Technische Universiteit
Eindhoven
University of Technology

Active Lane Centering System

Authors:
Anshuman Singh
Davide Occello
Raymond Wouters
Sharad Bhadgaonkar

Automotive Systems Design
Stan Ackermans Institute
Eindhoven University of Technology

January 30, 2017

Active Lane Centering System

HIGH LEVEL DESIGN DOCUMENT

Eindhoven University of Technology
Stan Ackermans Institute / Automotive Systems Design

The design that is described in this report has been carried out in accordance with the TU/e code of scientific conduct

Partners:

Technische Universiteit
Eindhoven
University of Technology

Stakeholders:
Gijs Dubbelman
Gerardo Daalderop
Rameez Ismail
Peter Heuberger

TECHNISCHE UNIVERSITEIT EINDHOVEN

Abstract

Automotive System Design
Department of Mathematics and Computer Science

Active Lane Centering System

by ASD Group ALC

This report presents a High Level Design Document of an Active Lane Centering System (ALC). This document represents the deliverable for the second phase of the project, which mainly focuses on the high level design of the ALC system with a top down approach, and later on the analysis and improvement of the existing hardware architecture. The high level design was derived using the CAFCR approach, which derives a conceptual architecture from customer objectives. This design was combined with the insights coming from the previous phase on functional safety in order to produce a safer concept architecture. The second part of this phase involved the analysis of the hardware components of the existing architecture and of the possible automotive grade communication protocols. Based on the insights deriving from this analysis, a modified version of the architecture is proposed, which aims to improve the safety, the performance and the practical implementation of the system. The proposed architectures follow the development process of the final ALC system. One will be implemented in the following phase of this project, and other two architecture proposals which will be more efficient and compact for future implementations.

Contents

Chapter 1

Introduction

Advanced driver assistance systems are one of the fastest-growing segments in automotive electronics [11]. These are systems developed to automate/adapt/enhance the vehicle systems for safety and better driving. Safety features are designed to avoid collisions and accidents by offering technologies that alert the driver to potential problems, or to avoid collisions by implementing safeguards and taking over control of the vehicle. A large percentage of accidents occur due to unintentional lane change [7]. Active Lane Centering (ALC) is a system designed to avoid such accidents or collisions by actively maintaining the vehicle at the center of the lane. If unwanted drift from lane center is detected, ALC will perform an appropriate control action aimed to steer the vehicle back to the desired trajectory. Hence, broad level goals for ALC system are to improve the safety of the driver by keeping the car in the safest zone of the lane and reducing the driver workload by automating the most common lateral control actions.

1.1 Project Objective

The second phase of the ALC project, focuses on the delivery of a high level design of the ALC system. The design is broken down into different phases, which follows the CAFCR Framework as a guideline. The major project objectives of this phase are listed below.

- Definition of the Customer Objectives
- Definition the Application areas of the system
- Derivations of Functional Requirements
- Improvement of the Concept functional architecture
- Analysis of the available Hardware Components
- Benchmarking of communication technologies
- Proposal of High Level Architecture Designs

All the information which was acquired in the previous phase [2] is used in this phase to deliver a safer architecture.

Chapter 2

System Design

2.1 Introduction to CAFCR Framework

The CAFCR approach is a method which is used to transform customer objective and key drivers in an actual system design. This method consists of five different views that include: customer view, application view, functional view, conceptual view and realization view. The customer view deals with the desires of the customer in terms of key drivers. The application view describes the needs of the customer in terms of how the customer would like to realize his goals. These two views provide the justification for the design of the system in the other three phases, or in other words, **why** the system should have certain functions. The functional view describes the system from an external perspective and shows **what** the system should do .
The last two phases describe **how** this functionality is realized. The CAFCR method is key to transform the customer's objectives into a possible solution which ensures a good system design which complies with the customer needs [4]. The five views are shown in Fig. 2.1.

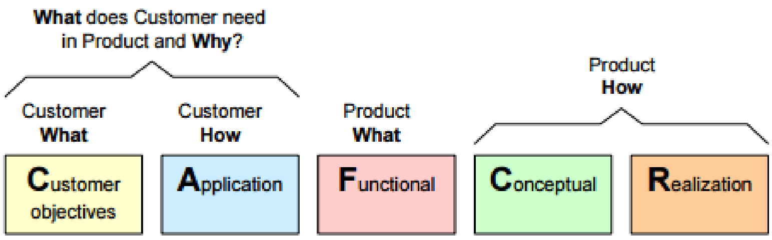

FIGURE 2.1: CAFCR Framework [4]

2.2 Customer View

The customer view from the CAFCR framework is used to capture the key drivers of different stakeholders. The combined customer objectives are listed in in Fig. 2.2 and are ranked in terms of performance (top-to-bottom). These can be divided in: Safety, Comfort, Maintainability and User Friendliness. Safety is a common objective among the stakeholders, Comfort ensure a system which will be pleasurable to use on top of the safety requirements. The maintainability originates from the desire to improve the ALC in the future, as this is a project that aims to deliver a prototype. Also in the future, the ACC system will be integrated with the ALC system and therefore this makes maintainability a key driver.Finally, User Friendliness also ensures a pleasurable system for the user.

2.3 Application view

The customers objectives are further clarified in the application view, which are again shown in Fig. 2.2. This view also deals with the demands from other users, which are in this case the regulation/law, driver, car manufacturer and dealer. For now, the car manufacturer and dealer are assumed to be out of scope, since they do not impose strict functional requirements on the system yet. Two examples regarding the application view is explained in the following lines. First of all, to ensure safety, the system should be robust, designed in compliance with the standards and the sensor measurements should be reliable. Secondly, driver comfort can be ensured by reducing the workload of the driver, ensure smooth operation and feedback the ALC operating status back to the driver.

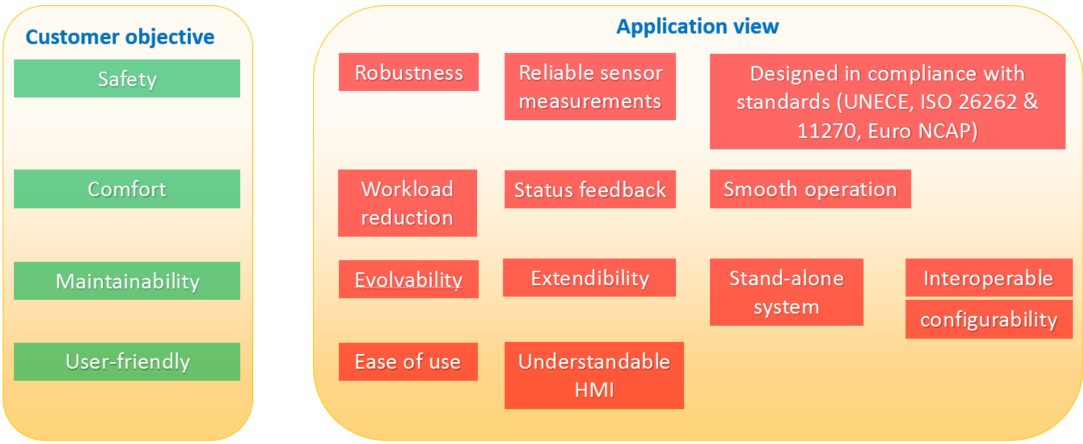

FIGURE 2.2: C and A view within the CAFCR Framework

2.4 Functional View

In the functional view of the CAFCR approach, the system is seen as a black box. This means to view it from an external perspective, without knowing how it works (Ref. Figure 2.3) and determining **what** the system should do, rather than **how** it should do it. This view is used to derive the effective functional and non-functional requirements based on the input/output behavior, and by considering interfaces, restrictions, boundaries, exceptions and regulations.

