The tables (Figure 37 through 40) focused on the different specifications of the three sensors.

Figure 37 and 38 provides the specifications of LIDAR sensors. The acronyms that appears in those figures are defined as follows.

VFoV:	Vertical-Field-of-View
HFoV:	Horizontal FoV
FPS:	Frames per second
RNG:	Detecting Range
Acc:	Accuracy
HR:	Horizontal Resolution
VR:	Vertical Resolution
λ:	Wavelength

Category	Company	Model	Channels/ Layers	FPS(Hz)	Acc(m)	RNG(m)	VF OV (°)	HFO V(°)	HR	VR	λ	Ref
Mechanical / Spinning LiDARS	Velodyne	VLP-16 VLP– 32C HDL-32E HDL-64E VLS 128 (Alpha Prime)	16 32 32 64 128	5-20 5-20 5-20 5-20 5-20	±0.03 ±0.03 ±0.02 ±0.02 ±0.03	1..100 1..200 2-100 3..120 Max 245	30 40 41.3 3 26.8 40	360 360 360 360 360	0.1-04 0.1-04 0.08- 0.33 0.09 0.1-0.4	2 0.33 1.33 0.33 0.11	903 903 903 903 903	[15]
	Hesai	Pandar64 Pandar40P	64 40	10,20 10,20	±0.02 ±0.02	0.3..20 0.3..200	40 40	360 360	0.2,0.4 0.2,0.4	0.167 0.167	905 05	[16]
	Ouster	OSI-64 Gen1 OSI-16 Gen 1	64 16	10,20 10.20	±0.03 ±0.03	0.8—120 0.8120	33.2 33.2	360 360	0.7.0.35	0.53 0.53	850 850	[17]
	RoboSense	RS-LiDAR 32	32	5.10,20	±0.03	0.4-200	40	360	0.18,10.3 6	2	905	[18]
	LeiShen	C32-151A C16-700B	32 16	5,10,20 5,10,20	±0.03 ±0.02	0.5-..70 0.5..150	32 30	360 360	0.09 0.18,0.36	1 2	905 905	[19]
	Hokuyo	YVT-35LX- F0	-	20	±0.05	0.3..35	40	210	-	-	905	[20]
Solid State LiDARS	IBEO	LUX 4L Standard LUX HD LUX SL	4 4 8	25 25 25	0.1 0.1 0.1	50 50 30	3.2 3.2 6.4	110 110 110	0.25 0.25 0.25	0.8 0.8 0.8	905 905 905	[21]
	SICK	LD- MRS400102S 01 HD LD- MRS8001S01 HD	4 8	50 50	-	30 50	3.2 6.4	110 110	0.125- 0.5 0.125- 0.5	- -	- -	[19]

Figure 37: Different type of LIDAR compared—part A (adopted from Ignatious et al. (2022)).

The table in Figure 37 was split into two parts, the other part can be observed in Figure 38.

	Cepton	Vista P60 Vista P90 Vista X90	- - -	10 10 40	- - -	200 200 200	22 27 25	60 90 90	0.25 0.25 0.13	0.25 0.25 0.13	905 905 905	[16]

Figure 38: Different type of LIDAR compared—part B (adopted from Ignatious et al. (2022)).

