UNIVERSITY
OF WEST BOHEMIA

Faculty of Electrical Engineering
Department of Electronics and Information Technology

Doctoral Thesis

Electronics Control Systems and Advanced Methods of Evaluation for Automotive Applications

Autonomous Robotic Platform for ADAS Systems Testing

Elektronické řídicí systémy a pokročilé metody vyhodnocování využitelné v dopravních aplikacích

Autonomní robotická platforma pro testování ADAS systémů

Author: Ing. Ondřej Lufinka

Pilsen 2021

UNIVERSITY
OF WEST BOHEMIA

Faculty of Electrical Engineering
Department of Electronics and Information Technology

Doctoral Thesis

to obtain the academic title of Ph.D. in the field

Electronics

Supervisor: doc. Ing. Jiří Skála, Ph.D.

Pilsen 2021

Declaration

Hereby I declare, that I compiled my doctoral thesis independently under the supervision of supervisor using only the literature and other information sources that are all quoted in the thesis and listed in the references at the end of paper. As the author of this doctoral thesis I then declare that I did not violate the copyright law of third parties. I am fully aware of the consequences of this violation.

I also declare that all the software used in this doctoral thesis is legal.

Prohlášení

Prohlašuji, že jsem svoji disertační práci vypracoval samostatně pod dohledem vedoucího práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené disertační práce dále prohlašuji, že jsem v souvislosti s vytvořením této práce neporušil autorská práva třetích osob, zejména nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních. Jsem si plně vědom následků tohoto porušení.

Také prohlašuji, že veškerý software použitý při řešení této disertační práce je legální.

In Pilsen on the 15th of September 2021

Ing. Ondřej Lufinka

. . . .

Signature

Acknowledgment

This thesis was made with the support of projects SGS-2015-002 ("Moderní metody řešení, návrhu a aplikace elektronických a komunikačních systémů"), SGS-2018-001 ("Výzkum a vývoj elektronických a komunikačních systémů ve vědeckých a inženýrských aplikacích") and SGS-2021-005 ("Výzkum, vývoj a implementace moderních elektronických a informačních systémů") in the Faculty of Electrical Engineering, University of West Bohemia, and further in cooperation with the company Valeo Autoklimatizace k.s. in Prague and its project Opportunities for Students – Valeo R&D Program.

I would like to thank to my supervisor doc. Ing. Jiří Skála, Ph.D. and expert consultant Ing. Kamil Kosturik, Ph.D. from the University of West Bohemia for their professional mentoring and advices. Also to the other project members from Valeo R&D student program without whose teamwork the project could not be possible. Last but not least, I would like to thank as well to my family and my partner Ivana for their permanent support.

Abstract

Lufinka, Ondřej. *Electronics Control Systems and Advanced Methods of Evaluation for Automotive Applications* [*Elektronické řídicí systémy a pokročilé metody vyhodnocování využitelné v dopravních aplikacích*]. Pilsen, 2021. Doctoral thesis (in English). University of West Bohemia. Faculty of Electrical Engineering. Department of Electronics and Information Technology. Supervisor: Jiří Skála

This doctoral thesis deals with the problem of the Advanced Driver Assistance Systems (ADAS) testing in the field of automotive. There already are different solutions of testing in development and lately robotic platforms are beginning to be used more and more widely. Autonomous Robotic Platform (ARP) prototype discussed in this paper is used to explore new ways of said platforms' design possibilities. Firstly, in its chapters the thesis focuses on the background research to get an inspiration in existing robotic platforms. Then it describes the prototype's concept, its principles and goals on how to improve the field of ADAS testing. The next chapters explain hardware and software propositions since this thesis is mainly focused on the electronics part of the project. They also discuss maintaining reliability and safety of a large embedded system design. Further, the paper talks about the key features that were explored during the prototype's development for the innovation of these platforms (e.g. modularity, cloud based communication, automatic safety mechanisms, omnidirectional movement, common and non-traditional sensors combined with Kalman filtering for localization, autonomous navigation possibilities or synchronization of more platforms and cars together). Last but not least, the future development possibilities of the project that could lead to different innovations in the field are proposed.

Keywords

ADAS Systems, Autonomous Robotic Platform, Embedded Systems, Hardware, Kalman Filter, Localization, Modularity, Multiple Robots Synchronization, Navigation, Omnidirectional Movement, Safety Mechanisms.

Abstrakt

Tato disertační práce se zabývá problémem testování systémů nazvaných Advanced Driver Assistance Systems (ADAS) v oblasti automotive. Existují různá řešení pro testování, která jsou již ve vývoji, a v poslední době se stále více uplatňují robotické platformy. Prototyp Autonomní Robotické Platformy (ARP) popisovaný v této práci do detailu prozkoumává nové možnosti jejich návrhu. Ve svých kapitolách se práce nejprve věnuje výzkumu existujících zařízení, kde získává inspiraci. Dále popisuje koncept vyvinutého prototypu, jeho principy a cíle pro vylepšení tohoto odvětví. Další kapitoly vysvětlují hardwarové a softwarové návrhy, jelikož tato práce se soustředí především na elektronickou stránku projektu. Je v nich také probíráno, jak navrhovat rozsáhlý embedded systém a zachovávat jeho spolehlivost a bezpečnost. Poté práce rozebírá klíčové vlastnosti, které byly využity ve vyvinutém prototypu pro inovaci těchto platforem (např. modulárnost, komunikace s využitím cloudu, automatické bezpečnostní mechanismy, všesměrový pohyb, využití běžně dostupných i netradičních senzorů v kombinaci s Kalmanovým filtrem pro lokalizaci, možnosti autonomní navigace nebo synchronizace více platforem a automobilů). V neposlední řadě jsou nastíněny možnosti budoucího vývoje tohoto projektu, jejichž implementace by mohla vést k dalším inovacím v tomto oboru.

Klíčová slova

ADAS systémy, Autonomní Robotická Platforma, Bezpečnostní mechanismy, Embedded systémy, Hardware, Kalmanův filtr, Lokalizace, Modulárnost, Navigace, Synchronizace více robotů, Všesměrový pohyb.

Table of Contents

List of Figures

List of Tables

List of Symbols and Abbreviations

ADAS	Advanced Driver Assistance Systems
ADC	Analog to Digital Converter
AEB	Automatic Emergency Braking
AES	Advanced Encryption Standard
ARP	Autonomous Robotic Platform used in automotive
BLDC	Brushless DC electric engine
BTN	Button
CAN	Controller Area Network
DGPS	Differential GPS
DMA	Direct Memory Access
EU	European Union
Euro NCAP	European New Car Assessment Programme
EVT	Euro NCAP Vehicle Target
GND	Ground signal
GPIO	General-Purpose Input/Output
GPS	Global Positioning System
GVT	Global Vehicle Target
HIL	Hardware-in-the-Loop
HW	Hardware
IIC	Inter-Integrated Circuit
IEEE	Institute of Electrical and Electronics Engineers
IMU	Inertial Measurement Unit
IoT	Internet of Things
IPxx	International Protection marking code
ISM	Industrial, Scientific and Medical radio bands
LCD	Liquid-Crystal Display
LED	Light-Emitting Diode
Li-Po	Lithium Polymer battery
MCU	Microcontroller Unit
NMEA	National Marine Electronics Association
P2P	Point-to-Point
PC	Personal Computer

PID	Proportional Integral Derivative controller
PPS	Pulse per Second
PWM	Pulse Width Modulation
RC	Radio Control
RF	Radio Frequency
RPM	Revolutions per Minute
RTC	Real-Time Clock
SI	International System of Units
SIL	Software-in-the-Loop
SW	Software
UART	Universal Asynchronous Receiver-Transmitter
USB	Universal Serial Bus
VBAT	Battery Voltage
VUT	Vehicle under Test
WDT	Watchdog Timer

List of Nomenclature and Subscripts

Nomenclature

t	Time [s]
Δt	Time change per one iteration [s]
a	Acceleration [m/s^2] (*)
v	Velocity [m/s; km/h] (*)
d	Distance [m] (*)
ω	Angular velocity [rad/s] (*)
Φ	Heading [rad] (*)
r	Radius of turning [m] (*)
α	Steering angle of the wheel [rad] (*)
T	Temperature [°C]
$\sigma(x)$	Standard deviation of x
$\mathbf{x}$	State vector
$\mathbf{u}$	Input/control vector
$\mathbf{w}$	Process noise vector
$\mathbf{z}$	Observation/measurement vector
$\mathbf{n}$	Observation/measurement noise vector with cov. matrix $\mathbf{Q}$
$\mathbf{F}$	State transition matrix with process noise cov. matrix $\mathbf{Q}$
$\mathbf{G}$	Input/control matrix
$\mathbf{H}$	Observation/measurement matrix
$\mathbf{K}$	Kalman gain matrix
$\mathbf{P}$	Covariance matrix of the state vector
$\mathbf{Q}$	Process noise covariance matrix
$\mathbf{R}$	Observation/measurement noise covariance matrix
$\mathbf{I}$	Identity matrix

*) Distance, speed and acceleration are changing to the positive values in the backward-to-forward longitudinal direction and in the left-to-right lateral direction. Angles and angular velocities are changing to the positive values in the clockwise direction. Heading is zero for the south-to-north direction.

Subscripts

k	At time k (in case of discrete time operations)
K	Number of samples (in case of discrete time operations)
x or X	In longitudinal direction of object's movement (see Fig. 3.1)
y or Y	In lateral direction of object's movement (see Fig. 3.1)
z or Z	In vertical direction of object's movement (see Fig. 3.1)
lat	In direction of Earth's latitude
$long$	In direction of Earth's longitude
$left$	Value for the wheels on the left side of ARP
$right$	Value for the wheels on the right side of ARP
φ	Rotation in roll manner (see Fig. 3.1)
θ	Rotation in pitch manner (see Fig. 3.1)
ψ	Rotation in yaw manner (see Fig. 3.1)
ref	Reference value
req	Requested value
$meas$	Measured value
$filt$	Filtered value

1

Introduction

Flat and overrunable Autonomous Robotic Platforms (ARP) are beginning to be used more and more widely in the field of automotive. Their main purpose is to test the Advanced Driver Assistance Systems (ADAS). There are many solutions that can be found among the ADAS systems; for example adaptive cruise control, automatic parking, blind spot monitor, emergency driver assistant, lane change assistance or pedestrian protection system. More information about these systems can be obtained from the articles [7], [19] and [22] in the references section. There are many ways to test them by using moving or stationary targets. These targets can be operated by different towing mechanisms, which work fine, but they are not very variable. On the other hand, if the ARP is used, it makes the testing much more autonomous and modifiable. Nevertheless, design of such an ARP brings many challenges to the research. The main goal of this thesis is to explore these problems and come out with different approaches for improving this field.

To use the ARP for ADAS testing, it needs to be equipped with an ADAS testing target (also called a dummy). The dummy can be a pedestrian, a cyclist, a balloon car or similar. The ARP must be able to carry it on its top and then the scenarios according to the Euro NCAP (European New Car Assessment Programme, [17]) standards can be performed. For example, there is a scenario to test the Automatic Emergency Braking (AEB) of a car where the platform moves with the dummy of a different car in front of a tested car and then suddenly brakes to a halt. The parameters such as deceleration are defined by these standards as well (Fig. 1.1). After that, the car's reaction is measured and evaluated again according to the Euro NCAP standards. One of the results can be assignment of the well-known Euro NCAP stars to the car that is being tested. It is not mandatory for automotive manufacturers to fulfil the Euro NCAP requirements, however many customers do take them into account. That makes these standards very important and therefore the testing too. The testing is necessary during the development of new features as well as to make the feature approved for the production. If the test is required to be officially accepted by the Euro NCAP standards, the presumption is that the ARP and the lab where the test is performed are homologated by these standards too.

There are several more or less autonomous robotic platforms in the development

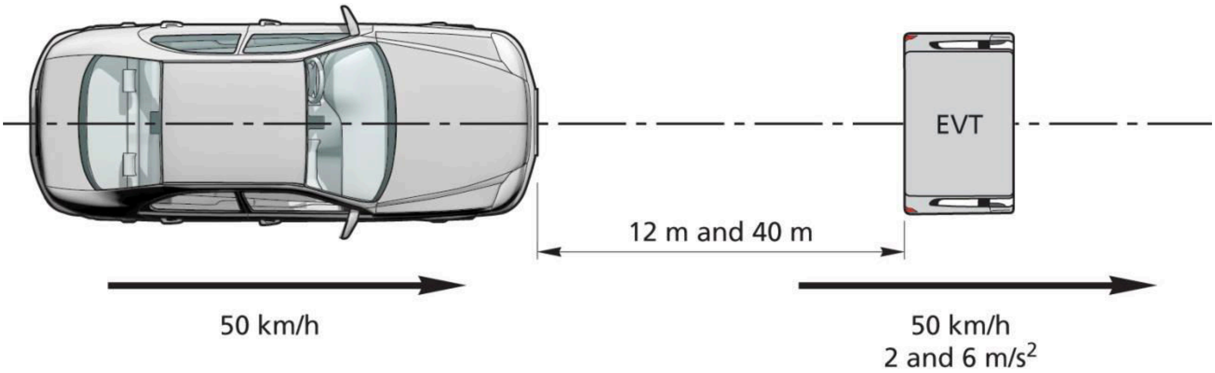


Fig. 1.1: Example of the AEB testing scenario (tested car on the left, EVT – Euro NCAP Vehicle Target – that can be moved by the ARP on the right) |Borrowed from [17]|

already. The goal of this doctoral thesis is to explore the design of such a large embedded system and the maintenance of its reliability and safety. Also it explores new possibilities that can be used to make the ADAS testing more automatic. Usually the ARPs are equipped with differential GPS to compute its position that makes the costs per device very high. Also the movement is available only in the car's manner (front steering axles and rear fixed axles with the drive). This new device uses common sensors for the localization and then performs the software filtering to make the position more precise. The good estimation of position is the key for autonomous control, i.e. navigation. There are also new ways of the movement (e.g. all driven and steering wheels – omnidirectional movement) to be explored. The platform is resistant against being driven over by car and must preserve low profile from the ground so it can be ignored by sensors used in automotive. Also, safety mechanisms are very important because the platform's weight can be over 100 kg and it can reach high speeds (up to 80 km/h). Modularity of the platform enables fast modifications, servicing, battery exchange, cleaning, etc. The development leads to the platform which is synchronized with the tested car and also the cooperation of more such platforms is discussed.

To make the platform available to certify the sensors used on the cars in automotive, it is required that the ARP is certified as well. The standards for these platforms are normalized by the Euro NCAP standards. The goal of this platform is internal company research therefore its certification is not required at the moment. Nonetheless, it is developed to meet these standards because the tests must be performed according to the requirements during the research as well. New features developed with the help of this ARP can be certified later with the official Euro NCAP testing device. Further, this ARP could be certified in the future and used to certify the ADAS systems officially because the standards are followed by its design.

The doctoral thesis is focused mainly on the electronics (embedded systems) part of the project. The whole design has been developed with the other team members in the company Valeo Autoklimatizace k.s. in Prague and its project Opportunities for Students – Valeo R&D Program. There is a master thesis that deals with the mechanical part of the design (see [26]). It was created at the Czech Technical University in Prague (CTU) by one of the team members and it inspired the mechanical design of this platform.

In its chapters, this paper first talks about the background research that can be found in the field. There are several devices for the ADAS testing that led to the inspiration for this project and they are mentioned there. Next, the whole concept of the current prototype is explained and the goals of the thesis are summarized. Two following chapters are focused on the embedded systems point of view (hardware and software) of the prototype. They use block diagrams and flowcharts to clearly explain the functionalities of the system's design to make it reliable and safe. In the next chapter, innovative features that were used in the current prototype are described in more detail to outline the project design capabilities. Those are the automatic safety mechanisms, the cloud based communication and the omnidirectional movement. Last two chapters deal with the problem of the ARP's localization and navigation. Different sensors are described and compared by the performed measurements and Kalman filtering is used to combine their outputs. Several ways of control are then introduced as well as the ARP's synchronization possibilities. Localization and autonomous navigation can make the platform fully automatic, which is the main goal of this research. Only a driver in the VUT (Vehicle under Test) is needed to operate the tests then. Further, multiple platforms synchronization is proposed to create more difficult scenarios. There is also an idea that even the VUT could be replaced by a platform. It would have a dummy of the same size and it would use the same sensors as a real-life car. That would lead to the full autonomy of the ADAS testing, which minimizes its overall costs and time requirements.