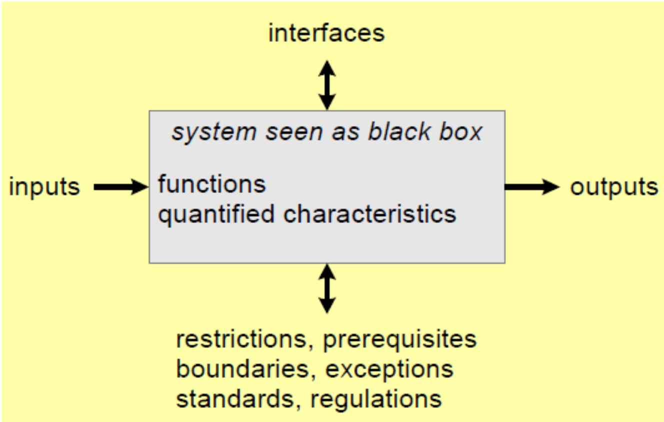

FIGURE 2.3: Black box view of Systems

2.4.1 Use cases

The ALC system will have to work in a wide range of conditions, and react appropriately to each situation which is considered to be in the scope of the project.

A basic situation the system will encounter is a curve, on a highway, with radius between 250m and 1000 m, vehicle speed between 50 km/h and 130 km/h and lane markings in compliance with the EU highway regulation. Figure 2.4 shows a graphical representation of such a situation. In this use case, the system is supposed to react by steering the car back to the center of the lane, without flashing any warning. The full reference to all the scenarios we considered can be found in the previous report in Appendix B [2].
The use cases, the EURO NCAP requirements and the Customer Objectives and Application view combined helped in the determination of the following list of functional and non functional requirements.

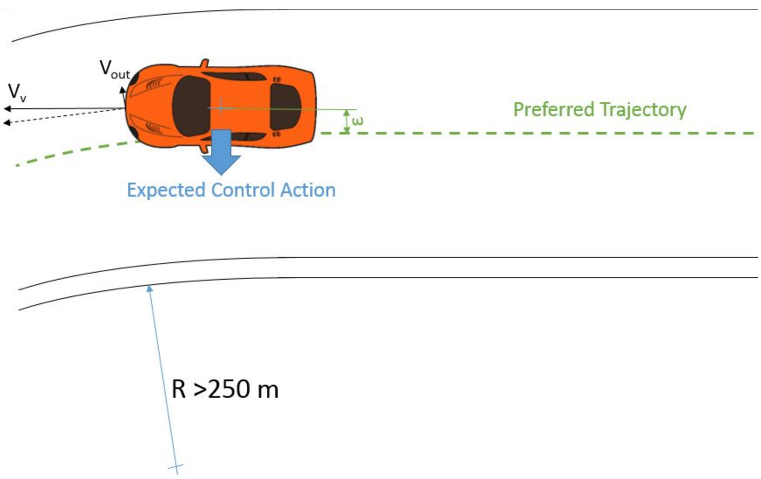

FIGURE 2.4: Example of use case for ALC [4].

2.4.2 Functional Requirements

Each customer objective is broken down in a multitude of desired characteristics of the system in the previous phases as can be seen in Figure 2.2. The following requirements are written in the effort of fulfilling these characteristics.
The Subscript F and NF in each requirement must be read as functional and nonfunctional requirements respectively. Whenever necessary, source for the requirement is provided. If the source is not specifically written, assume the requirement as the part of own requirements.

Safety

1. **F**: ALC shall be default enabled (ON) at every key cycle once Engine is on. (Source: [2])
2. **F**: ALC shall act to keep the vehicle at the center of the lane within a tolerance of 0.15 meters. (Source: [2])
3. **F**: Additionally, ALC shall trigger a warning when unintentional lane change is detected.
4. **NF**: Unintentional change shall be detected at latest when
   - outside of the tire closest to the outside of the lane markings crosses 0.3 m. (Source: [2])
   - no turn signal available.
5. **F**: The driver shall be able to overrule (take control of) the system at any point of time.
6. **F**: ALC shall be deactivated in less than 0.5 second when :
   - Manually disabled (OFF) by the user
   - ALC is active and driver applies a counter steer torque (opposite to assist torque), with a torque value of more than 0.3 Nm.
   - ALC is active and driver doesn't intervene (e.g. keeping hands on the steering wheel) to the steering wheel within 5 seconds.
   - Driver activates the Turn signal.
   - Driver brakes (more than 50%).
   - Engine is off

   (Note: Quantified values to be calibrated on the vehicle.)
7. **F**: LCA shall warn the driver when driver doesn't intervene (e.g. keeping hands on the steering wheel) within 3 seconds when ALC is actively steering the vehicle to the center. (Note: Quantified values to be calibrated on the vehicle.)
8. **NF**: LCA shall be capable to detect the related component's functional failure and disable ALC with simultaneous warning to the driver.
9. **NF**: LCA shall warn the driver if the system is deactivated/disabled by the driver.

10. **F**: (Regulation) The ALC system shall be available only if vehicle possess Electronic Stability Control system in compliance with regulatory requirements. (Source: [2])

11. **F**: (Regulation) ALC shall be operational at least under below conditions while performing unintended lane change.
    - Lane width between 3 to 3.7 m
    - Dashed line on one side having width of 0.1 to 0.25
    - Solid line on other side with 0.1 to 0.25
    - Dry weather conditions
    - No precipitation
    - Horizontal visibility till 1 km
    - Ambient temperature between 5 to 40 deg
    - Natural ambient illumination excess of 2000 lux for day light with no strong shadow
    - Uniform solid paved surface with consistent slope and no irregularity within a lateral distance of 3.0 m to either side. The minimum peak braking coefficient shall be 0.9
    - Wind speed less than 10 m/s
    - Slope of the surface between 0 and 1 deg
    - Original fitment of tires according to make, model, size, speed and load operating specified by the manufacturer with correct pressure.
    - Slope of the surface between 0 and 1 deg
    - Default wheel alignment measure set by the OEM

    (Source: [2])

12. **F**: (Regulation) The system must have an accuracy of:
    - 0.1 km/h in longitudinal speed
    - 0.03 m in longitudinal and lateral position
    - 0.1 degrees in heading angle
    - 0.1 deg/sec in yaw rate
    - 0.1 m/sec2 in longitudinal acceleration
    - 1 deg/sec in steering wheel velocity

    (Source: [2])

13. **F**: (Regulation) Once disabled (OFF), the ALC can be enabled only if the time between OFF and new ON exceeds 1 second. (Note: Quantified values to be calibrated on the vehicle.)

14. **F**: It shall be possible to disable (OFF) the ALC during runtime, but only when ALC is currently inactive.

15. **NF**: The system shall perform cyclic diagnostic tests on its communication interfaces and HW in order to detect latent faults.