	Company	Model	Channels or Layers	FPS (Hz)	Acc. (m)	RNG (m)	VFOV (°)	HFOV (°)	HR (°)	VR (°)	λ (nm)	Ø (mm)
Mechanical/Spinning LiDARs	Velodyne	VLP-16	16	5–20	±0.03	1 … 100	30	360	0.1–0.4	2	903	103.3
		VLP-32C	32	5–20	±0.03	1 … 200	40	360	0.1–0.4	0.33 [1]	903	103
		HDL-32E	32	5–20	±0.02	2 … 100	41.33	360	0.08–0.33	1.33	903	85.3
		HDL-64E	64	5–20	±0.02	3 … 120	26.8	360	0.09	0.33	903	223.5
		VLS-128 Alpha Prime	128	5–20	±0.03	max 245	40	360	0.1–0.4	0.11 [1]	903	165.5
	Hesai	Pandar64	64	10,20	±0.02	0.3 … 200	40	360	0.2,0.4	0.167 [1]	905	116
		Pandar40P	40	10,20	±0.02	0.3 … 200	40	360	0.2,0.4	0.167 [1]	905	116
	Ouster	OS1–64 Gen 1	64	10,20	±0.03	0.8 … 120	33.2	360	0.7,0.35, 0.17	0.53	850	85
		OS1-16 Gen 1	16	10,20	±0.03	0.8 … 120	33.2	360		0.53	850	85
	RoboSense	RS-Lidar32	32	5,10,20	±0.03	0.4 … 200	40	360	0.1–0.4	0.33 [1]	905	114
	LeiShen	C32-151A	32	5,10,20	±0.02	0.5 … 70	32	360	0.09, 0.18,0.36	1	905	120
		C16-700B	16	5,10,20	±0.02	0.5 … 150	30	360		2	905	102
	Hokuyo	YVT-35LX-F0	-	20 [3]	±0.05 [3]	0.3 … 35 [3]	40	210	-	-	905	◊
Solid State LiDARs	IBEO	LUX 4L Standard	4	25	0.1	50 [2]	3.2	110	0.25	0.8	905	◊
		LUX HD	4	25	0.1	50 [2]	3.2	110	0.25	0.8	905	◊
		LUX 8L	8	25	0.1	30 [2]	6.4	110	0.25	0.8	905	◊
	SICK	LD-MRS400102S01 HD	4	50	-	30 [2]	3.2	110	0.125 … 0.5		-	◊
		LD-MRS800001S01	8	50	-	50 [2]	6.4	110	0.125 … 0.5		-	◊
	Cepton	Vista P60	-	10	-	200	22	60	0.25	0.25	905	◊
		Vista P90	-	10	-	200	27	90	0.25	0.25	905	◊
		Vista X90	-	40	-	200	25	90	0.13	0.13	905	◊

Figure 39: Different type of LIDAR compared (adopted from Yeong et al. (2021)).

	Aptiv Delpi		Continental	SmartMicro
Category	ESR2.5	SRR@	ARS 408-21	UMMR-96-T-153
Freq(GHz)	76.5	76.5	76…77	79(77..81)
HFOV(°) Short-Range Mid-Range Long-Range	±45 ±10	±75	±9 ±60	≥ 130 ≥ 130 ≥ 100 (squint beam)
VFOV(°) Short-Range Long-Range	4.4	10	20 14	15
Range(m) Short-Range Mid-Range Long-Range	-	±0.5 noise and ±0.5% bias	-	<0.15 (or) 1% (bigger of) <0.30 (or) 1% (bigger of) <0.50 (or) 1% (bigger of)
Vel Range (km/h) Short-Range Mid-Range Long-Range	-	-	$-400 \ldots +200$	$-400 \ldots +100$ $-340 \ldots +140$ $-340 \ldots +140$
IO Interfaces	CAN/Ethernet	PCAN	CAN	CAN/Automotive Ethernet

Figure 40: Different type of RADAR compared (adopted from Vargas et al. (2021)).

The comparison of what different sensors can can do according to Ignatious et al. (2022) can is provided in Figure 41. Here it can be noted that the sensor fusion of the *resolution* of camera, LIDAR, and/or *RADAR* was omitted as discussed in Chapter 2.

Factors	Camera	LiDAR	RADAR	Fusion
Range	--	--	✓	✓
Resolution	✓	--	×	
Distance Accuracy	--	✓	✓	✓
Velocity	--	×	✓	✓
Color Perception, e.g Traffic lights	✓	×	×	✓
Object Detection	×	✓	✓	✓
Object Classification	✓	×	×	✓
Lane Detection	✓	×	×	✓
Obstacle Edge Detection	✓	✓	×	✓
Illuminations Conditions	×	✓	✓	✓
Weather Conditions	×	--	✓	✓

Figure 41: Different sensor comparison according to Ignatious et al. (2022).