2

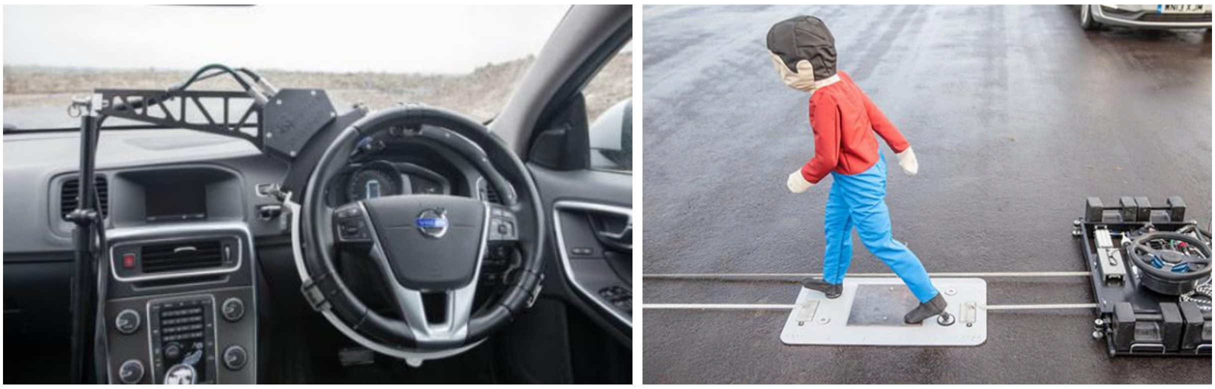
Background Research of ARP Devices

In this chapter, a research of existing platforms that are being developed is done. The fact that the platforms already exist and perform measurements of the ADAS systems does not mean that there is no space for improvement. The paper gets inspiration from the existing devices and in the next chapters it proposes such ideas to make the testing more autonomous and possibly more cost efficient due to the used technologies.

The introduction of the testing devices in development is listed here. More information about ADAS systems, scenarios, targets and testing platforms in general can be found later in the chapter 3. The possibilities of the ARP design are explained there too.

2.1 ABDynamics

ABDynamics is a British company founded in 1982 specializing in the ADAS testing. They provide wide range of products such as car driving robots, ADAS targets or positioning systems. In the figure 2.1, steering robot and soft pedestrian target are shown.



(a) Steering robot

(b) Soft pedestrian target

Fig. 2.1: Examples of the ABDynamics products |Borrowed from [5]|

They also develop a testing platform called LaunchPad which is designed to carry the ADAS targets and perform the defined scenarios. LaunchPad platform is capable of carrying dummies of pedestrians and cyclists, it can be driven over by car and it can follow predefined trajectories. It can go up to 50 km/h, the height is 65 mm and it can carry a payload at maximum of 15 kg. The platform is shown in the figure 2.2. More information about the products can be found in [5].

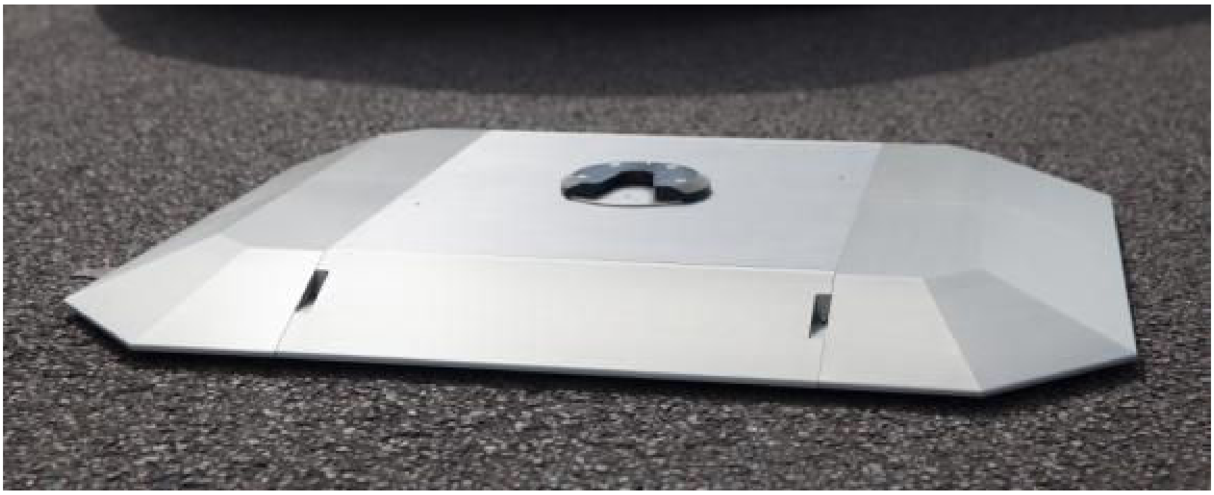


Fig. 2.2: LaunchPad platform |Borrowed from [5]|

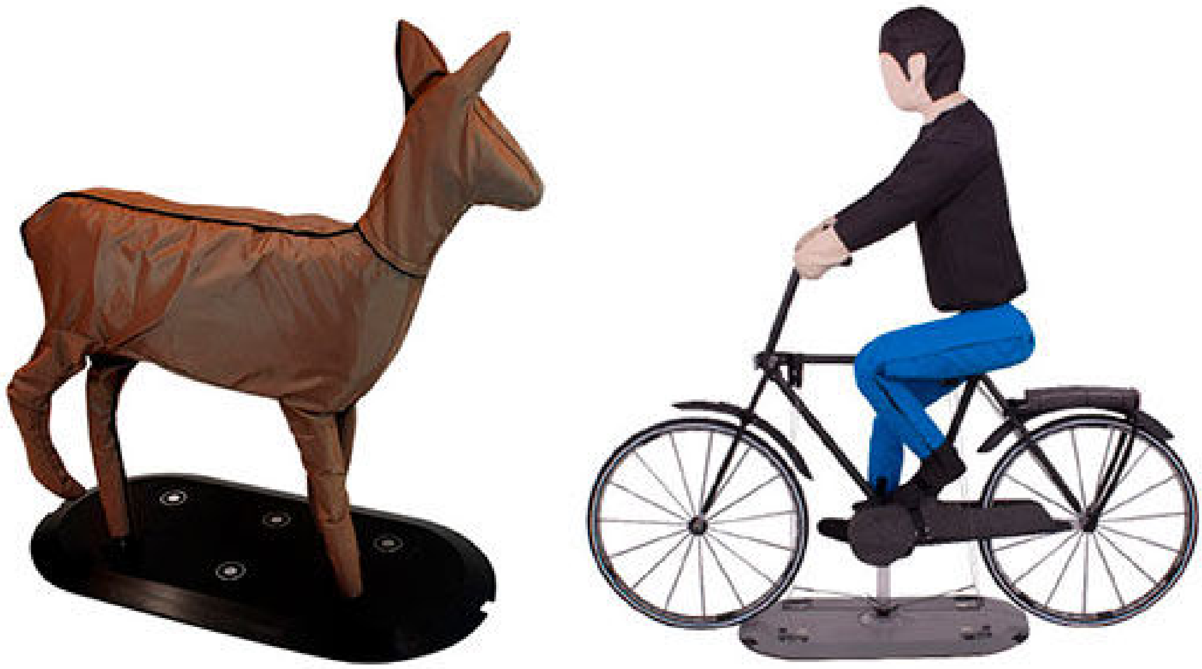
2.2 DSD

DSD (Dr. Steffan Datentechnik GmbH) company was founded in 1990 and it is situated in Austria. They focus on software for a car's accident reconstruction. In 1999, their product called UFO (Ultraflat Overrunable Robot for experimental ADAS Testing) was introduced and it developed into a platform capable of performing scenarios with car targets. The top speed limit is 80 km/h, it is 95 mm high and it can be loaded with 235 kg. More information is in [12].

2.3 4activeSystems

4activeSystems is a company located in Austria and it is a part of 4a Technology Group. They develop dummies and equipment for the ADAS testing. In the figure 2.3, there are examples of a cyclist and an animal dummy.

Their platform for ADAS testing is called 4activeFB. It is capable of going the speeds up to 50 km/h, its height is 85 mm. It focuses mainly on the AEB testing scenarios. It uses three driven wheels for motion, DGPS system for navigation and robust construction to withstand being driven over by car. It is waterproof as well. There are safety mechanisms to avoid system's failure (e.g. timeouts for data, emergency stop, double CPU). Its visualization is attached in the figure 2.4.



(a) Dummy of an animal

(b) Dummy of a cyclist

Fig. 2.3: Examples of the 4activeSystems products |Borrowed from [1]|

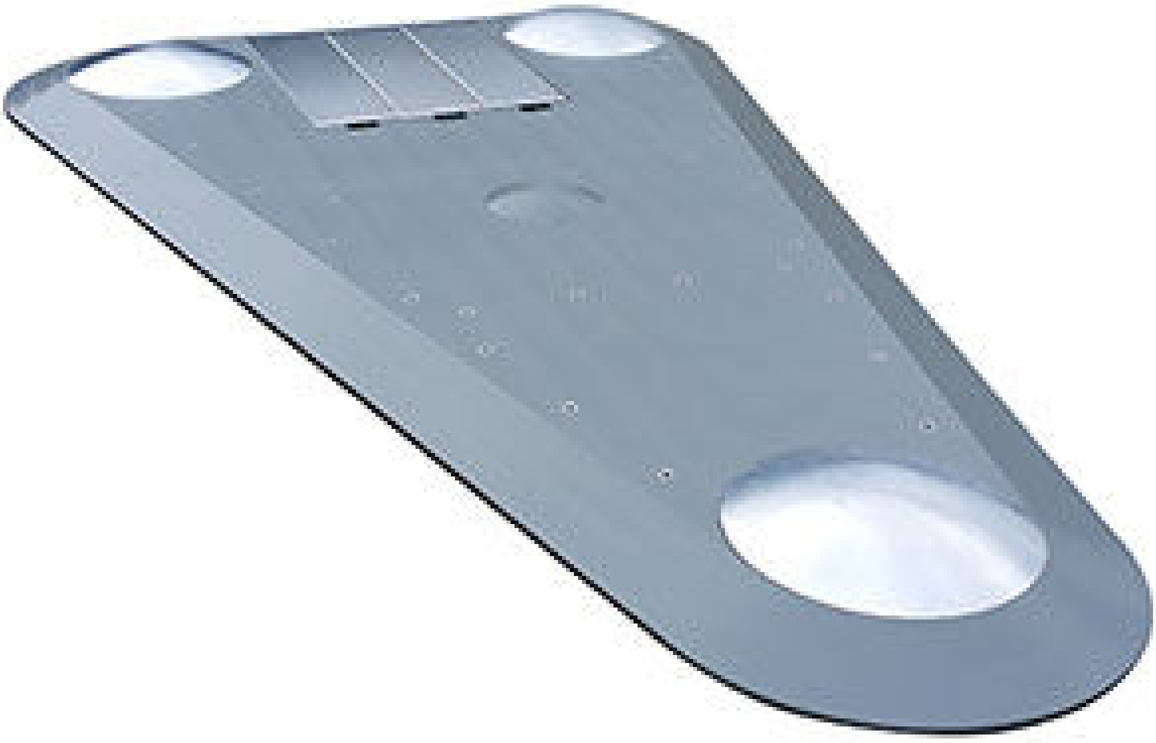


Fig. 2.4: 4activeFB platform |Borrowed from [1]|

The construction of the side ramps provides low radar cross section, which makes it less detectable by car's sensors. These parameters are repeated more or less with all the platforms by different manufacturers since all the ADAS tests must detect the target but not the platform itself. For more information about this device see [1].

2.4 Other Devices

There are other solutions that have been found during the background research of the field at the time of compiling this paper. They can be found in the references [10], [13], [27] and [37]. In general, these platforms are beginning to be used more and more widely. They are an interesting branch of automotive industry to explore since they are used in the research of fully autonomous cars, which is an attractive topic itself.

3

Concept of ARP and Goals of Doctoral Thesis

This chapter focuses on the concept of the ARP and then it summarizes goals of the thesis. It talks about the ARP in general and it explains ADAS systems and why their testing is necessary as well. Ideas that can be used to develop and improve these ARPs are discussed here too. Then the prototype concept that is currently under the development in the company Valeo Autoklimatizace k.s. (as mentioned in the introduction) is shown plus its embedded design is then explained in the chapters 4 and 5 in more detail. This prototype also helps to test new ideas that the thesis talks about later in the chapters 6, 7 and 8. These ideas are the main contribution of this research to the field of ADAS testing.

Mostly the devices described in the previous chapter (2) use differential GPS for localization so exploration of other conventional and also non-traditional sensors can be interesting area of research. Including the possibilities of navigation based on those sensors and comparing the measurements according to an achieved precision (how different sensors perform in comparison with DGPS, etc.). Also focusing on a movement with all four driven and steerable wheels is an interesting idea since the platforms usually move in the car's manner (this principle leads to an omnidirectional type movement). The research leads to the synchronization possibilities of more such platforms with the tested car as well. Last but not least, the safety mechanisms to ensure a prevention against failures are proposed.

3.1 ADAS Systems and Their Tests

Let's first focus on the ADAS systems that stand for Advanced Driver Assistance Systems. As mentioned in the introduction, literature sources that can be studied for more information are [7], [19], [22] and [28]. These systems are in general any electronic systems that are good for improving safety or driving comfort of a car. They are for example:

- Adaptive cruise control.

- Automatic parking.
- Blind spot monitor.
- Collision avoidance system.
- Lane change assistance.
- Pedestrian protection system.
- Rain sensor.
- Traffic sign recognition.

Automotive is a very special industry sector where the testing is highly important since any failure of implemented systems can lead to injuries of car's passengers. In order to obtain a homologation for a car, all the systems must be certified. That is possible through testing of all the features and overall design of the car. That includes ADAS systems, which is the focus of this thesis. The final testing in the real traffic to put a car to production is all about collecting huge amount of data and evaluating them. To make these tests as smooth as possible, it is convenient to test all the different features separately before these real traffic tests are performed. The reasons are that if a car's system fails during these final tests, it costs many resources to redesign plus these tests are very time demanding themselves. Therefore, at the beginning, hardware is usually tested through HIL (Hardware-in-the-Loop) simulations and software through SIL (Software-in-the-Loop) simulations. ADAS systems then need to be tested on the test tracks since they are more complex. This testing on a test track brings many more possibilities for automation than the real traffic tests. That makes them more cost and time efficient. That is the area where the ARPs can be very useful.

The way the ADAS systems are tested is described by the standards called Euro NCAP (European New Car Assessment Programme, [17]). The standards contain information about scenarios that must be performed for different ADAS systems and about dummy targets that must be used. That is important for this thesis because the developed ARP prototype must comply with them as well. Further, there are standards to evaluate these tests too. They are used for the overall car's Euro NCAP rating. That is important for a customer but not so much for this research, since the ARPs are performing the tests, not evaluating them. As said in the introduction, there are no mandatory requirements given by the law to fulfil these ratings, but their results are taken into account by experts as well as by public. And public is who buys the cars at the first place. As a result, these ratings are then very important for cars' manufactures and so is the testing of ADAS systems during their development.

3.1.1 Euro NCAP Scenarios

Testing of the ADAS systems on the test tracks is based on performing prescribed scenarios. These standards are described by Euro NCAP as mentioned before. There was one of the scenarios already briefly explained in the introduction and now, the other one is described in more detail. There is also paper [38] cited in the references that deals with scenarios design problems where more information about their preparation can be obtained.