16. **NF**: The system shall receive data from the vehicle ECU regarding:
    - Vehicle speed
    - Steering angle
    - Steering torque
    - Yaw rate
    - Longitudinal and lateral accelerations
    - Brake pedal position
    - Steering wheel buttons and levers state
    - Wheel speed information
17. **NF**: The system should be able to control the following vehicle parameters:
    - Steering wheel angle
    - Steering control mode (safe / high performance mode)
    - HMI display, icons and if available haptic and acoustic signals

Comfort

1. **F**: The ALC system shall reduce the workload of the driver by actively steering the vehicle to the center.
2. **F**: The driver must be able to enable/disable the ALC system using a hardware lever/button.
3. **NF**: The switching of control between ALC and the driver shall be smooth.
4. **F**: The steering to counter lateral deviation, shall be in smooth controlled manner and with minimal overshoot.
    - Lateral acceleration < 2 m/s2 while cornering,
    - Lateral acceleration < 0.5 m/s2 while driving straight
    - Lateral jerk < 5 m/s3 overall,
    - Longitudinal deceleration < 3 m/s2
    - If Longitudinal deceleration > 1 m/s2 then, longitudinal speed reduction < 18 km/h. (Source: [2])
5. **NF**: ALC enable (ON) and disable (OFF) button shall be easily reachable to the driver.
6. **NF**: Warning signals shall not distract the driver.

Robust

1. **F**: The ALC system shall be functional only when driving on highways with forward driving speed more than 50 Kph but less than 130 Kph.
   (Source: [2])
2. **NF**: The system shall be able to identify lane markings and lane width according to the country of operation.
   (Source: [2])

3. **F**: The ALC system shall be operational at least when driving on straight road with radius more than 1000m and 250 m on curved road, unless manually deactivated.
   (Source: [2])

4. **F**: ALC shall be disabled during when:
   - There is an uncertainty in lane detection (due to e.g. weather conditions, debris).
   - The temperature range exceeds the range of -20 and 40 deg Celsius. (Note: It must be noted that the Temperature of the Speedgoat is 0-50 deg).
   - Driving on roads other than Highways.

5. **F**: The effectiveness of ALC shall not be adversely affected by magnetic or electric field.
   (Source: [2])

6. **NF**: The failure warning signal shall be activated and remain activated while the vehicle is being driven and be reactivated after a subsequent ignition off – ignition on cycle as long as the failure exists.
   (Source: [2])

7. **NF**: When the driver is provided with a visual warning signal to indicate that the ALC is temporarily not available, for example due to inclement weather conditions, the signal shall be constant.
   (Source: [2])

Maintainability

1. **NF**: It shall be easy to maintain and upgrade the system with additional features e.g. using modularisation approach.

2. **NF**: If needed, ALC shall be able to operate standalone using only the lateral control and necessary sensors and hardware.

3. **NF**: If needed, it shall be possible to interact with other ADAS systems in the vehicle like ACC.

4. **NF**: The communication interface diagram for ALC system shall be provided.

User Friendly

1. **NF**: The ALC enable/disable lever shall be easily visible to the driver from the sitting position.

2. **NF**: The ALC enable/disable lever shall have clear markings for enabling and disabling the ALC system.

3. **NF**: The visual warning signals shall be visible even by daylight and it shall be easily readable by the driver from the driver's seat.

4. **F**: ALC shall notify the current ALC status using HMI Status:
Enabled

    Active lane centering ON

    Active LDW (Degraded) ON

    Active LDW (Degraded) OFF

    Active lane centering OFF.

Disabled

5. **F**: The warning provided to the driver shall be noticeable by the driver and be provided by:

    • At least two warning means out of optical, acoustic and haptic, or

    • One warning means out of haptic and acoustic, with spatial indication about the direction of unintended drift of the vehicle.
(Source: [2])

2.5 Conceptual view

The fourth view in the CAFCR method is the conceptual view. This view describes how the system should function to meet the requirements and key drivers from the three preceding views. The conceptual view is explained by means of a functional architecture diagram. The ALC architecture is shown in Fig. 2.5, which originates from the Functional Safety study performed in [2]. The functional architecture in Fig. 2.5 includes safety measures, to ensure safe operation of the system.

The ALC system acts to ensure that the vehicle is in center of the lane at all times to reduce the work load of the driver. The ALC system should be able to determine the vehicle position with respect to lane markings by means of sensors to determine the set point for the controller. The required sensors are shown on the left side of the architecture. On the right side of the architecture, the outputs are shown. These outputs consist of a steer actuator and warning signals (e.g. haptic and visual). Within the ALC system several actions need to be performed which are explained next.

First of all, the ALC system needs to detect the lane markings based on the sensor inputs. The lane detection is covered by the vision based sensing part. Secondly, the driver can deactivate the system by pressing a button. The deactivation/activation state of the system is checked by the On/Off supervisor. Both the detected lane markings and on/off supervisor signals are fed to a high level supervisor. The high level supervisor also uses the indicator signal input, to determine whether the driver is overtaking or not. This high level supervisor determines the operating state of the system. This can either be actuating or warning the driver. As a last functionality, the high level supervisor can also deactivate the system when an error is detected.

When the high level supervisor decides to actuate the vehicle, the lateral controller calculates the required steer torque to keep the vehicle at the center of the lane. This steer torque is send to the actuation switch, which acts as a last safety measure. On the other hand, if the high level supervisor decides to only trigger warnings, a signal is send to the Warning Signal Generator. The Warning Signal Generator determines the kind of warning and sends it to the dashboard.

The implemented safety measures are represented by the orange blocks in Fig. 2.5. The safety measures detect errors in the system. When an error occurs, the ALC system informs the driver and deactivate the system. Most of the safety measures check the arrival and correctness of data at different functions in the system. Other safety measures include formal verification of the supervisors, acknowledgments of send signals, limit steer torque and compare the send output steer torque with the actual applied steer torque.

An important safety measure is the Mobileye sensor redundancy. The Mobileye lane detection output is compared with the other lane detection algorithm to increase the safety of the system. When the position with respect to lane markings differ a lot between both algorithms, the system shall trigger a warning to the driver and deactivate the system. This description of the lane detection algorithm comparison concludes the brief explanation of the conceptual view. More information about the individual components of the system is provided in the next Chapter.

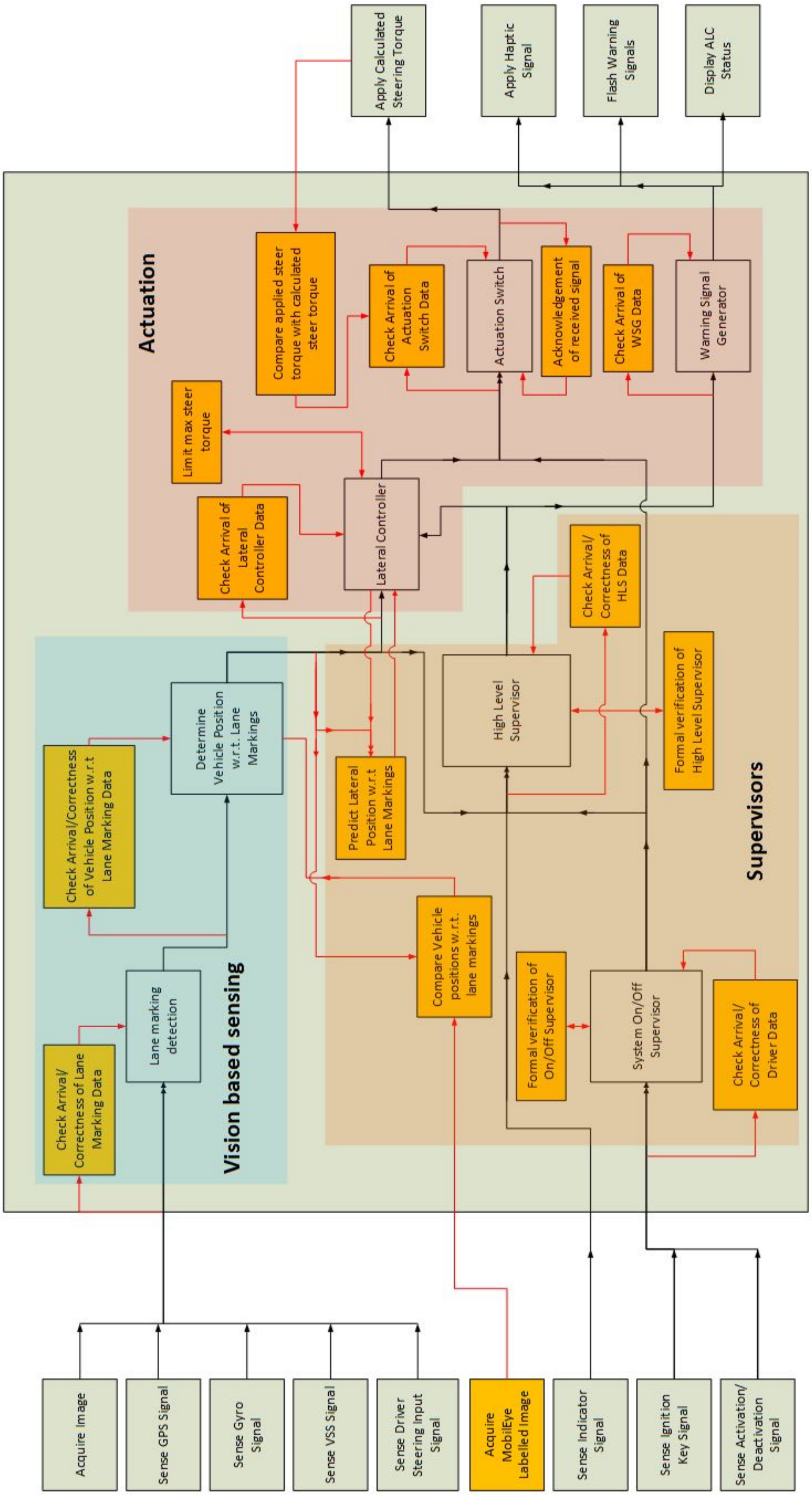

FIGURE 2.5: Functional Architecture of the system

Chapter 3

Hardware Architecture Design

This chapter describes the design of the hardware architecture of the ALC system, going though all the steps which were followed.

The design started from the Original Architecture (Figure 3.4), which was analyzed in detail from the performance, safety and practical realization point of view. The hardware specifications of the available components was analyzed. The communication strategies were compared and analyzed. This analysis lead to the proposal of alternative HW architectures which are compatible with the requirements. A selection between these will be done in the next phases.

3.1 Component specification

The aim of this section is to briefly describe the HW components of the architecture that are key to the implementation of ALC. Figure 3.4 shows the original architecture with all the components which are available for the implementation phase of the project.

The selected components are the following:

- Processors:

  Real-time target PC (SpeedGoat)

  NXP Bluexbox

- Sensors and Actuators:

  TNO MoveBox

  Mobileye C2-270

  HDR Mono Camera

  XSens Mti-200 VRU (Accelerometer, Gyroscope, GPS)

The next few paragraphs will briefly describe the most important among these.

3.1.1 Speedgoat Real Time Target PC

Speedgoat provides a range of real time target computers for rapid vision, DSP and control prototyping. The hardware along with I/O modules can be used with Simulink Real time package for Hardware In Loop (HIL) simulations and embedded application deployment. The target computer used will be the performance real-time target machine from speedgoat which is meant for the laboratory use and has the powerful computing power. Speedgoat provides support for a wide range of I/O modules and communication protocols. The hardware specifications, I/O modules and communication protocols are listed in Appendix A.

3.1.2 NXP Bluebox

Bluebox is a mobile computing platform which can be used for development of ADAS functionalities. This provides a single computing solution for testing and developing ADAS applications which need to process sensor data from different sources and decide control action for the vehicle in real time. It enables autonomy up to SAE level 4 for self-driving cars and possesses ASIL-B readiness. The Bluebox has two different processors namely a S32V234 automotive vision/sensor fusion processor and a LS2085A embedded compute processor. Both the processors run independent embedded Linux systems. Therefore the main host communication interface is the Linux terminal which can be accessed via UART or Ethernet. The Ethernet is a faster and more reliable connection than UART and is therefore preferred over UART. The Bluebox has a number of different kinds of ports to connect to different kinds of inputs like CAN and other sensor data. The hardware specification for the Bluebox is listed in Appendix A.

3.1.3 TNO MOVE BOX

The TNO MOVE box, is an interface between the CAN network of the Toyota Prius 3rd generation and the CAN network connected to the external processors and sensors.

The main functions of the of the Movebox are the following:

- Connection between internal and external network
- Safety

FIGURE 3.1: Movebox description [9]

Connection between internal and external network

The interface is capable of transferring a large amount of sensor data from the car network to the external network, and also control actions from the external network to the car (See figure 3.1).

This is a very important task as it enables the testing of the ALC on the selected car without having to know the proprietary CAN messages of the Toyota Prius, but simply knowing the messages of the Movebox. The following list is a selection of vehicle data which will be used in the project.

Sensor data:

- User requested steering torque
- Steering angle
- Velocity
- Four independent wheel speeds
- Brake pressed
- Steering wheel buttons and lever
- Lateral acceleration
- Yawrate

Control Actions:

- Control of HMI display icons (e.g. warning, Adaptive Cruise Control interface, Lane Keeping Assist interface)
- Controlling steering wheel angle
- Controlling Steering control mode (Safe Mode or Performance Mode)

Safety

The Movebox ensures that the connection between the two networks adheres to certain safety requirements. The way the system ensures safety is by providing a set of hard limitations for control parameters and by transitioning to a safe state (user in control) when problems arise.

The following safety functions are relevant for the lateral control 3.1:

- Steering torque is limited to 1.5 Nm
- Steering angle is limited to +/- 540 degrees, further limits are given according to the vehicle speed [9]
- Steering wheel rotational speed is limited to 200 rad/s
- The enabling of lateral control is possible only under 25 km/h (it is suggested also to enable it at low steering angles)
- If an error is detected MOVE will be directly switched off and a warning will inform the driver.
- There is an emergency button which allows the user to disconnect the external network from the internal one

3.1.4 Mobileye C2-270

The Mobileye is described on the user manual as a "single-camera-based safety solution for collision prevention and mitigation". In our application, it acts as a multipurpose sensor which detects the car distance from the lanes, and information about lane curvature. Mobileye does not guarantee 100% accuracy in the detection driving lanes, but it is accurate enough to start the development of the system, while the "vision algorithm" is being developed and as a redundancy in case of system failure.