To give an illustrated example of the ADAS scenario, the AEB car to car test is used. It is also the kind of scenario that can be tested with the ARP equipped with a dummy that is shown in the next section. The whole process is thoroughly explained in the document [16] and in the figures below (3.1, 3.2, 3.3 and 3.4), there is its brief description. Firstly, the coordinate system and notations are defined. They are also used throughout the whole thesis in the relevant equations and calculations and summarized in the nomenclature section. On the next figure, there is a representation of the lateral offset of VUT (Vehicle under Test) and EVT (Euro NCAP Vehicle Target). It is defined by a table for each kind of test and it must be kept in its range during the complete duration of the scenario. Its computation is done by the equation 3.1.

$$Lateral\ offset = Y_{VUT}\ error + Y_{EVT}\ error \tag{3.1}$$

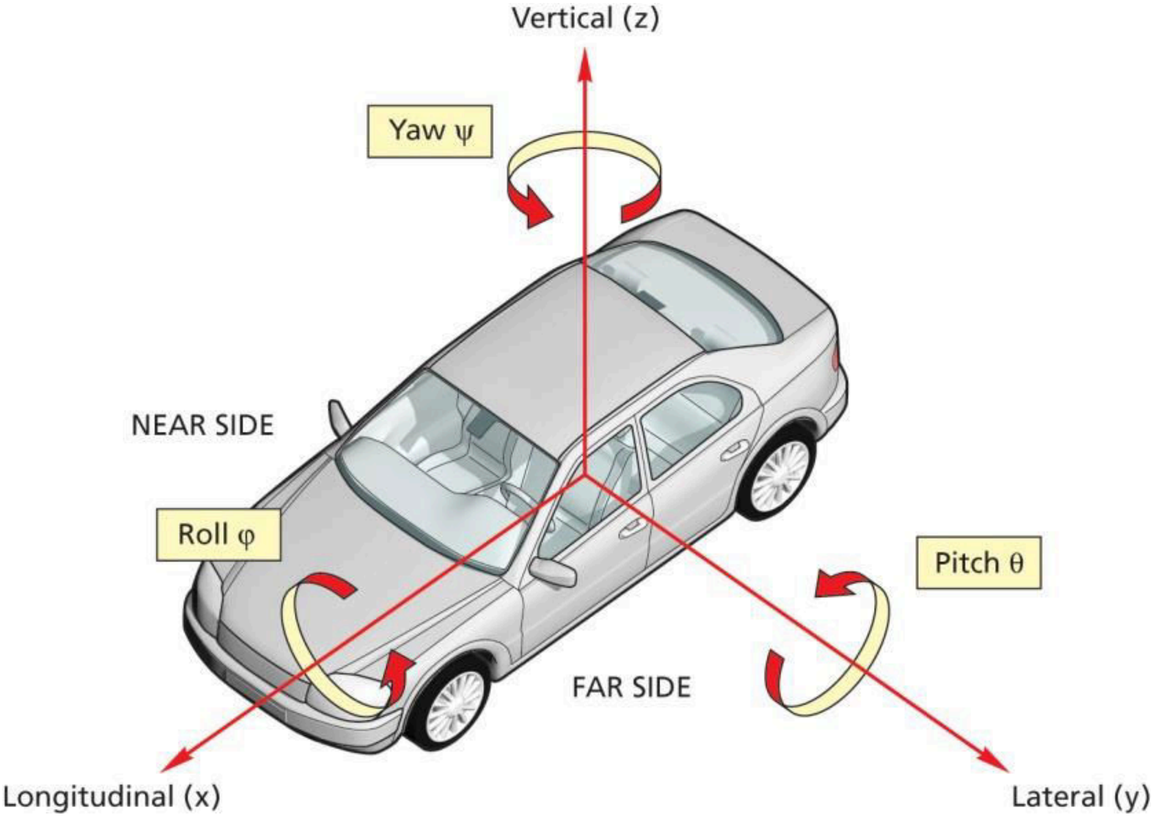


Fig. 3.1: Coordinate system and notation |Borrowed from [16]|

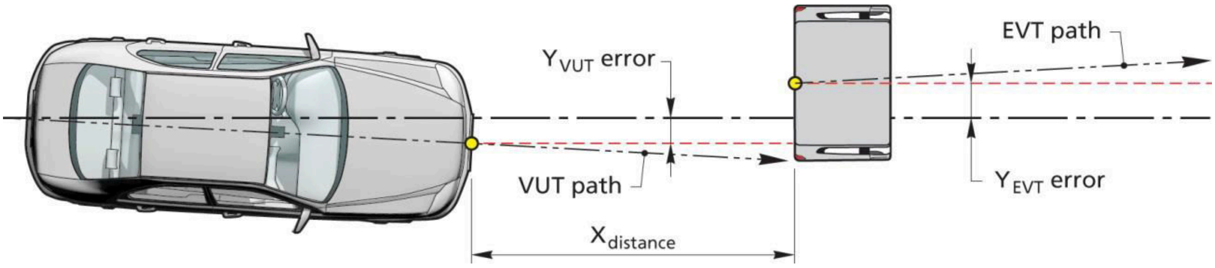


Fig. 3.2: Lateral offset |Borrowed from [16]|

In the following figures (3.3 and 3.4), there is the particular AEB scenario and the Euro NCAP target used exactly for this test. In this scenario, the VUT is moving at the speed of 30 to 80 km/h and the EVT at the speed of 20 km/h. The distance obviously closes if they are lined up. The evaluation of the test is the distance where the car automatically stops or the impact force of the crash (when it does not stop).

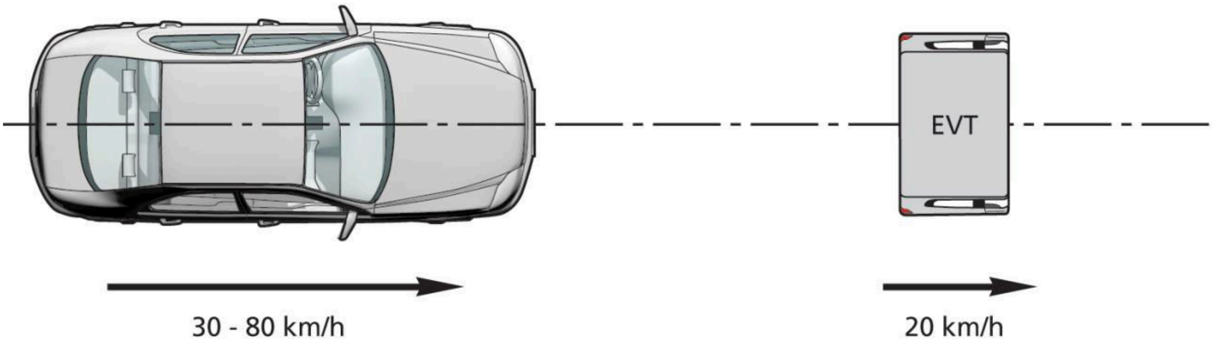


Fig. 3.3: Rear car-to-car moving scenario |Borrowed from [16]|

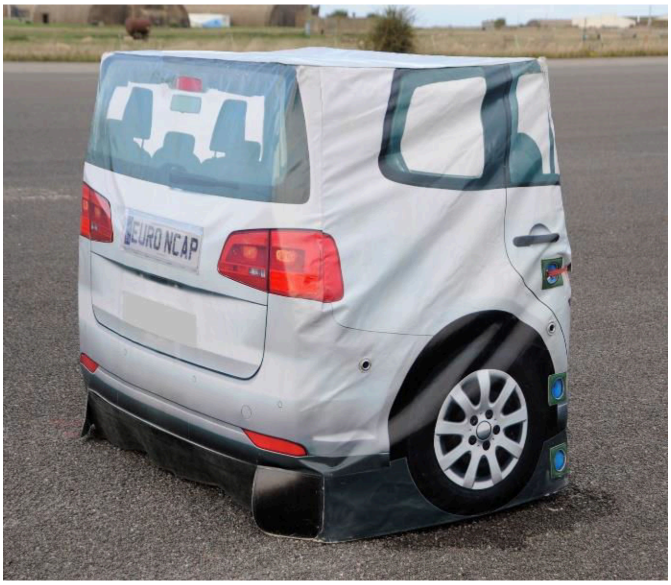
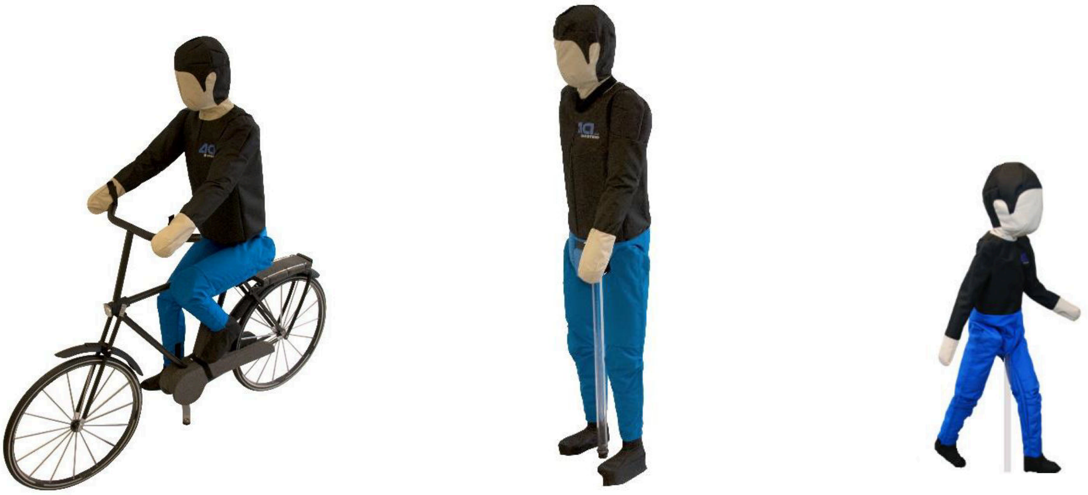


Fig. 3.4: Euro NCAP Vehicle Target (EVT) |Borrowed from [16]|

These values are again standardized by Euro NCAP and they result to the ratings for the tested car. The goal of the ARP, if it is used for the testing, is to simulate the EVT's movement at the desired trajectory, speed and deviation.

3.1.2 Euro NCAP Targets

There are several Euro NCAP targets (also called dummies) that can be used in the ADAS tests and they are shown here. Some of them could be already seen in the previous section during the AEB test description and in the chapter 2. Each ARP needs to be designed in the way that it can be equipped with these dummies. Examples of the dummies are shown in the figures 3.5 and 3.6. Difference between EVT and GVT is that EVT is a standard for European tests, whereas GVT can be used globally (different world areas can have different specifications for ADAS testing). This paper is focused on the EU region dummies.



(a) Cyclist (b) Pedestrian (c) Child pedestrian

Fig. 3.5: Examples of the Euro NCAP targets |Borrowed from [17]|

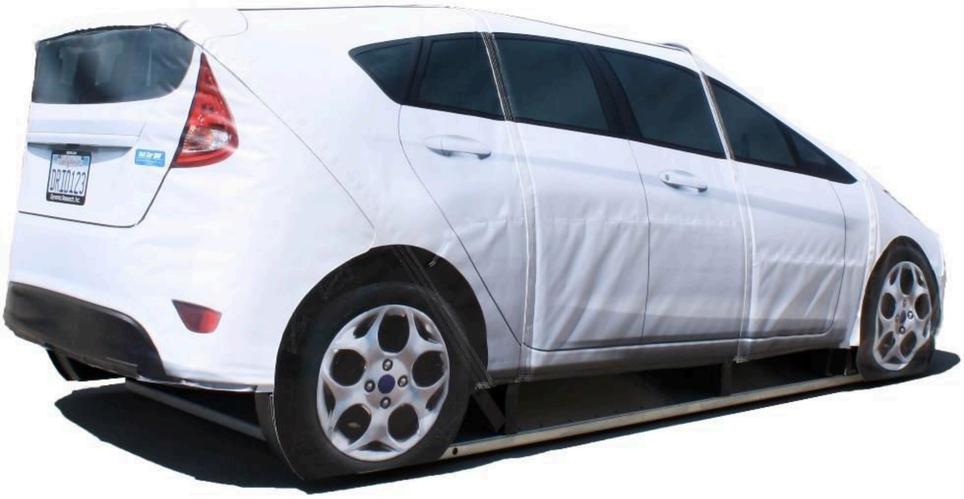


Fig. 3.6: Example of the Global Vehicle Target (GVT) |Borrowed from [17]|

3.2 Concept of ARP

Since the ADAS systems, their testing scenarios and Euro NCAP targets have been explained, this section can now talk about the ARP concept. In general, an autonomous robotic platform to test the ADAS system is a device that must be able to perform desired movement according to one of the Euro NCAP scenarios. Further, it must be capable of mounting testing targets (dummies) and it must withstand being driven over by car. The concept was also described in the author's paper [Pub. 5].

3.2.1 General Requirements

There is a concept of ARP that is currently under the development in the company Valeo Autoklimatizace k.s. as mentioned before. This concept is built according to the Euro NCAP standards because that is the beginning of any testing attempt in the automotive, nowadays. It does not seek to be Euro NCAP certified, since it is supposed to be used for internal company research, but its design can allow this certification in the future. It also takes inspiration in the devices described in the chapter 2. This device is used for implementation of new ideas for automated ADAS testing which is the core of this doctoral thesis. Furthermore, it is used for the ADAS testing itself.

At the beginning, the key constructional features must be taken into account. Any ARP that is going to be used for the ADAS testing must meet these requirements. They are shown in the table 3.1. These requirements come from the current Euro NCAP standards ([17]) and the future course is also taken into consideration (see the Euro NCAP 2025 Roadmap in [15]).

Tab. 3.1

Parameters	Values
Scenarios to perform	AEB cyclist AEB pedestrian – Back-over AEB – Junction & Crossing AEB – Head-on Adaptive cruise control Automatic Emergency Steering Blind spot monitor
Targets to carry	Pedestrian dummy Child pedestrian dummy Animal dummy Cyclist dummy Motorcyclist dummy EVT/GVT dummy

Table continues on the next page...

Tab. 3.1 – continuation

Parameters	Values
Durability	Withstand being driven over by car Dust resistance Water resistance
Power management	Cover whole day of testing ($\approx$ 8 h)
Weight to carry	Up to 100 kg
Overall height	10 cm (not to affect car's sensors)
Maximum speed	5 km/h (pedestrian scenarios) 25 km/h (cyclist scenarios) 80 km/h (car to car scenarios)
Acceleration	3 m/s^2
Deceleration	6 m/s^2

Tab. 3.1: Requirements for the ADAS testing platform

The following section of this chapter explains the currently developed concept. The next chapters are then talking about the prototype in more detail from the embedded systems point of view since it is the core of this thesis (hardware and software design) as well as about the innovative ideas that are explored. The prototype has been developed in cooperation with the Ph.D. students' team in the project Opportunities for Students – Valeo R&D Program. The mechanical part took inspiration in the master thesis [26] and is not further explained in this paper. It can be briefly seen in the figure 3.7 and also in the appendices A and B (figures A.1 and A.2 – 3D construction models, figures B.1, B.2, B.3 and B.4 – photos of the prototype).