FIGURE 3.2: Lane Detection by Mobileye [12]

3.2 Communication strategies

All electronic embedded systems used to control vehicle functions (specifically ADAS) need communications networks and protocols to manage all the process information [3]. Generally, these vehicle control functions receive the input information from a network of sensors (camera, wheel speed sensor etc.) and send commands to the control stage or the application software which further interacts with actuators or HMI elements to execute commands. Aspects of communication networks like assurance of message delivery, safety and security of messages, time of delivery, cost, EMF noise resilience, overall robustness and reliability and routing therefore become an important criteria while choosing the right communication network and protocol. Depending on different requirements of complex automotive functions, many communications networks and protocols are developed. The Society for Automotive Engineers (SAE) defined a classification for automotive communication protocols based on data transmission speed and functions those are distributed over the networks. This classification can be referred in [8]. Some of the most commonly used communication protocols and standards in now a day vehicles being CAN, LIN, FlexRay, MOST, SAE J1939 [3]. Conventional computer networking technologies (such as Ethernet and TCP/IP) are rarely used [17]. This section explains in short the widely used CAN communication. It also elaborates on Ethernet and Mobile Industry Processor Interface (MIPI). Table 3.1 further provides the differences between different vehicle communication networks.

3.2.1 CAN

One of the first and most enduring control networks, the CAN bus, is the most widely used network [10]. CAN became an ISO standard in 1994 and is now a defacto industrial standard in Europe for data transmission in automotive applications due to its low cost, its robustness and the bounded communication delays [8]. CAN is a multi-master serial bus standard for connecting Electronic Control Units [ECUs] also known as nodes [15]. Each node requires a Central processing unit (microprocessor, or host processor), CAN controller and a Transceiver. The host processor decides what the received messages mean and what messages it wants to transmit. CAN controller as shown in Figure 3.3, is often a part of microcontroller which stores the received serial bits from the bus until an entire message is available. It also transmits the bits serially onto the bus when the bus is free. The transceiver mainly converts the data stream from CAN bus levels to levels that the CAN controller uses and vice-versa.

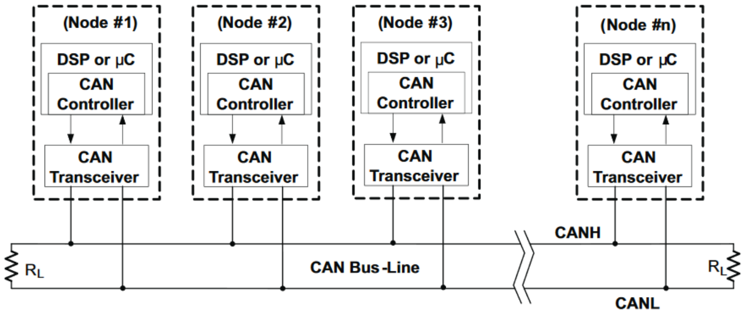

FIGURE 3.3: Details of a CAN bus[13]

Typically the CAN runs at speeds with 125 Kbps (low speed CAN with applications like body control electronics like seat, window etc) and with 500 Kbps (high speed CAN with applications like chassis control systems, lane keep systems). The typical message lengths are message lengths (CAN frames) are typically 50 to 100 bits in length. In CAN, any node may start a transmission when the bus is idle. To guarantee that the transmission of top priority message always remains first, CAN uses decentralized, reliable, priority driven CSMA/CD (Carrier Sense multiple Assess/ Collision Avoidance Detection) control method. CSMA means that each node on a bus must wait for a prescribed period of inactivity before attempting to send the message. CD means that the collisions of the messages are resolved by bit wise arbitration based on pre-programmed priority. The higher priority always wins the bus access. CAN offers an error detection mechanism which detects data transfer errors, interrupts and erroneous transmission with an error flag that indicates the re-transmission of the affected message. It also contains a mechanism for automatic fault localization including disconnection of the faulty controller. More explanation about CAN can be found in [13]. It must be also noted that CAN is a low-level protocol and does not support any security features intrinsically. There is also no encryption in standard CAN implementations, which leaves these networks open to man-in-the-middle packet interception. In most implementations, applications are expected to deploy their own security mechanisms [15].

3.2.2 Ethernet

Although Ethernet being rarely used in cars, the first motivation for Ethernet is that is a low cost and a mature technology that offers much more bandwidth that is available today. This is generally of interest for infotainment and active safety systems [8]. The original 10BASE5 Ethernet uses coaxial cable as a shared medium, while the newer Ethernet variants use twisted pair and fiber optic links in conjunction with hubs or switches [16]. Over the course of its history, Ethernet data transfer rates have been increased from the original 2.94 megabits per second (Mbit/s) to the latest 100 gigabits per second (Gbit/s). Systems communicating over Ethernet divide a stream of data into shorter pieces called frames which are transferred to nodes on various parts of the network. Each frame contains source and destination addresses, and error-checking data so that damaged frames can be detected and discarded; most often, higher-layer protocols trigger re-transmission of lost frames. Also like the CAN bus, Ethernet is bidirectional, and the speed possible on any individual link decreases as the number of nodes on the system increases. Still, Ethernet can transport data over a link 100 times faster than a CAN bus [6]. Ethernet would be an ideal choice to replace the CAN bus, but Ethernet's cost per node is higher. Hence, it can be used to augment the CAN bus like for mid-bandwidth communications rather than to replace [6]. Although it can be used for data-intensive requirements like cameras and infotainment systems the use is limited as it comes with greater complexity. This is because, typically the video needs to be compressed and decompressed at source and destination to avoid exceeding the bandwidth of Ethernet. However to fully meet the automotive requirements, multiple new specifications and revisions to specification are being done in the IEEE 802.3 and 802.1 groups focusing EMF/RFI emission susceptibility, latency requirement, synchronization and network management requirements [5].

3.2.3 MIPI: CSI-2

The bandwidths of today's host processor-to-camera sensor interfaces are being pushed to their limits by the demand for higher image resolution, greater color depth and faster frame rates [1]. But more bandwidth is simply not enough with performance targets that are also being pushed to their limits for the complex embedded systems like camera based ADAS applications. It is needed to have a standard, robust, scalable, low-power, high-speed, cost-effective camera interface that supports a wide range of imaging solutions. The Mobile Industry Processor Interface (MIPI) Alliance is a global, open membership organization that develops interface specifications for the mobile ecosystem including mobile-influenced industries. The Camera Serial Interface (CSI) is a specification of the (MIPI) Alliance. It defines an interface between a camera and a host processor. CSI-1 was the original standard MIPI interface for cameras. It emerged as an architecture to define the interface between a camera and a host processor [14]. The MIPI CSI-2 and MIPI CSI-3 are the successors of the original MIPI camera interface standard, and both these standards are continue to evolve. Latest MIPI CSI-2 was released in September 2014 and it offers higher interface bandwidth and greater channel layout flexibility than its predecessor. It also supports packetized transmission which helps in line management, error detection and error correction. More details can be found in [1].

Comparison table : different bus network Below is the comparison between different automotive communication network.

TABLE 3.1: Comparison table : different bus network [3], [8], [10]

Bus	CAN	LIN	FlexRay	MOST	Ethernet
Speed	Up to 1Mbps	20 kbps	10 Mbps	Upto 23 Mbps	Upto Gbps
Cable type	Twisted pair	Single wire	Optical Fiber Dual-Wire	Fiber Optic/ Coax	One or more twisted pair
Cost	$$	$	$$$	$$$$	$$
applications	Soft real time	Low cost low speed	Hard real time	Multimedia	Camera systems
application examples	Power train, engine control, Driving assistants	Elelectric seats, power window, rain sensor	Break-by-wire, Steer-by-wire, Emergency systems	Infotainment, navigation	IP Cameras, Infotainment
Control	Multi-master	Single-master	Multi-master	Time-master	Multi-master
Access control	CSMA/CDA	Polling	TDMA FTDMA	TDMA CSMA/CA	CSMA/CDA
Error Detection	CRC Parity bits	Checksum Parity bits	CRC Bus Guardian	CRC System - Service	CRC
Redundancy	None	None	Yes (2 channels)	None	None

3.3 Architecture Proposals

In this section, the hardware architecture for the ALC system is discussed. The original architecture, as can be seen in Figure 3.4 shows all the hardware components which are available for the implementation of ALC system. However not all these components are relevant for the ALC system and therefore only a selection of these components will be used during the next phase of implementation.

3.3.1 Proposed Architecture

The proposed architecture shows the layout of how all the hardware components will be connected when the ALC system is being developed and tested in real time. The proposed architecture aims to implement the concept architecture in the most compact way possible. The Bluebox has the capability to process sensor data from different sources and decide control actions for the vehicle in a particular scenario. However the 'number cruncher' part of the Bluebox is still unavailable and therefore a real time target computer will be used during the next phase. Figure 3.5 shows the layout of components in proposed architecture.

3.3.2 Implemented Architecture

The implemented architecture in Figure 3.6 shows the layout of how all the hardware components will be connected when the ALC system is being developed and tested in real time during the implementation phase of the project. The vision part of the

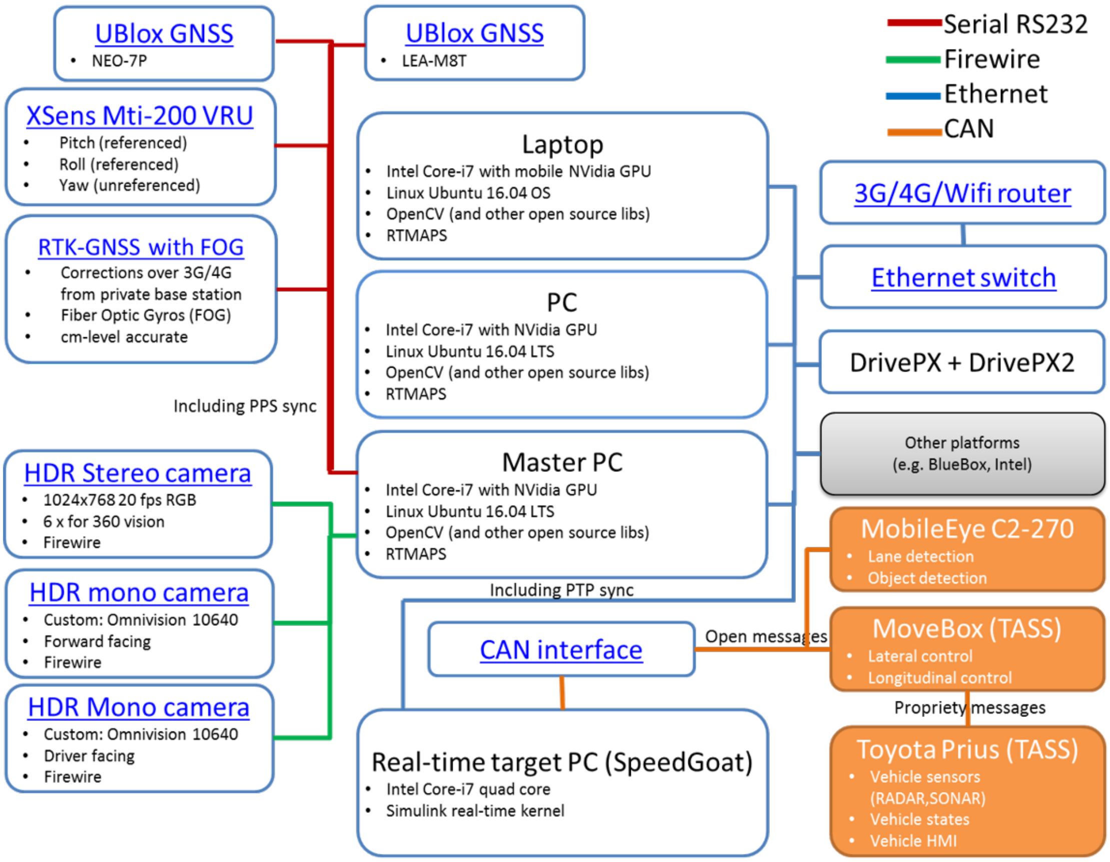

FIGURE 3.4: Available Hardware

functional architecture i.e. lane detection will be executed on the Bluebox and the high level, low level control and actuation will be executed on the speedgoat real time target computer. The communication between them is via Ethernet. An alternative solution will be where the Bluebox communicates directly with vehicle via CAN bus which is shown by a dotted line in the diagram.