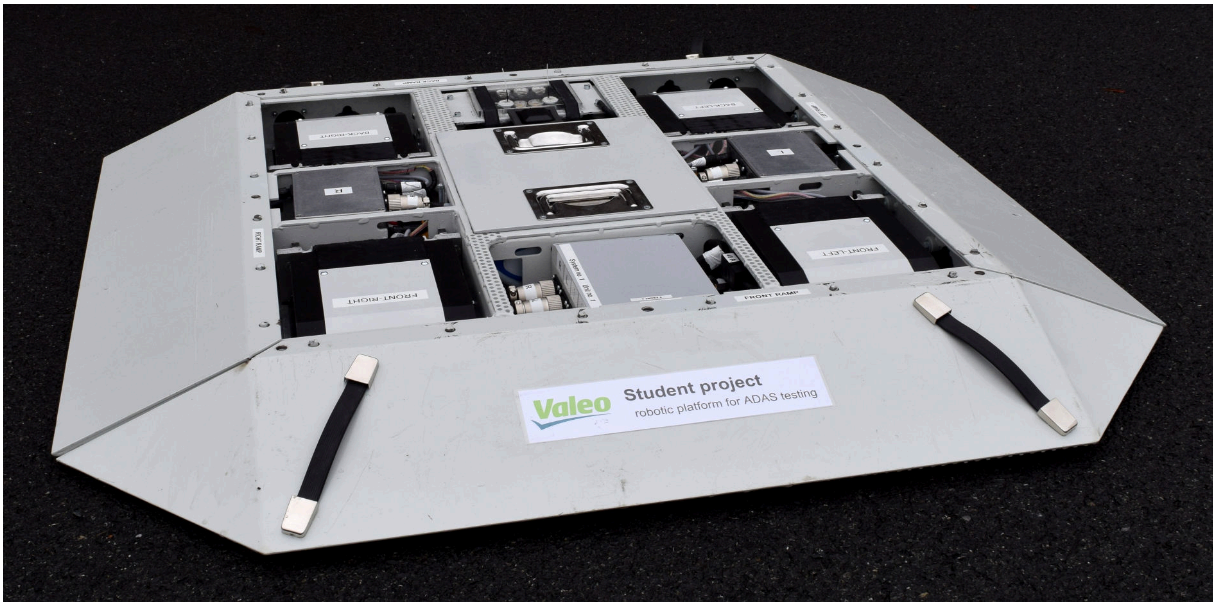


Fig. 3.7: Photo of the ARP prototype

3.2.2 Detailed Description

In this section there is a more detailed look into the current ARP prototype design. In the figure 3.8, there is a block diagram of the developed platform. At the first sight, it is obvious that the concept is modular from the mechanical as well as electrical perspective. That makes the design improvements easier (e.g. trying different wheels modules) or switching the modules faster (e.g. replacement of an empty battery or of a damaged module).

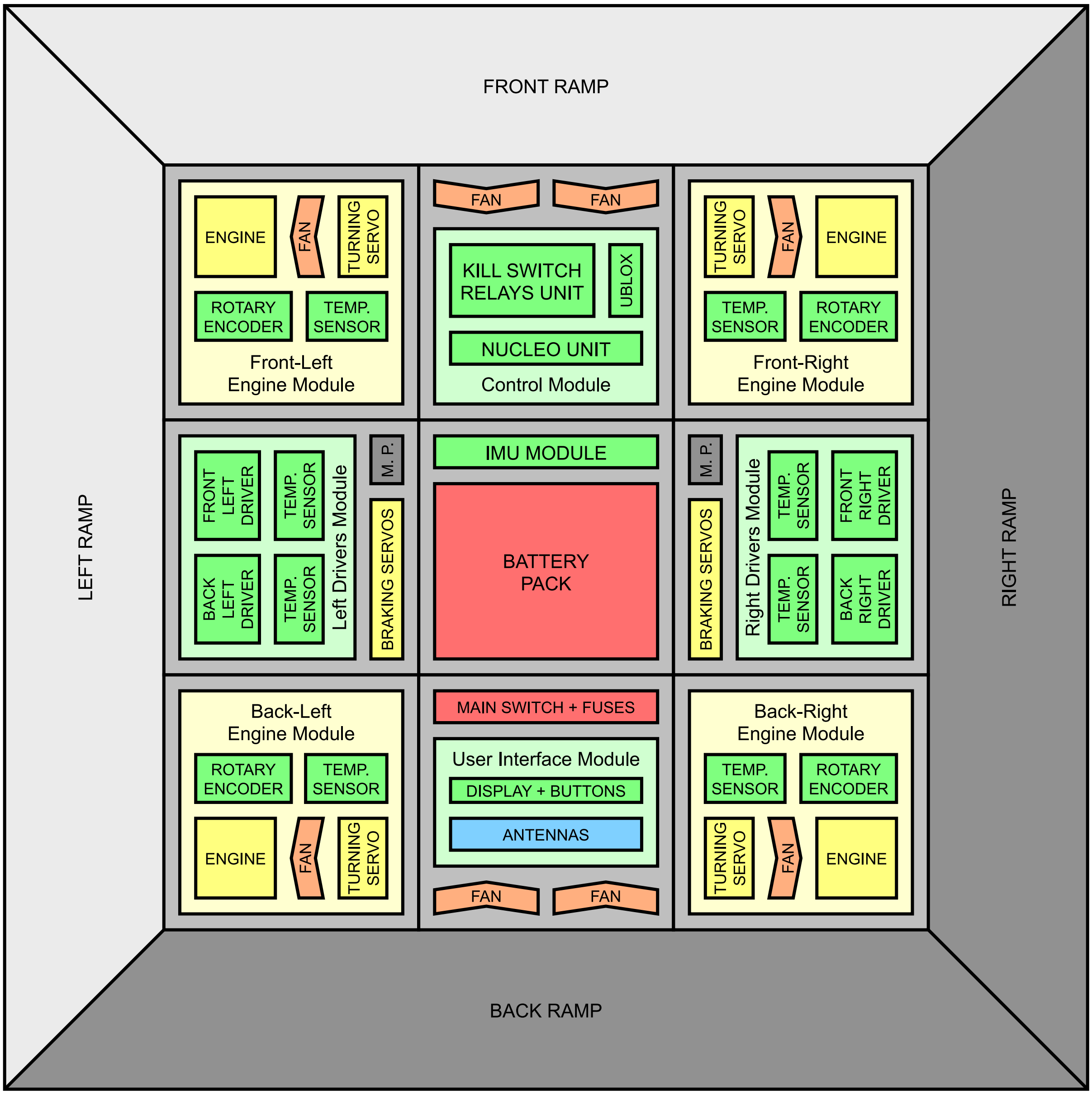


Fig. 3.8: Layout of the components in the ARP device

It can be seen in the figure that the platform has four side ramps to make it easily overrunable. It also deflects car's sensors' signals to make it harder to detect because the ARP's detection by car is unwanted (car needs to react only to the dummy not to the platform). Then there is an inside area where the modules can be found. In the middle, there is a lithium polymer (Li-Po) battery pack with the voltage of 24 V. It has a capacity of 80 Ah which provides enough power for a whole day testing. Plus it can be easily replaced for a spare one if necessary.

Then the design consists of four identical Engines Modules in the corners. In each of them, there is engine with its temperature sensor, wheel, rotary encoder, fan and possibly steering servo in case of steerable wheels modules. As said before, the modularity of the design allows testing of different possibilities. There are two choices for movement (thus for kind of wheels) possible so far. FIrst, the prototype has been tested with the straight wheels modules (see Fig. A.2a) where the slight turning is possible with the different RPM (Revolutions per Minute) on each side. These modules use separate BLDC engines. Then there is a possibility of the steerable wheels modules (see Fig. A.2b) which enables the omnidirectional movement feature. They are already designed and tested. In these modules, there are wheels with the BLDC engines embedded directly inside of them. The chapter 6 describes this idea of omnidirectional movement in more detail. The engines used for the different modules are described in the chapter 4.4.6.

The four remaining compartments contain the control electronics of the platform. On the left and right side there are the Drivers Modules. They each consist of two BLDC drivers (for the front and back engine on the particular side) with their own temperature sensors. Temperature measurement is one of the safety mechanisms used in the platform. Next, there is space for additional servos in these modules to ensure mechanical braking of the wheels. It is not implemented yet mechanically but the control of these servos is ready from the embedded systems point of view. On the top of the platform (above these compartments) there are mounting points (M. P. in the figure 3.8) to enable carrying the dummy targets that are described earlier in this chapter.

In the front compartment, there is the main Control Unit of the whole device which consists of the controlling MCU (microcontroller unit) board, a unit which enables emergency disconnection of the engines by Kill Switch Button and other modules used for the localization (e.g. GPS device – discussed in the next chapter – 4). Also, the IMU (Inertial Measurement Unit) module, which contains accelerometer, gyroscope and magnetometer sensors, is connected to the main Control Unit. It can be seen in the battery pack compartment but in fact it is placed above the whole platform because the magnetometer would not work inside the metal case. There are difficulties with its placement that will be discussed later. In the back compartment, there is User Interface Module which provides display and buttons to control the platform manually without PC Application. There is also the main switch with fuses and the antennas for communication as well. Additionally, there are fans for cooling the modules down. The air is then driven through the whole platform and is blown out along the engines with their own fans. This ensures cooling of all parts of the platform plus hot air from the engines does not affect the rest of electronics.

The whole system of the ARP for ADAS testing does not consist just of the platform itself but there are other devices connected through a wireless network as well. Also, there is a PC Application to provide the user interface. More information about them as well as the detailed description of the modules inside the ARP can be found in the chapters 4 and 5.

3.3 Goals and Methods of Doctoral Thesis

After the description of the ADAS testing field in automotive industry and the basics of the ARP, the goals of this doctoral thesis can now be summarized. As said before, the research is taking inspiration in the existing platforms (see chapter 2) and then it focuses on exploration of new ideas to improve this field.

3.3.1 Goals of Research

The goals are listed here:

1. Background research of the field of automated ADAS testing. Theory of ADAS tests, scenarios and dummies in general. Exploration of the possibilities to perform the tests using the Autonomous Robotic Platform (ARP). Taking inspiration in the existing devices:

   - Described so far in the introduction and in the chapters 2 and 3.

2. Design of the functional ARP prototype concept that can be used for ADAS testing as well as to try new ideas on how to improve the field:

   - The idea of the prototype's concept can be found in this chapter; its more detailed description from the embedded point of view (hardware and software) is then in the chapters 4 and 5. The prototype was developed in the company Valeo Autoklimatizace k.s. in cooperation with the other team members. Mechanical construction took inspiration in [26]. This doctoral thesis is focused on embedded systems part of the prototype and on the implementation of different innovative features that are summarized in the next items of this list. These goals are also a contribution of the author of this doctoral thesis to the field and were researched solely by this author.

3. Implementation of innovative features directly into the prototype's concept. They are the following:

   (a) Modularity.

   (b) Cloud based communication.

   (c) Automatic safety mechanisms.

   (d) Omnidirectional movement.

- These ideas are shown in the chapter 6.

4. Exploration of different possibilities for localization of the platform using conventional and also non-traditional sensors. Consecutive software filtering to obtain the position with sufficient precision (Kalman filter is used). The sensors that were tested during the research are for example:

   (a) Odometry.

   (b) IMU (Inertial Measurement Unit).

   (c) GPS (Global Positioning System).

   (d) Mouse Camera positioning.

   (e) Ultrasonic P2P (Point-to-Point) distance measurement.

   - These ideas are shown in the chapter 7.

5. Using the acquired position of the platform to perform its autonomous control (i.e. navigation) according to the trajectory defined by the scenario. Description of the different possibilities of ARPs and VUT synchronization:

   - This is the focus of the chapter 8.

6. Discussion of the obtained results, outline of the future development possibilities that can follow up on this research:

   - These can be found in the conclusion.

3.3.2 Scientific Research Methods

The doctoral thesis uses several methods to acquire the results arising from the specified goals:

1. At the beginning, the analysis of the field and the background literature research are performed. In the introduction and in the chapters 2 and 3, the papers that were used are cited.

2. Then the prototype's concept is proposed by theoretical approach using synthesis and deduction methods. Embedded systems blocks are described by the block diagrams to get the overall idea of the whole hardware concept. The software proposition for each block is realized by the flowcharts.

3. Each block's hardware is built using methods to design electrical schematics and printed circuit boards. The software for each block is developed using methods for microprocessor programming (see [29]).

4. Empirical methods are then used to test each of the blocks, where inputs are simulated (or given by a different block that was tested before) and outputs observed, measured and evaluated.

5. The prototype is then used to test other innovative features. Each feature is analysed, described theoretically and implemented in the prototype. Then the experiments are performed to show the results and to compare them with other similar methods. The results are used to conclude the efficiency of different features and to give ideas for their possible future improvements.

4

Hardware Design

The general concept of the Autonomous Robotic Platform for ADAS testing was described in the previous chapter (3) and now this chapter talks about the hardware design of the prototype. In the next chapter (5), the software propositions will be explained. Together these chapters give a detailed description of the whole system from the embedded systems point of view since this is the core of this thesis as mentioned before. The mechanical construction of the platform took inspiration in the master thesis [26] and the more in-depth description of this topic can be found there. This thesis will only cite its information when it is required. Topic of this chapter was also used in the papers [Pub. 6] and [Pub. 8] published by the author of this doctoral thesis.

The whole system does not contain just the ARP device but it consists of several devices connected over the wireless network. This concept can be seen in the figure 4.1. There is PC Application (which runs on the PC), Base Station, Kill Switch Button and ARP itself. Each of these units has its own XBee module that provides a wireless connection to the other units (ARP has two of them – one for Control Unit and one for the kill switch system – more about that will be explained later). Additionally to the wireless network, there are USB ports for communication between the modules. They can be used to update firmware in the units or to debug them from the PC.

XBee modules provide complete solution for a wireless communication using different protocols. Detailed information about XBee modules can be found in [11]. Our ARP system uses devices which communicate through the network according to the IEEE 802.15.4 standard good for low-rate wireless personal area networks (more about that in [21]).

Tab. 4.1

Parameters	Values
Data rate	250 kbit/s

Table continues on the next page . . .

Tab. 4.1 – continuation

Parameters	Values
RF line-of-sight range	3200 m
Transmit power	63 mW
Frequency band	ISM 2.4 GHz
Protocol	Proprietary IEEE 802.15.4
Encryption	128-bit AES

Tab. 4.1: Parameters of the IEEE 802.15.4 network using XBee modules

Further, the network uses the DigiMesh protocol designed by the XBee manufacturer ([11]) that makes the communication robust and safe. Main advantage is that it is not necessary for all the nodes to be visible to each other but they can retransmit the messages between themselves to find the final destination. The parameters of such a wireless network are shown in the table 4.1. More details about the network implementation to the system can be found in the chapter 6.2.

In the following sections, there is a description of the blocks from the diagram in the figure 4.1 on the next page. The ARP has its own block diagram that can be seen later in the figure 4.2 due to its higher complexity. The key features to design a reliable embedded system from the hardware point of view are to maintain different units small and focused on their goal and also to make the whole system as modular as possible. If one unit malfunctions it must be easily detectable and it must not influence or damage the rest of the system.

The mechanisms to ensure reliability and safety are implemented in the hardware as well as in the firmware of the modules (chapter 5). Last but not least, the whole hardware is designed to be dust and water resistant (IP55) which suits the test track environment with all the conditions such as rainy weather. This applies to all the boxes, connectors and cables. The used connectors must also withstand vibration arising from the movement on the bumpy surface. This can be subject to the further testing in the future.

4.1 PC Application

PC Application runs on PC or laptop and provides a user interface to the whole system. It can set its parameters, visualize its data or control its behaviour. Detailed functionality is explained in the chapter 5. It communicates with the rest of the system through PC Dongle which is connected to the USB port and enables interface to the XBee device. There is a functionality for the automatic detection of the connection or disconnection of the XBee module. In the PC, it behaves as a virtual serial port and serial port number is also found automatically by using the manufacturer's ID on the device. The XBee device is connected to the wireless network as said before. This PC can also be used to debug other devices through the USB ports or to update their firmware.