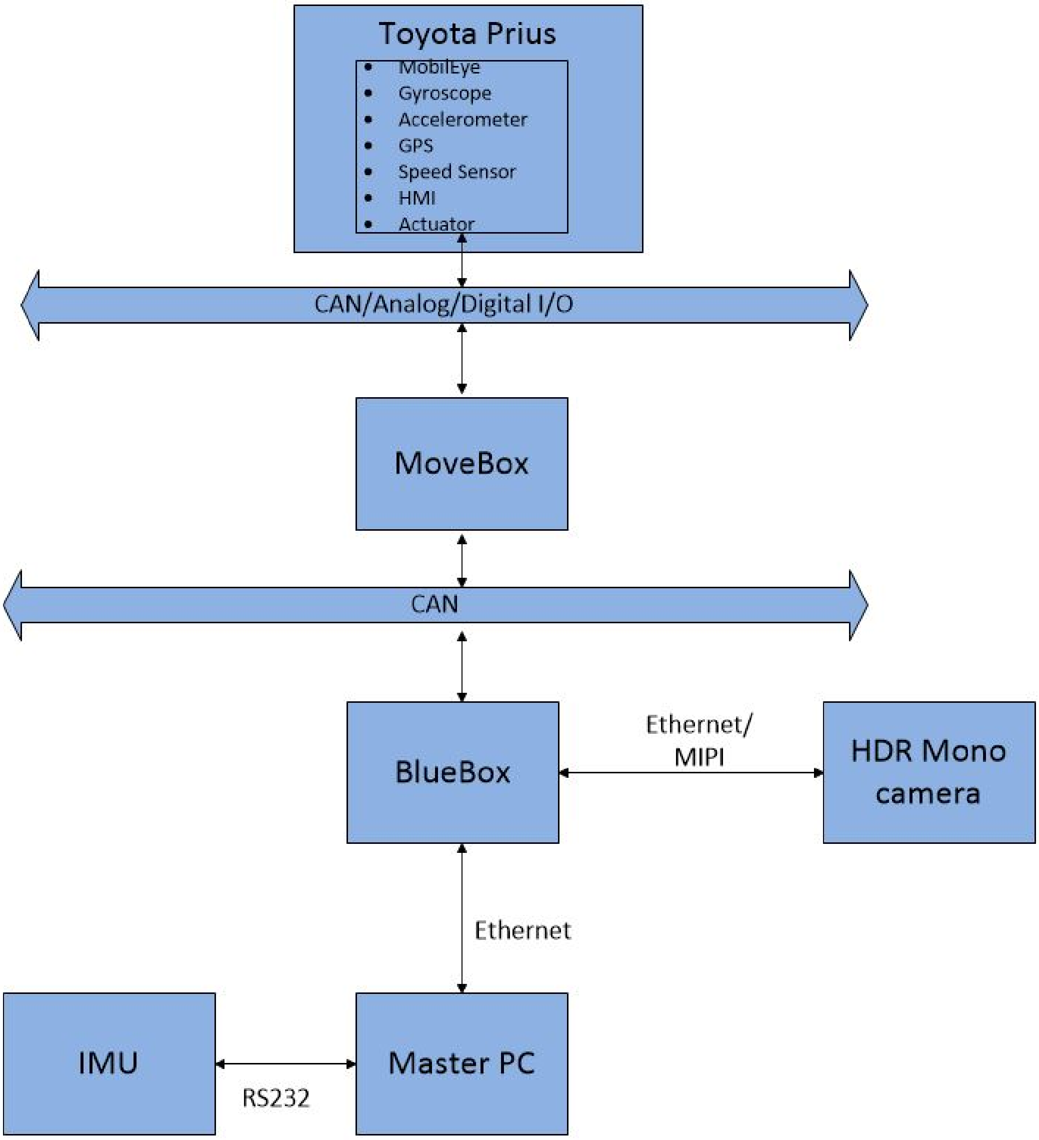

FIGURE 3.5: Proposed Hardware Architecture

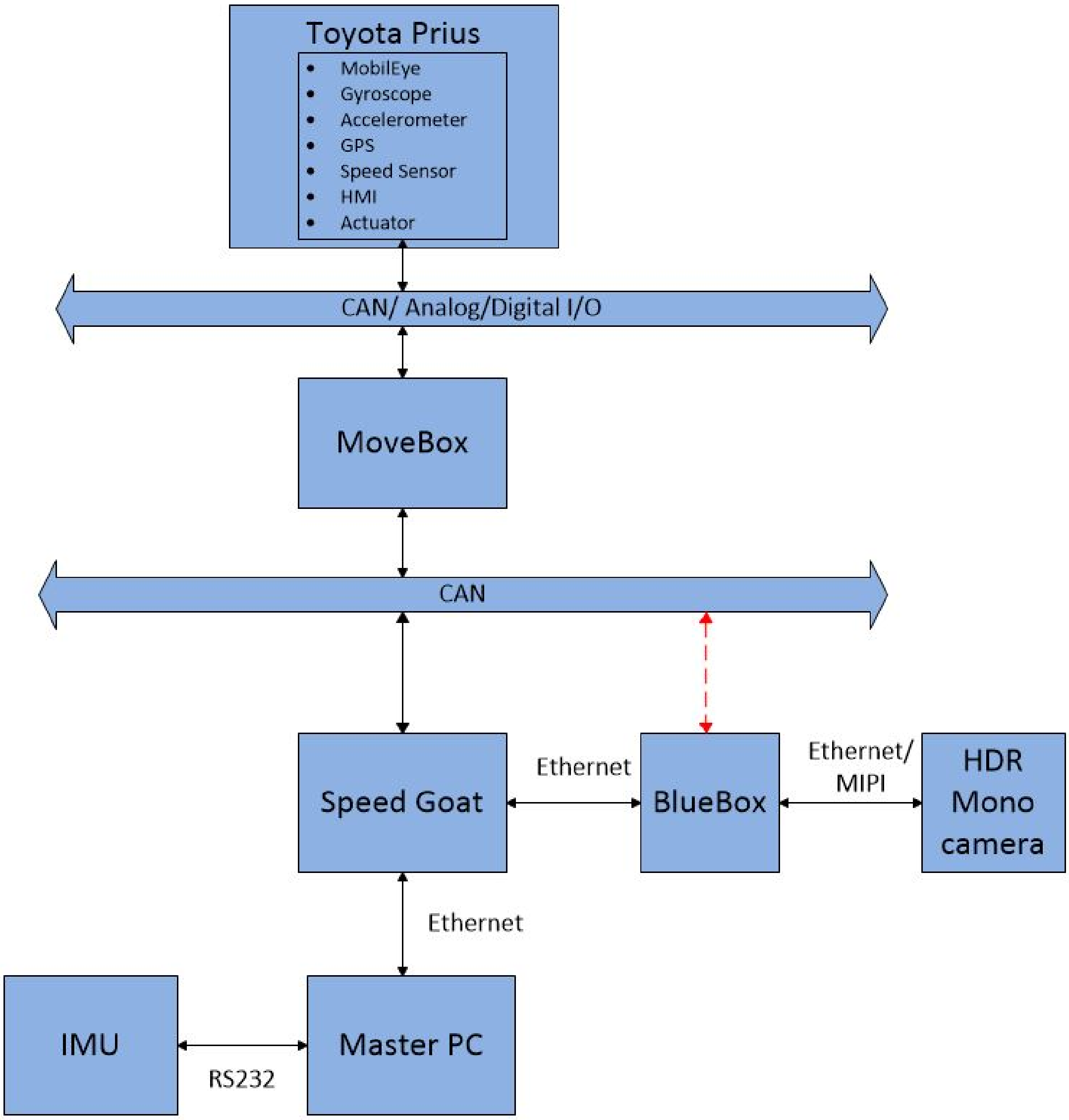

FIGURE 3.6: Implemented Hardware Architecture

Chapter 4

Conclusion and Future Work

In this report, the CAFCR views of the system are explained. The functional and non functional requirements for the ALC system are derived using this CAFCR approach. These requirements have been listed in Chapter 2. The components which will be used during the implementation course of the project are described briefly in Chapter 3. These components include TNO Movebox, NXP Bluebox, MobilEye and Speedgoat real time target computer. The communication strategies which can be selected during the development procedure are also described and compared. Based on the information about the components, two solutions are proposed for the hardware architecture. The first solution is a proposed solution where all computation both related to vision and control are executed on the NXP Bluebox. The second solution proposes to execute lane detection part of the system on the Bluebox vision processor whereas the control part will be executed on the Speedgoat. There is an alternative to the second solution where the communication between the Speedgoat and Bluebox is eliminated by facilitating a direct CAN communication between the Movebox and Bluebox for exchange of vehicle data.

Based on the findings during the second phase of the project, the second solution for hardware architecture is chosen for implementation. During the next phase, both high level and low level control solution for the system will be implemented on the Speedgoat real time target computer using Simulink realtime package. For the vision part of the system, initially MobilEye will be used for the lane detections in the control algorithm. However the system will be designed so that input from the MobilEye can be replaced by the Bluebox lane detection algorithm in the future when it is available.

Bibliography

[1] mipi alliance. "http://mipi.org/specifications/camera-interface". In: *Camera Interface Specifications*.

[2] Raymond Wouters Sharad Bhadgaonkar ASD TU Eindhoven Anshuman Singh Davide Occello. "Active Lane Centering System". In: *Module -I: Technical Report*. January 2017.

[3] David Daurenjou Frank Badstübner Martin Kunert. "DESERVE: Development Platform for Safe and Efficient Driving: ADAS development platform". In: *Standard interfaces definition – 2nd Release*. September 2012.