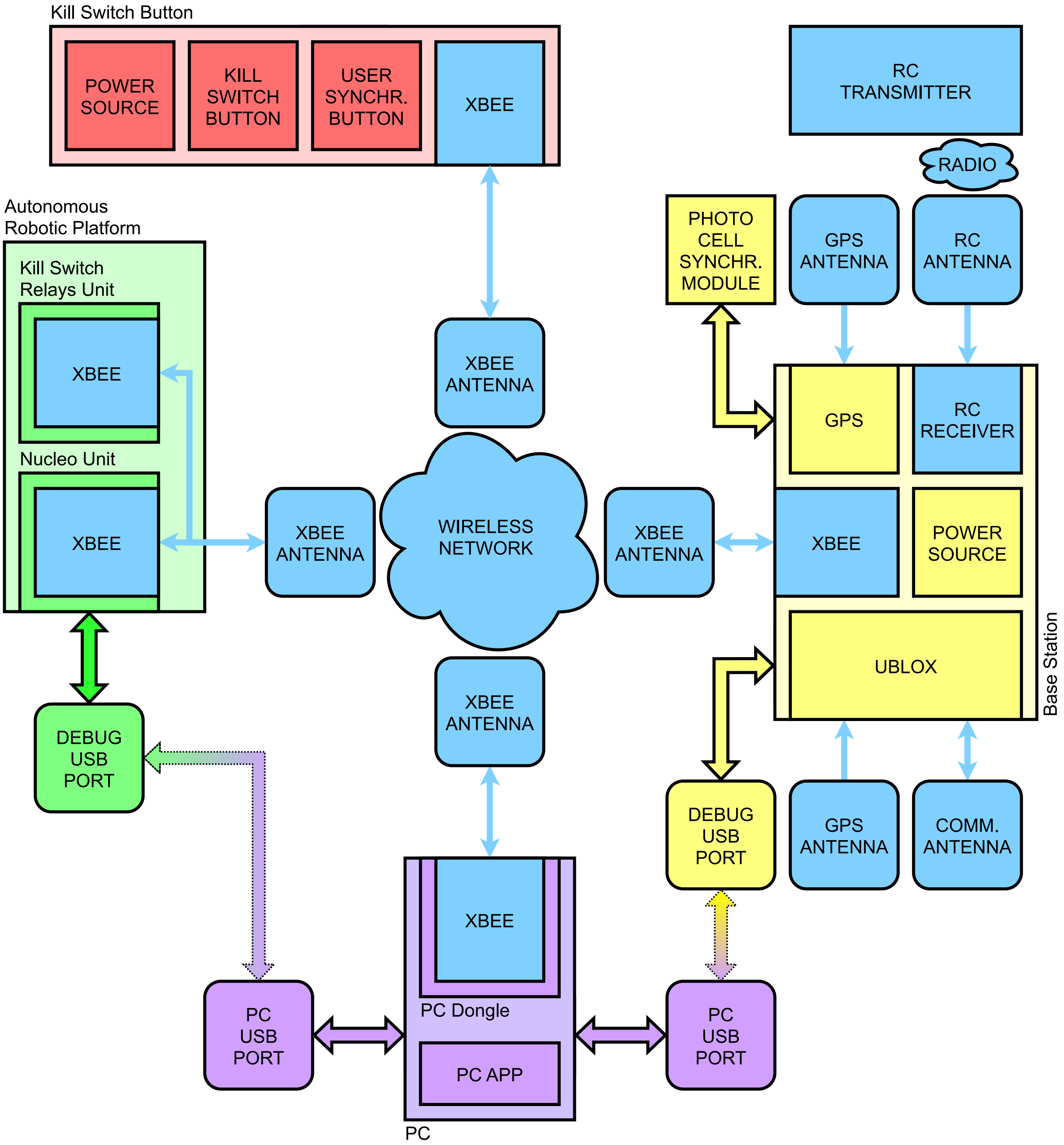


Fig. 4.1: Block diagram of the whole system

4.2 Base Station

Base Station is a device that has a role of the coordinator of the whole system. It has its own power source sufficient for at least a whole day of testing. It is connected to the wireless network through its XBee unit and it stores variables that need to be exchanged over the wireless network for all the other modules. In every cycle, all connected devices upload their data to Base Station which transmits them to the network where the other devices can read them.

It has its own GPS with an antenna so it knows its position. It makes a reference coordinates point of the testing system too. More about the localization possibilities can be found in the chapter 7. Instead of an ordinary GPS, it can use different solutions in the form of devices that work on a DGPS principle (such as u-blox device [36]) which is represented in the block diagram as the UBLOX block with its GPS and communication antennas. These are planned to be tested for the localization as well.

It has connection possibilities to the other components that provide the user control. There is the RC receiver that can be paired with the RC transmitter. This transmitter can be used to control the ARP manually. Then it contains port for the connection of a photocell synchronization module which can synchronize the ARP with the VUT. Once the vehicle goes through the coordinates represented by photocell the ARP can start performing its scenario.

4.3 Kill Switch Button

Kill Switch Button device has its own power source for a whole day testing and its XBee device for a network connection too. It contains an emergency button called a kill switch which provides possibility to stop the ARP in case of malfunction. It has been said before that the heavy platform can be dangerous when it reaches high speeds so there needs to be an option like this. It disconnects the drivers of the engines which causes the platform to brake to a halt electrically, then it can activate the braking servos for mechanical braking and finally it disconnects the power from the whole ARP through its main relay switch. Since it is wireless, it works on the principle – while there is a signal, the system works. Therefore, in case when even Kill Switch Button malfunctions, the ARP is stopped immediately. The functionality is shown in more detail in the next section about ARP and also in the chapter 5 that deals with software.

Additionally, this device has a user synchronization button which can be used to trigger scenarios manually on the same principle as the photocell synchronization from Base Station works.

4.4 Autonomous Robotic Platform (ARP)

This section talks about the ARP itself. The concept of the platform has been introduced in the chapter 3 and now its modules are described more closely. The platform consists of the battery pack with the main switch, Control Module, IMU Module, User Interface Module, Drivers Modules and Engines Modules. Its block diagram can be seen in the figure 4.2; its physical layout in the figure 3.8 back in the chapter 3. In the next sections this block diagram is explained.

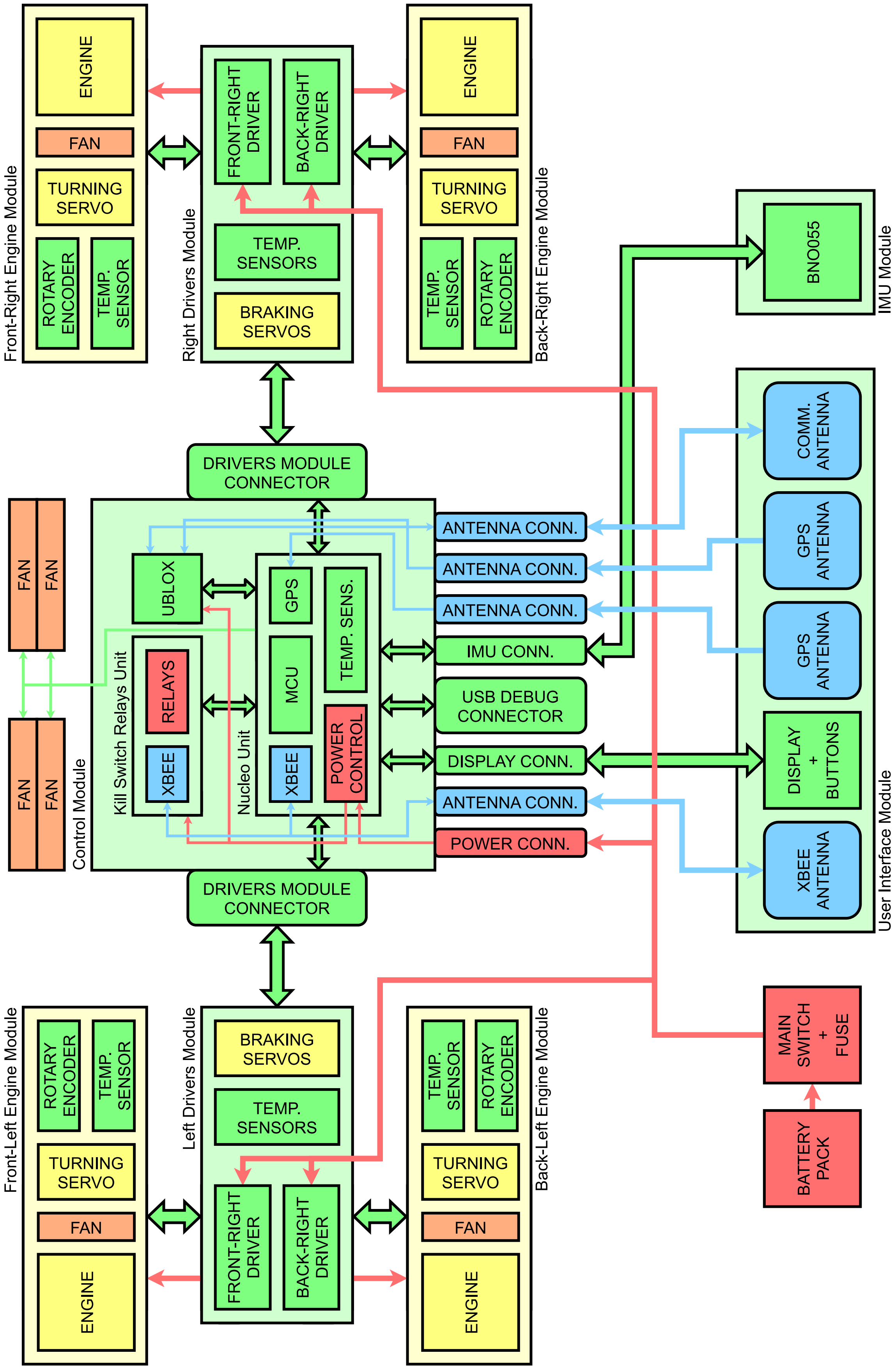


Fig. 4.2: Block diagram of the ARP device

Colour scheme in this figure corresponds to the figure 3.8. Green colour is used for the control electronics, yellow represents the engines related electronics, blue stands for antennas and communication modules, red shows the power electronic parts and orange is used for cooling fans.

4.4.1 Battery Pack and Main Switch

The platform uses the Li-Po battery pack with six cells connected in series and four in parallel. Nominal voltage per cell is 3.7 V so the overall nominal voltage gets to 22.2 V. Its capacity is about 80 Ah (1.776 kWh with the nominal voltage). The battery is easily replaceable and two batteries are then enough for a whole day of testing which is considered to be about 8 h. Detailed power consumption values are shown in the table 4.2.

Tab. 4.2

Mode	Current (at 22.2 V)	Consumption	Time of operation
Standby mode	430 mA	9.55 W	186.0 h
Fans only	720 mA	15.98 W	111.1 h
Speed of 5 km/h	18 A	399.60 W	4.4 h
Speed of 25 km/h	88 A	1953.60 W	0.9 h
Acceleration of 2 m/s^2	124 A	2752.80 W	0.6 h
Total (*)	17 A	384.12 W	4.6 h

Tab. 4.2: Power consumption of the ARP

The main switch is an automotive relay capable of switching up to 200 A. With the maximum current consumption limited by the drivers to 40 A per Engine Module, it is sufficient for the whole platform. It can be turned on and off manually by the main switch buttons on User Interface Module. That makes it one of the possibilities for turning the platform off in case of malfunction too. It can be also turned off by Kill Switch Relays Unit (see section about this unit below). There are fuses for different parts of the platform placed near the main switch as well.

4.4.2 Control Module

In this module, there is Control Unit (shown as Nucleo Unit) and Kill Switch Relays Unit. First, Control Unit is discussed. It is in control of the whole platform's operation. As its

*) Total consumption is computed from the scenario where 75 % of time is taken by the standby mode, 10 % by the platform going at speed of 5 km/h (pedestrian movement), 10 % by the platform going at speed of 25 km/h (cyclist movement) and 5 % by the platform's acceleration. Fans are turned on for 25 % of total time.

core, there is a 32-bit Arm Cortex-M7 MCU embedded inside (specifically STM32F767ZI – see its datasheet [32]). It is used together with the hardware module called Nucleo by the company ST Microelectronics that ensures its basic operation. This MCU communicates with the other wireless devices (PC Application, Base Station and Kill Switch Button) with its XBee device and its antenna. It also gets data from the platform's sensors (IMU Module, GPS with its antenna or Hall effect sensors embedded in the engines).

It can then control the platform's movement through the communication with Drivers Modules. It has its own temperature sensor and power control block for battery voltage measurement (temperature measurement throughout the platform and battery voltage measurement are included as the safety mechanisms of the platform and the ARP is turned off when they reach critical values). Control MCU can also control the cooling fans in case the temperature rises. There is also a possibility to connect different solutions for localization which is represented as the UBLOX module with its GPS and communication antennas. They are planned to be tested in the future and some of them are already implemented in the measurements shown in the chapter 7. Finally, direct user input and output is possible through User Interface Module.

The stand-alone Kill Switch Relays Unit which uses its own XBee and works as a hardware safety mechanism is also a part of Control Module. This XBee communicates with Kill Switch Button device. Once the communication is lost (e.g. Kill Switch Button is pressed, Kill Switch Button device malfunctions or the platform gets too far away out of this signal range) the unit stops the platform's movement. More details are shown in the chapter 5.

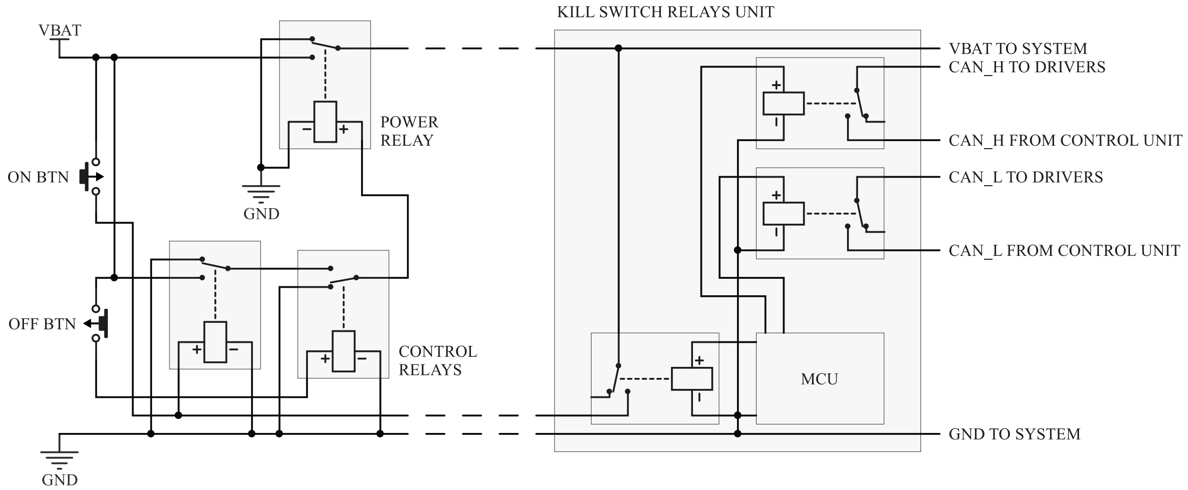


Fig. 4.3: Connection of the power relays

From the hardware point of view, the connection of the relays is an interesting feature (figure 4.3). The main power relay is controlled by manual "on/off" buttons (they work as a manual safety mechanism as well) or by control relays. When the "on" button (BTN) is pressed, the left control relay routes power to the power relay which turns the ARP on. The left control relay is then held on by the MCU in Kill Switch Relays

Unit which keeps the ARP running after the ”on” button is released. After that, the ”off” button can turn the ARP off manually through the right control relay (it can also be used as a direct kill button in case of emergency). It works on the reverse principle (pushed means disconnected). The ”off” button also commands the MCU to release its relays as a backup in case the right control relay fails to turn off. MCU can also turn the power off through its relay without the ”off” button (used for Kill Switch Button module functionality as said before).

In the schematic, there are also additional relays that can interrupt the CAN bus (CAN_H and CAN_L) communication from Control Unit to Drivers Modules which causes the ARP to stop its movement (see the chapter 5 for more information).

All the relays are connected in the way that if anything fails it leads to disconnection of the main power. Further, the correct operation of the relays is automatically diagnosed by the MCU and also indicated to the status LEDs for manual diagnostics (which should be done regularly). That makes the main power connection as safe as possible.

4.4.3 IMU Module

The IMU (Inertial Measurement Unit) Module is connected to Control Module and gives the controlling MCU information about acceleration, rotation and heading of the platform. It consists of three sensors (accelerometer, gyroscope and magnetometer). It is connected by the CAN bus which is chosen due to its high noise tolerance. From the hardware perspective, the interesting feature is its connection. It is attached through a magnetic connector and located on the rod about 50 cm above the platform in an impact resilient box. In case a car drives over the platform it is tossed away and it must survive the impact. Then it can be mounted back again. The reason is that the magnetometer would not work inside the steel chassis of the platform and the magnetometer sensor is crucial for correct localization. It has a backup battery so it is not turned off when disconnected which would cause a loss of its calibration.

4.4.4 User Interface Module

User Interface Module consists of user buttons and LCD display. It enables direct user access and also switching the platform on and off. The display contains brief information about the ARP’s status (battery voltage, temperature, errors, etc.). There are also antennas leading from Control Module mounted here (it is beneficial to place all the components that need to be accessible from above to one place from the platform’s durability point of view). The antennas for XBee devices are from flexible wires and the GPS antenna is sunk under the platform’s surface so they are not destroyed by the car’s impact. Only other parts above the platform are the mounting points for the Euro NCAP targets and the IMU Module (described before) – these do not interfere with the platform’s durability.

4.4.5 Drivers Modules

Drivers Modules consists of the BLDC drivers that control the engines, temperature sensors and braking servos for mechanical braking. The platform uses VESC drivers (see [6]). Drivers communicate over the CAN bus with Control Unit; they receive the requested RPM and send back the status about the actual RPM, currents and possible errors. The CAN bus is chosen due to its high noise resistance. Drivers are powered directly from the battery pack and they also enable the power recuperation through the electrical braking of the engines. There are four drivers in the whole platform. The embedded temperature sensors (that measure the temperature of the drivers) and the braking servos are connected directly to Control Unit as well. In case of emergency, the braking servos can be also activated by Kill Switch Relays Unit. Their state is monitored by the Hall effect sensors and sent back to Control Unit.

4.4.6 Engines Modules

The last part of the platform are Engines Modules. There are the BLDC engines and wheels for the movement of the platform. There is a possibility of the straight wheels modules (Fig. A.2a) or the steerable wheels modules (Fig. A.2b). The first one uses the standard BLDC engines (see [24]) connected to the wheels by the belt pulleys. The second one uses the in-HUB BLDC engines (see [14]) embedded directly inside the wheels. In these steerable modules, there are also additional steering servos controlled by Control Unit. Parameters of the engines are summarized in the table 4.3.

Tab. 4.3

Parameters	Standard BLDC engine	HUB BLDC engine
Number of poles	6	28
Number of phases	3	3
Rated power	220 W	1500 W
Rated DC voltage	24 V	50.4 V
Rated constant current	16.6 A	30 A
Rated peak current	40 A	60 A
Rated RPM	3500	2875
Rated torque	0.6 Nm	5.3 Nm

Tab. 4.3: Parameters of the possible BLDC engines

The straight wheels modules have an advantage of being more simple and durable and they are able to slightly steer the platform to maintain its direct straight movement based on its localization. If the ARP is planned to be used only for scenarios with straight trajectories, it is advantageous to use these modules. On the other hand, when the ARP is used for complex scenarios, it needs to be equipped with the steerable modules. Plus, the modularity of the ARP's hardware enables easy switching of these modules, therefore the ARP can use proper modules for different scenarios. It can be quickly adjusted before every testing.

The steerable modules give the platform possibilities of different types of movement which is one of the innovative features tested by this research. More about that can be found in the chapter 6.

In Engines Modules, there are also the temperature sensors (mounted on the engines and sending their data to Control Unit) and fans controlled by Control Unit. In case the temperature rises, the fans are turned on. Next, the rotary encoders are good for steerable wheels modules to measure their position in the sense of the platform's direction. The RPM of the engines in both modules is measured by the Hall effect sensors embedded in the engines. These Hall effect sensors are connected to the BLDC drivers.

5

Software Proposition

With the previous chapter (4) describing the hardware of the prototype, this chapter can now focus on the software proposition. In its sections, there is the firmware of the MCUs of the different system parts explained. Several flowcharts are used because they are a good tool for software description. Also, PC Application that works as user interface of the system is shown here. In the flowcharts, there are the main ideas and procedures that make the whole prototype work and ready for ADAS testing. This chapter led to the papers that can be found under items [Pub. 7] and [Pub. 8] published by the author of this doctoral thesis.

The firmware of the MCUs is programmed in the C language, while the application in the C# language. They are all designed to be as modular as possible to maintain their robustness and order in the large embedded system. The main code is keeping the basic structure clear and it uses other source codes with functions for different peripherals and modules. The important factor of the code is to operate in the real-time manner. There is an embedded hardware timer in each device dedicated for triggering the processing loop with the desired period. These timers are synchronized by the GPS PPS (Pulse per Second) signal so they work with the same timing among all the devices in the system.

Embedded watchdog timers (WDT) in the MCUs then control that every processing iteration of each task is finished before the next one is requested. Other hardware timers are set for the peripherals, communications etc. They work as watchdog timers as well. Every peripheral in the system is controlled by such timer. If it starts malfunctioning, its timer detects that and the proper error state is set. Any error in the system means the movement of the platform is stopped. In fact the system is running its constant diagnostics of proper operation. Together, real time operation, software watchdog timers and other safety mechanisms implemented in the hardware (see chapter 4) provide the maximum level of safety in the whole ARP system.

Other aspect of the software in the large embedded system is its reliability. Wireless communication is equipped with a handshake mechanism plus it is controlled by watchdog timers. Wired communications are controlled by checksums (computed over the messages) and watchdog timers as well. Every task in the processing loop has its priority (e.g. the engines communication is more important than sending data for visualization). Tasks that are not needed to be processed in one iteration can be divided and every task has its watchdog timer to control that it is finished when necessary. Everything must be able to work together and minimize the probability of getting stuck in an unwanted state. Each module is capable of its own diagnostics and resetting itself in case of malfunction. If the error that caused the reset is sporadic, the reset will solve the problem and the system will continue its operation. On the other hand, if the error persists, it will be indicated as the error state and the system evaluates its severity. The less severe errors (such as loss of wireless signal) lead to the stop of ARP movement and indication of the error to supervisor who can resolve it. Then the operation continues as well. Critical errors (such as exceeded temperature of engines) lead to the shutdown of ARP and must be assessed more deeply before the ARP can be used again. More about the overall functionality of safety mechanisms can be found in the chapter 6.2.

The idea of the communication between the devices is that Base Station works as a data cloud and that the mesh based communication is used. The Base Station stores data from the devices and periodically sends them out to the network. This cloud version is more convenient for the possibility to build larger systems with more ARP devices. The devices can also communicate with each other directly, when the message is not used by more than two devices, to reduce the data in the network. In the next list, the idea of what happens during every processing iteration in the cloud is outlined (if more ARPs are included):

1. Every ARP sends data to the Base Station cloud.

2. All Kill Switch Button devices send data to the cloud.

3. PC Application sends its data to the cloud.

4. Base Station broadcasts data from its cloud and other devices read what they need.

More detailed information about the communication is described in the chapter 6. It includes diagram of the communication network, as well as the different steps of the process that leads to information exchange among the units.

5.1 PC Application

PC Application takes care of user interface. It is connected to the wireless network through the USB PC Dongle for communication with the other modules of the system.

In the flowchart in the figure 5.1, it can be seen that it uses events from the buttons (START, STOP, CLOSE and CONTROL BTN) to send commands to the application or to set parameters to the system. It reads incoming bytes over the wireless

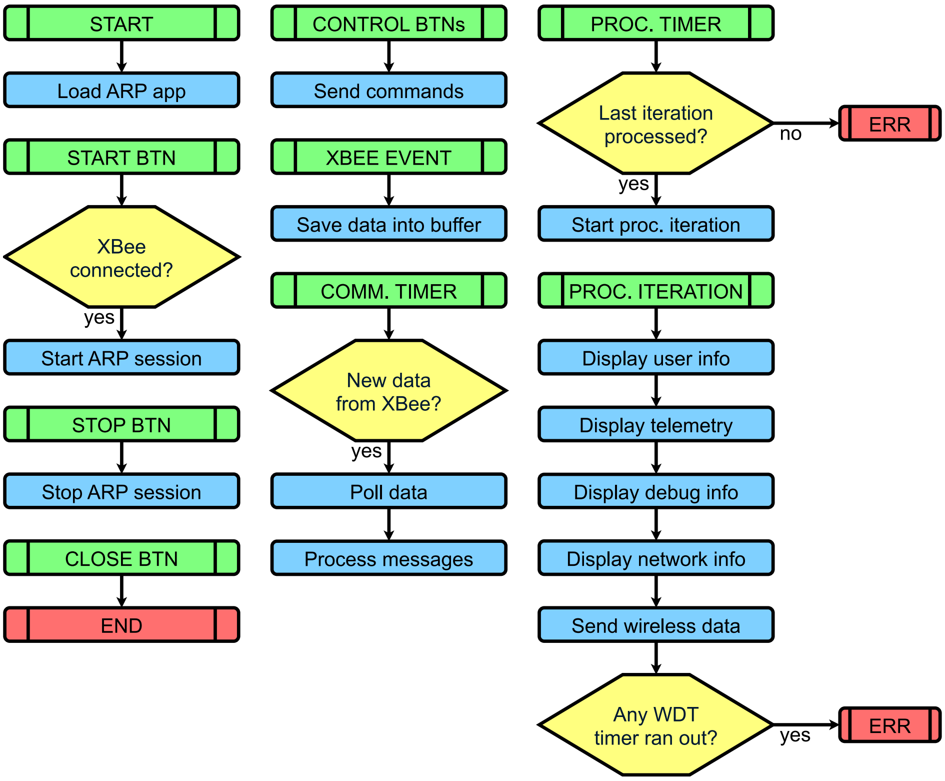


Fig. 5.1: Block diagram of PC Application

network with another dedicated event and then it processes the messages by polling them through a timer. The processing timer is triggering the periodical processing loop, which displays system status, telemetry, debugging information or wireless network status. Then the wireless data are sent to the other devices as well. As said before, the application controls its performance by the watchdog timers and errors are displayed if needed. Its visualization can be found in the attachment C.

5.2 Base Station

Base Station is a device that makes a coordinator of the wireless network and provides reference position for the whole system. Flowchart in the figure 5.2 explains its firmware.

First, the MCU performs initialization and then it reads wireless and sensor data in the infinite loop. GPS data are received by processing the messages with NMEA-0183 format captured on the UART bus. It is a protocol used by most of the GPS solutions worldwide. See [25] for the complete standard's description. The most suitable message to obtain all the necessary information is called GGA (Global Positioning System Fixed Data). It contains time, position, and fix related data and its format is included in the following example plus it is explained in the figure 5.3.

```
$GPGGA,161229.487,3723.2475,N,12158.3416,W,1,07,1.0,9.0,M, , , ,0000*18
```

Also, the processing iteration is activated periodically if it is requested by the timer.

The timer is synchronized with the GPS PPS signal and it runs the watchdog timers too.

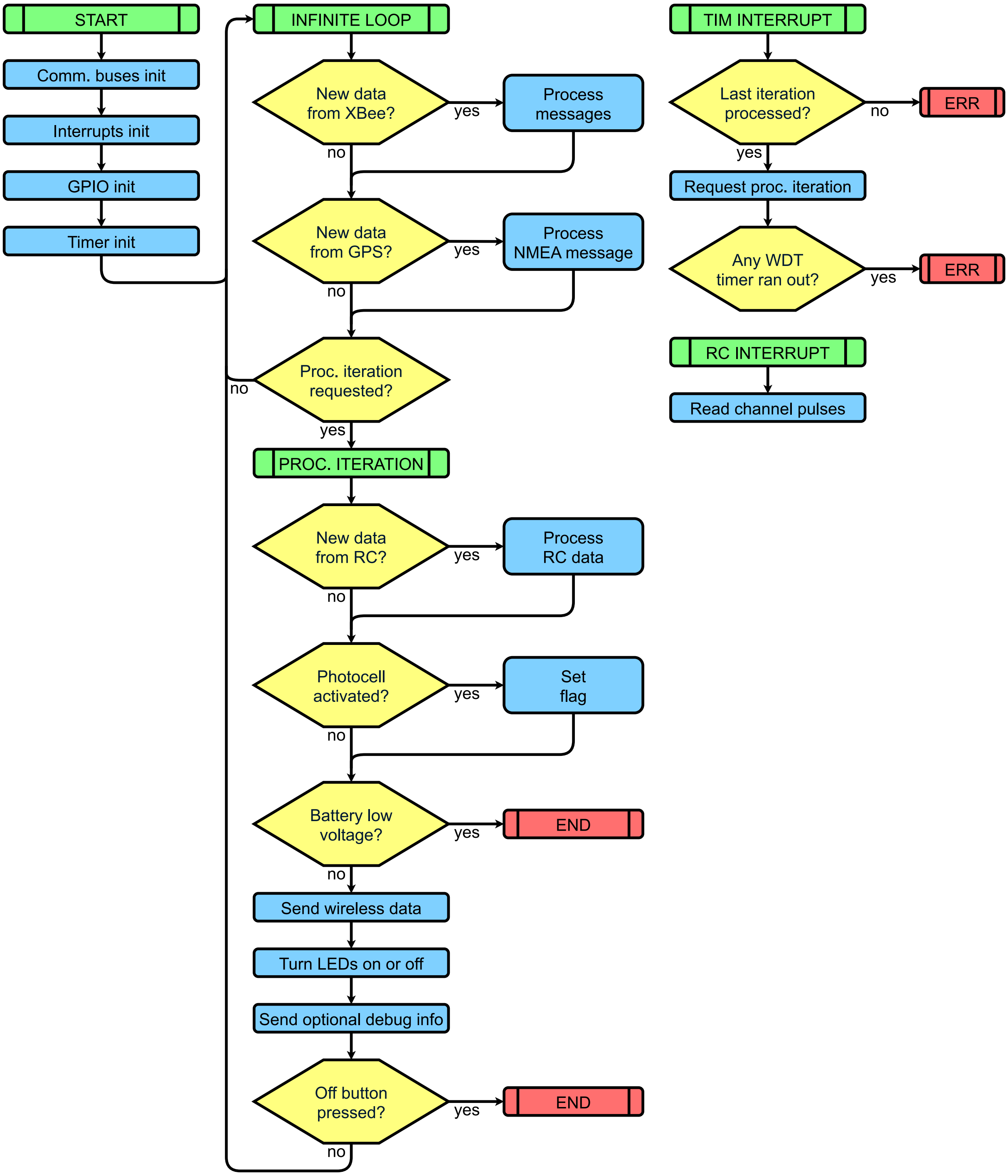
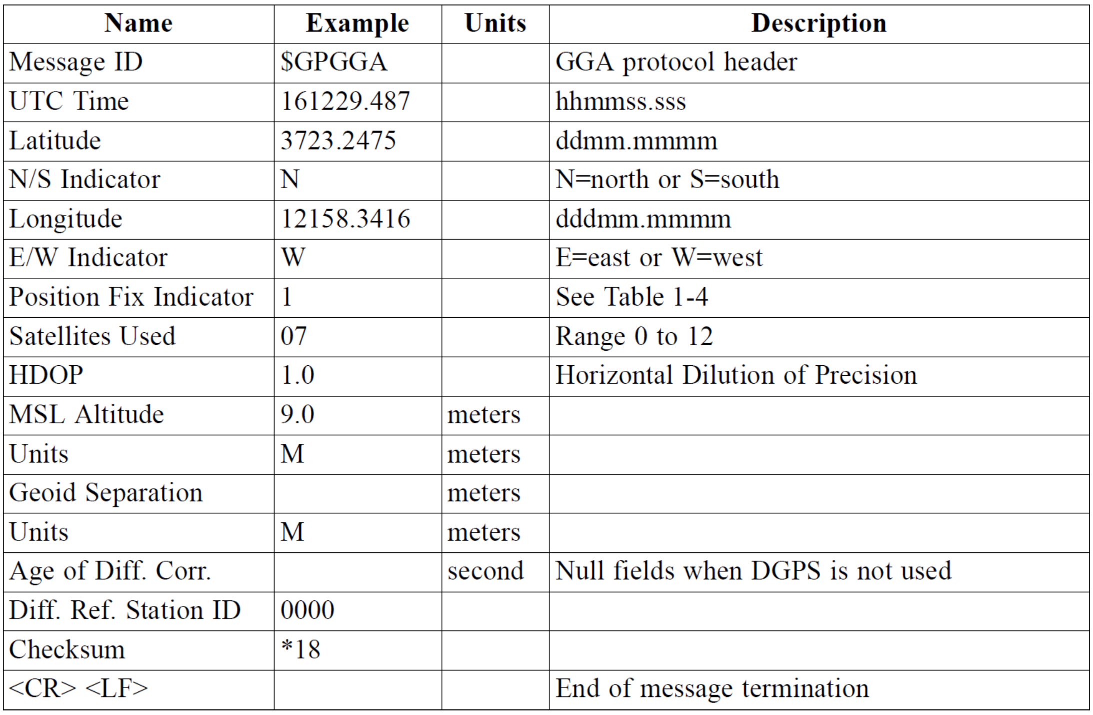