[4] ASD TU Eindhoven Ger Schoeber. "Systems Architecting". In: November 2016.

[5] ixia. "Automotive Ethernet: An Overview". In: May 2014.

[6] Texas Instruments Mark Sauerwald. "CAN bus, Ethernet, or FPD-Link: Which is best for automotive communications". In: *High-Performance Analog Products: Analog Applications Journal*. October 2014.

[7] Euro NCAP. "TEST PROTOCOL – Lane Support Systems. Version 1.0". In: *EUROPEAN NEW CAR ASSESSMENT PROGRAMME*. November 2015.

[8] Françoise Simonot-Lion Nicolas Navet. "In-vehicle communication networks - a historical perspective and review". In: August, 2013.

[9] F.E.B. Ophelders. "TNO Report". In: *MOVE User Manual*. 2014.

[10] Karen Parnell. "Put the Right Bus in Your Car". In: 2004.

[11] Ian Riches. "Strategy Analytics: Automotive Ethernet: Market Growth Outlook." In: *Keynote Speech 2014 IEEE SA - Ethernet and IP at Automotive Technology Day*. 2014.

[12] Gideon P Stein, Ofer Mano, and Amnon Shashua. "Vision-based ACC with a single camera: bounds on range and range rate accuracy". In: *Intelligent vehicles symposium, 2003. Proceedings. IEEE*. IEEE. 2003, pp. 120–125.

[13] Texas Instruments Steve Corrigan. "Introduction to the Controller Area Network (CAN)". In: *Application Report*. July 2008.

[14] Wikipedia. "https://en.wikipedia.org/wiki/Camera_Serial_Interface". In: *Camera Serial Interface*.

[15] Wikipedia. "https://en.wikipedia.org/wiki/CAN_bus". In: *CAN bus*.

[16] Wikipedia. "https://en.wikipedia.org/wiki/Ethernet". In: *Ethernet*.

[17] Wikipedia. "https://en.wikipedia.org/wiki/Vehicle_bus". In: *Vehicle bus*.

Appendix A

Hardware Specifications

A.1 Speedgoat

I/O Modules:

- Analog I/O: AD, DA, DMA, 16-24 bit, with optional configurable FPGAs
- Digital: TTL, RS422, LVDS, MGTs, with optional configurable FPGAs
- PWM generation and capture
- Encoder measurement and simulation (quadrature, SSI, SSI2, EnDat 2.2, BiSS)
- Temperature measurement and simulation: thermocouples, RTD (PT100/PT1000), NTCs
- Strain gauge and pressure measurement and simulation
- IEPE/ICP vibration measurements
- Resistor, potentiometer, and reed relays (SPDT, DPST, SPST)
- LVDT, RVDT, Synchro, and Resolver measurement and simulation
- Cam and crank measurement and emulation
- Shared/Reflective memory (ScramNet, PIORC5565)
- Fault/signal insertion switches

Communication Protocols:

- CAN, CANopen, LIN, SAE J1939
- Serial UART (RS232, RS422, RS485), and SDLC/HDLC
- SPI Master and SPI Slave
- I2C Master and I2C Slave
- Real-Time UDP, and Real-Time Raw Ethernet
- EtherCAT Master and EtherCAT Slave
- EtherNet/IP Scanner (Master) and EtherNet/IP Adapter (Slave)
- PROFIBUS Master and PROFIBUS Slave

- PROFINET Master and PROFINET Slave
- Ethernet POWERLINK Master and Ethernet POWERLINK Slave
- Modbus TCP and Modbus RTU
- XCP Master and XCP Slave over CAN and Ethernet
- FlexRay
- MIL-STD-1553 and ARINC-429
- IRIG and Precision Time Protocol (PTP), IEEE 1588
- USB WebCam and CameraLink

Hardware Specifications:

Housing:

- Enclosure: 4U 19"-compatible aluminium chassis
- Color: Silver powder-coated, natural aluminium
- External dimensions:
  - Height: 177.8mm (4U)
  - Width: 440mm, 480mm (including rack mounts)
  - Depth (standard): 360mm (400mm including handles) Depth (deep option): 440mm (480mm including handles)
- Weight: 12kg (excluding I/O modules, cables, and terminal boards)
- Power supply: 400W, 100-240V, 50-60Hz, fan-less, zero-noise
- Fans: Two at rear (outtake), high quality, low-noise Papst fans
- Handles: 2 for desktop use , 2 for rack installation
- Certification: CE and FCC certified

Mainboard and CPU:

- Processor: Intel Core i3 3.5GHz
- Form factor: ATX
- Chipset: Intel C216
- Bus: PCI, 32-bit/33MHz
- Memory: 4096MB DDR3 RAM
- Graphics: Intel HD Graphics 400P onboard
- USB: 4 x USB 3.0 and 1 x USB 2.0 at front, 6 x USB 2.0 internal

- Ethernet: 2 x Gigabit at front
- Serial Ports (for baud rates up to 115kb/s only): 1 x RS232/422/485 at front,1 x RS232/422/485 and 4 x RS232 internal
- Keyboard and mouse: 1 x PS/2 at front
- BIOS: American Megatrend Inc. (AMI)
- Number of slots for I/O modules: 3 PCI, 4 x PCIe and 1 x Mini PCIe

Drives:

- Main Drive: 1 x 60GB SSD

Power:

- Power inlet: AC 100-240V, 50/60Hz, at rear
- Power switch at rear
- Secondary power switch at front
- Reset button none (secondary power switch)
- Power LED at front (combined with secondary power switch)

Environment:

- Temperature: 0 deg to +60 deg (operating)
- Humidity: 10-90 %, non-condensing

Software:

- OS / RTOS: FreeDOS / Simulink Real-Time kernel
- Development computer: Utilities for kernel transfer, I/O drivers and Simulink test models

A.2 NXP Bluebox

Hardware Specifications:

Vision Processor : S32V234

- 4 x ARM Cortex- A53 CPU, up to 1.0 GHz core speed
- ARM Cortex-M4 at 133 MHz for IO control and AutoSAR OS
- Accelerators: Dual APEX-2 image processing engine, ISP, 3D GPU
- Main Memory: 2 x 2 GB LPDDR2
- 1 x NOR flash min. 512 Mbit – Hyperflash
- PCI Express: 1 x PCIe Gen2

- Ethernet: 1 x GB Ethernet
- Additional I/O Interfaces: VIU, SDHC, UART, HDMI, FlexRay, FlexCAN, SIPI, LIN, Gyroscope + Digital compass, JTAG and Trace port

Embedded compute processor: LS2085A

- LS2085A CPU: up to 8 x ARM Cortex- A57 CPU, up to 2.0 GHz core speed
- Accelerators: Advanced I/O Processor, Security Engine
- Main Memory: 2 x 72b (ECC) DDR controllers, up to 2.1 GT/s
- 128 MB NOR flash, 8-bit 2 GB SLC NAND flash
- PCI Express
- Ethernet
- USB 3.0
- SATA

Appendix B

Appendix B: Glossary

```
ALC        : Active Lane Centering
LDW        : Lane Departure Warning
Euro NCAP  : European New Car Assessment Programme
ASIL       : Automotive Safety Integrity Level
LSS        : Lateral Support System
HMI        : Human Machine Interaction
```