Fig. 5.2: Block diagram of Base Station

In the processing iteration, the data are processed and following actions are taken. If the battery voltage is low the device is turned off. Data from the RC controller and the photocell are sent to the ARP Control Unit, where they can control its movement. RC is good for manual movement; photocell can trigger predefined scenarios. Also the indication LEDs are turned on and off. If the "off" button is pressed, the unit is switched off. Photocell synchronization is giving boolean value ("1" – photocell is interrupted, "0"

– photocell is empty). There is an additional interrupt that reads signals from the RC. Each RC channel is connected through one PWM modulated signal (see figure 5.4), where duty cycle is set according to its value (position of the control switch related to the RC channel). Period of each PWM signal is 20 ms.



Name	Example	Units	Description
Message ID	$GPGGA		GGA protocol header
UTC Time	161229.487		hhmmss.sss
Latitude	3723.2475		ddmm.mmmm
N/S Indicator	N		N=north or S=south
Longitude	12158.3416		dddmm.mmmm
E/W Indicator	W		E=east or W=west
Position Fix Indicator	1		See Table 1-4
Satellites Used	07		Range 0 to 12
HDOP	1.0		Horizontal Dilution of Precision
MSL Altitude	9.0	meters	
Units	M	meters	
Geoid Separation		meters	
Units	M	meters	
Age of Diff. Corr.		second	Null fields when DGPS is not used
Diff. Ref. Station ID	0000		
Checksum	*18		
<CR> <LF>			End of message termination

Fig. 5.3: NMEA-0183 GGA message format |Borrowed from [30]|

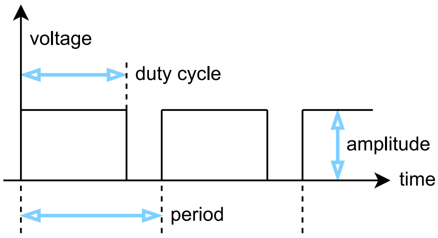


Fig. 5.4: PWM signal

5.3 Kill Switch Button

Kill Switch Button's firmware can be seen in the figure 5.5. It provides user safety mechanism for the whole system.

After the initialization of peripherals, it runs in the infinite loop. There is processing of XBee incoming messages from other modules and periodical call of the processing iteration. In the iteration triggered by the timer, there is the main function of the module – to read the kill switch and the trigger button. Also the battery voltage is measured and the unit is turned off if it is too low. Then the data are sent over the wireless network and indication LEDs are turned on and off. The trigger button information is sent to the ARP Control Unit where it can start the predefined scenarios in the same way as the photocell from Base Station. Finally, if the "off" button is pressed the unit is switched off.

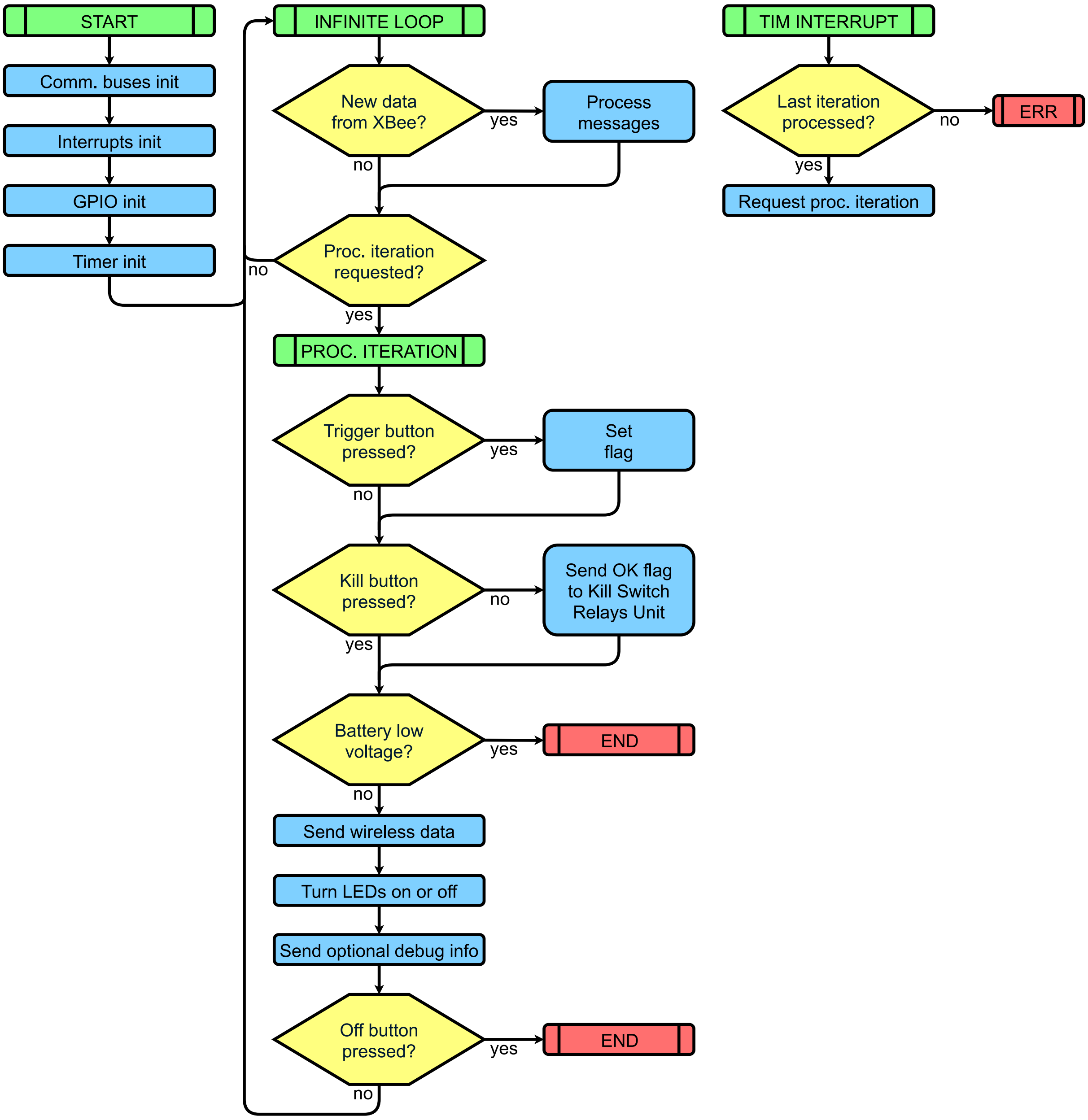


Fig. 5.5: Block diagram of Kill Switch Button

5.4 ARP Control Module

5.4.1 Control Unit

Control Unit is the core of the ARP device. It is explained in more detail in the figure 5.6 and in the rest of this section.

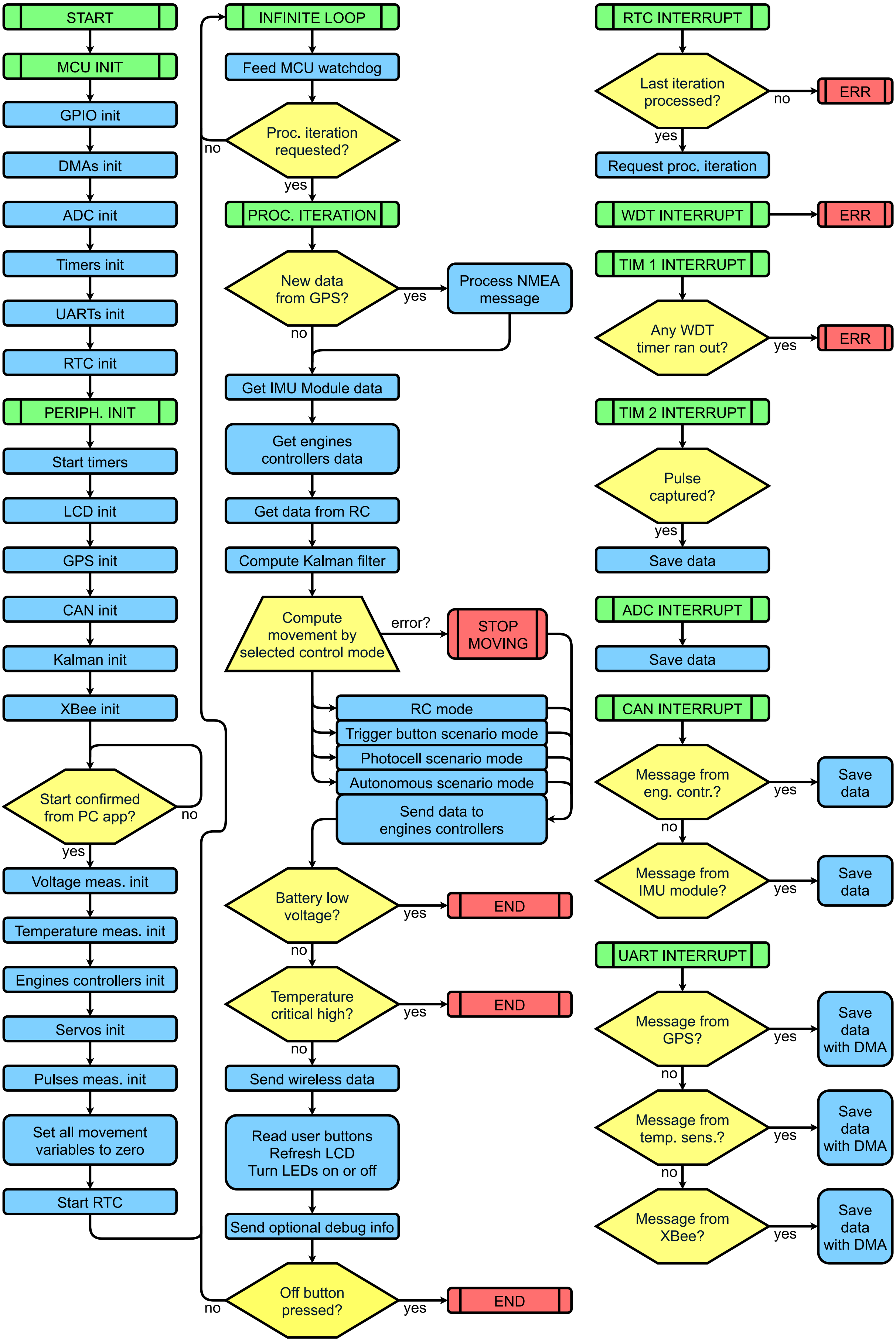


Fig. 5.6: Block diagram of Control Unit

The main aspects of Control Unit of the ARP are safety and reliability so the firmware implements mechanisms introduced in the introduction of this chapter (including the described modules self-diagnostics). Control Unit uses hardware watchdog timers for the MCU itself (dedicated timer with the detection of MCU freezing) and for all the peripherals as well (standard timers). Any timer activation means the movement being stopped. Tasks for different modules have their priority and can be interrupted by higher priority ones.

The flowchart represents the different tasks as they are performed in the MCU. First, it performs initialization of all the peripherals and the ARP waits for the confirmation of the start from PC Application. Then it continues its operation in the infinite loop where different tasks are performed periodically when it is requested by the RTC (Real-Time Clock) timer. This timer is synchronized to the GPS PPS signal to be in synchronization with the rest of the system.

First, the GPS data are read. It uses the same NMEA message as it was described in Base Station section and it is connected with the UART bus (Universal Asynchronous Receiver-Transmitter). IMU data (including acceleration, angular velocities and heading) is collected from the IMU Module and RPM is read from the engines controllers in Drivers Modules (all these units send their data over the CAN bus – Controller Area Network). RC data are given by the Base Station. When all the sensors are read, the current ARP position is evaluated using the Kalman filtering and the movement is computed (to perform ADAS testing scenarios which is the goal of the whole system):

1. Position is computed by the Kalman filter from the sensors data. More information about the Kalman filter can be found in the chapter 7.

2. Movement is computed by the control algorithms according to the selected mode. If there is any error, it is cancelled. ARP can move by the RC control, by the predefined scenarios triggered through the photocell at Base Station or through the trigger button at Kill Switch Button device or it can move autonomously on the predefined trajectory at the predefined time. Navigation possibilities are explained in more detail in the chapter 8.

3. Command is sent to the engine drivers in Drivers Modules to actually perform the movement. Engines controllers are connected over the CAN bus and then engines themselves are controlled by the sine voltage signal generated by the controllers.

Then Control Unit measures temperature in all the modules of the ARP and voltage on the battery. The temperature sensors are connected over the bus called 1-Wire; the voltage is measured using the ADC (Analog to Digital Converter). If the values are critical the system is shut down. Other means to stop the movement are for example the detection of blocked engines (ARP wants to move but cannot), the stop signal from PC Application, the stop signal from Kill Switch Button, the main switch on ARP or the detection of leaving the testing area (once the ARP leaves the defined GPS coordinates it stops). More information about the safety mechanisms is in the chapter 6.

Finally all the necessary wireless data are sent over the XBee unit to the rest of the system. This unit uses UART for communication. User input is read from the buttons, LCD information is refreshed and status LEDs are turned on/off.

In the additional interrupts, other data are processed outside the infinite loop as they arrive. Pulses from the RPM encoders (used for the steering servomechanisms' angles evaluation) are measured, ADC channels are read, CAN bus and UART messages are received.

All the peripherals that require time consuming communication from the MCU point of view (such as 1-Wire, UART or CAN bus) use the DMA (Direct Memory Access) to send or retrieve the data to minimize the consumption of the MCU's computation power.

5.4.2 IMU Module

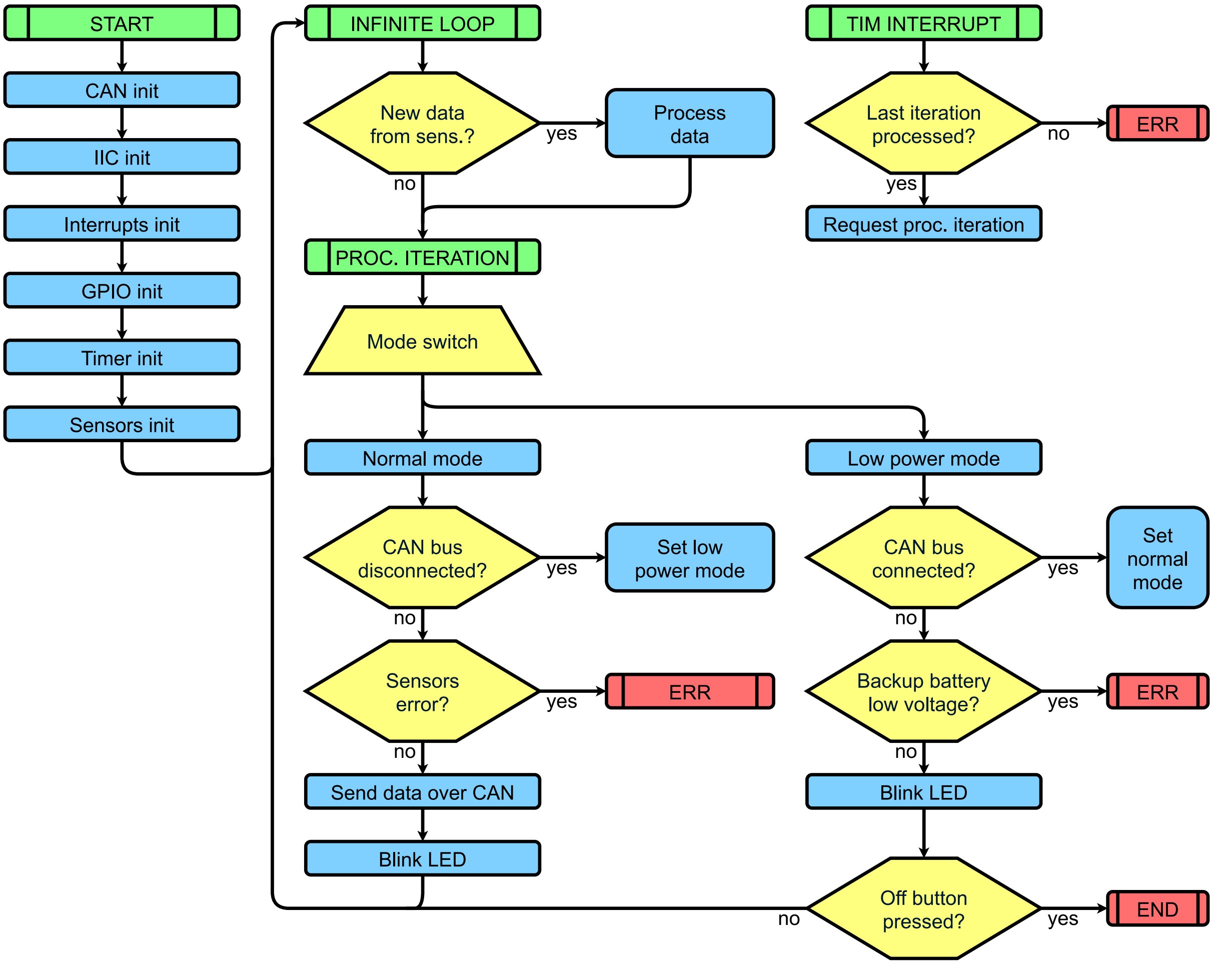


Fig. 5.7: Block diagram of the IMU Module

There is also the IMU module which is connected to Control Unit too. It has its own MCU so its firmware is explained in the figure 5.7. It must be placed outside of the metal construction as it was explained in the chapter 4.

First the initialization is done. Then in the infinite loop the data from the IMU sensors (accelerometer, gyroscope and magnetometer) are read. These data are given by the BNO055 ([8]) sensor connected over the IIC bus. The processing iteration is then triggered periodically by the timer interrupt. The module can be in two modes that decide about the next tasks that are performed in the iteration. If Control Unit is connected, the module is powered from Control Unit and it is in the normal operation mode. Also the backup battery is being recharged. The module checks for the sensors errors, sends data over the CAN bus to Control Unit and blinks the status LED with the longer flash. If the CAN bus is disconnected, the module enters the low power mode.

In the low power mode, the module is powered from the backup battery, it checks for its voltage and is turned off when it is too low. It uses low power MCU modes to prolong the battery life. If the "off" button is pressed it is turned off as well. Status LED is blinked with the shorter flash. If the CAN bus is connected, the module enters back into the normal operation mode. The reason for this low power mode with the backup battery is the fact that the IMU sensors need to be manually calibrated after the power up, which takes some time. So the low power mode can run with the backup battery up to one week to maintain the calibration for the next power up of the platform. Also, when the module is disconnected due to the collision with the VUT (see chapter 4.4.3), the calibration values persist.

5.4.3 Kill Switch Relays Unit

Kill Switch Relays Unit is part of Control Module as well. Its main function is to ensure safety of the system if there is a failure in Control Unit or in the engines drivers which could lead to uncontrolled ARP movement. Its firmware is shown in the figure 5.8.

After the initialization of the peripherals, there is again the infinite loop. It processes data from the other devices coming over the wireless network through the XBee module (connected over the UART). The processing iteration is triggered periodically by the timer interrupt. The main and only function of the unit is to check the status of Kill Switch Button. If there is OK flag from the button, acknowledge is sent and the relays that enable the platform's movement are kept turned on. If the OK flag is not received (kill button pressed or communication lost) Kill Switch Relays Unit activates its emergency sequence to stop the ARP's movement:

1. It sends a signal to Control Unit to brake to a halt electrically and if the movement continues it can activate mechanical braking as well (with the use of braking servomechanisms).

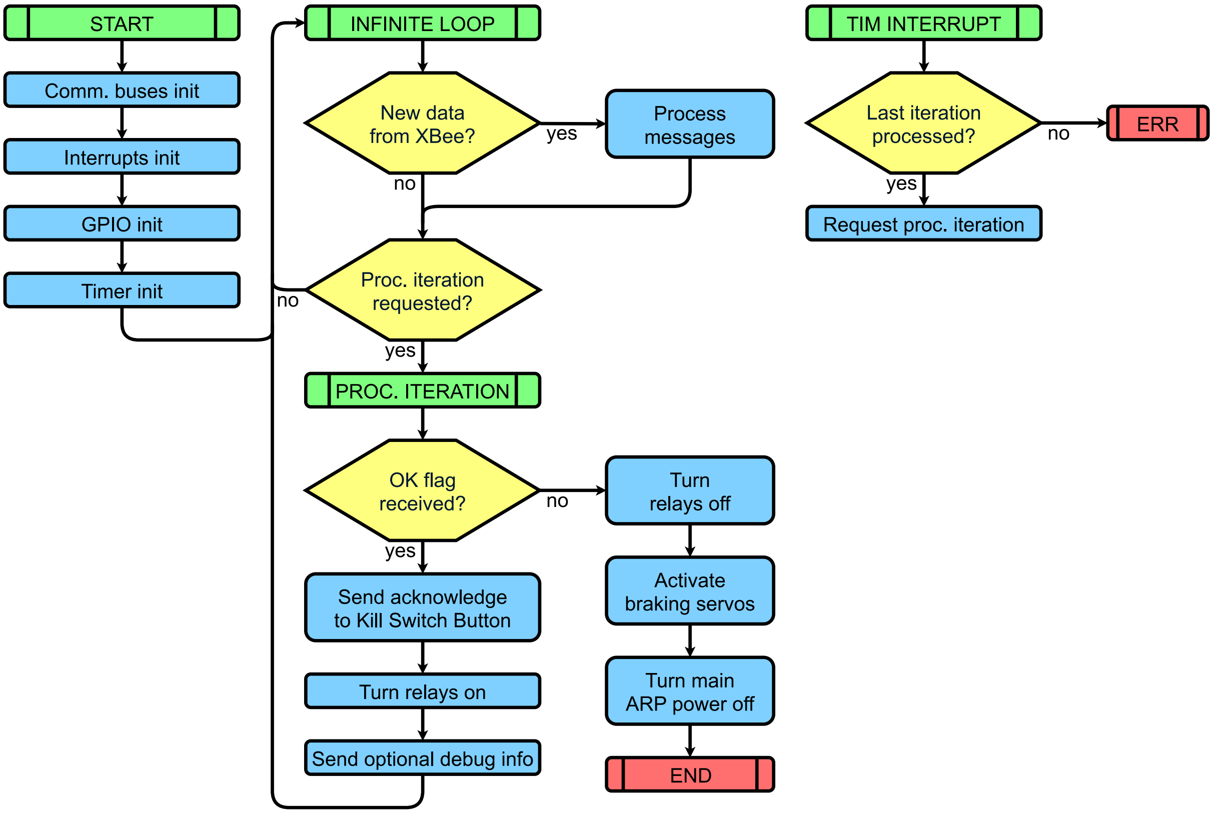


Fig. 5.8: Block diagram of Kill Switch Relays Unit

2. After the ARP is stopped or if braking is not working, it turns the relays shown in the figure 4.3 off. Relays interrupt the communication from Control Unit to Drivers Modules. That makes the drivers brake the engines electrically (they have their emergency stop after the loss of communication) in case Control Unit did not respond to the previous braking command.

3. If that fails too, last possibility to brake is to activate the braking servos for mechanical braking directly with other relays (independent on Control Unit and Drivers Modules).

4. Finally, it can turn the main switch off which interrupts the power to the whole platform with additional stand-alone relay (shown in the figure 4.3 too). That is performed when all the previous attempts failed or after the ARP is stopped. In the worst case that makes the platform stop by friction.

All the relays are connected in the manner – once there is power it is turned on otherwise it is turned off (see chapter 4). Also the communication from Kill Switch Button works on the same principle. Together with the relays independent of all the other systems this means that the operator can stop the platform anytime by Kill Switch Button. The platform also has its own safety mechanisms implemented in the firmware of controlling MCU as mentioned before. These stop the movement automatically in case something goes wrong. Together these safety mechanisms put the possibility of false movement to the minimum (malfunction of any system means the stop of the movement). Only possibility that the ARP gets stuck in an uncontrolled movement is that all the systems get stuck all at once (Control Unit sends signal to move, Drivers Modules still make the rotary magnetic field for BLDC engines, Kill Switch Button and Relays are stuck in the turned on position), which is highly improbable. And there is always the last possibility in the platform's manual main switch located on top of the platform. The idea is to use the whole top lid of ARP as the manual off switch, which would stop the platform when it is overridden by car or with the intervention of an operator. The electrical part is ready, however it is not implemented in the hardware yet.

6

Innovative Features of ARP's Design

After the prototype has been fully described, this chapter focuses on the innovative ideas that are being implemented to the platform. The ARP prototype serves as a tool for ADAS testing itself as well as a tool to test the functionality of these ideas. They are the core of the research of this doctoral thesis together with the following chapters 7 and 8 that are dealing with localization and navigation of the platform. Content of this chapter was also used for the paper [Pub. 8] that can be found among the publications of the doctoral thesis' author.

6.1 Modularity

The first feature that is described here is the modularity. It is used throughout the whole design of the platform from the mechanical as well as electrical hardware and software point of view. Its goals are to maintain different units as small and focused as possible to design a reliable embedded system. If a part of the system malfunctions, it must be easily detected and it must not influence the rest of the system. Plus, the ARP must be stopped in this case. The hardware modularity is obvious from the description of the modules, the software modularity is maintained in each MCU (microcontroller) as well. To keep this overall modularity was one of the key aspects of this research.

The mechanical modularity is important for the maintenance of the device, fast changing of battery pack or wheels modules. It is fully implemented in the mechanical design and it is described in the section 3.2.2. There is also a whole paper [26] dealing with this part of the design.

The modularity in the electrical hardware goes hand in hand with the mechanical design. It is necessary so that the different pieces of hardware, which are changeable, work together. Also it helps to provide safety to the whole design as it is explained in the chapter 4. All the modules must be connected together by the heavy duty connectors that are water resistant as well.

To summarize these up, mechanical and electrical hardware modularity enables changing of the battery pack to prolong the duration of testing depending on the power consumption (see 4.4.1 for detailed power consumption analysis). The platform uses one battery after about 4.6 h of continuous testing so two batteries are needed to cover one whole day of work duration (8 h). Therefore the battery pack has to be easily replaceable. There is also the second important element arising from this modularity which is an easy replacement of Engines Modules. This is important because the platform can use different wheels modules to enable different ways of movement. It was outlined already in the chapter 4.4.6 and it will be further explained in the chapter 6.4.

Software modularity is mainly good for maintaining safety and reliability in the ARP system. Further it makes it easier to keep the system real-time since the different parts of the code are divided into tasks. They are called to be processed by the synchronized processing timers that also track if the tasks are completed in time of the next processing request. The timers are synchronized over the whole system with use of GPS PPS. More information about the modularity in the software solution can be found in the chapter 5 and the safety mechanisms are then listed in the chapter 6.3.

6.2 Cloud Based Communication

Cloud based communication is the next feature explained in this section. It was already outlined in the chapter 4 and now it is described more deeply. The idea is to use an IoT (Internet of Things) network solution for the communication among the wireless units in the ARP system to increase its reliability and range. Its diagram can be seen in the figure 6.1.

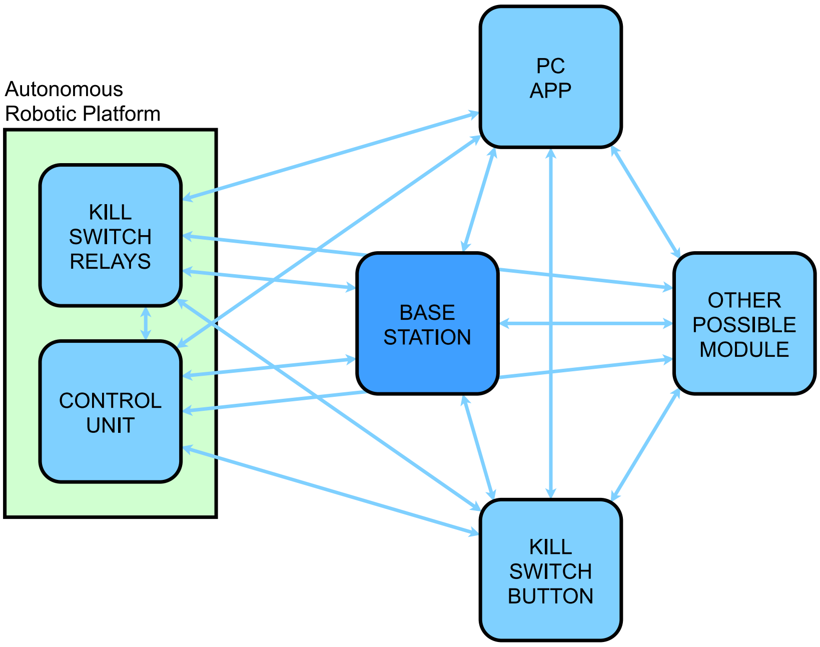


Fig. 6.1: Diagram of the cloud based communication

In such a network, all units can communicate with each other. The main advantage is that it enables retransmission of messages when the transmitter unit does not see the receiver unit of the current message. This feature is called messages hopping through the network. Network can be tuned with the maximum possible number of hops, with number of retransmissions that are attempted when the message is not delivered, etc. XBee modules are used to build such a network and their parameters were explained in the table 4.1. These XBee units can operate in different modes (P2P or different IoT modes). One of the modes that is designed to work as an IoT network is called DigiMesh protocol designed directly by their manufacturer ([11]). It is based on the IEEE 802.15.4 ZigBee standard and enhanced for use in full mesh topology where all the units are used as equal nodes.

This network already solves problems with security (encryption), hops, retransmissions and acknowledgements of messages automatically; therefore the MCUs in the ARP system do not need to use their computation power for low level aspects of communication. On top of these parameters, the ARP system adds other features to increase its reliability. That is additional checksum to skip incorrect messages (because errors can happen even in the UART that connects XBee to MCUs), message ID to ensure correct order of messages and watchdog timers that can detect complete loss of communication.

This network is then used to connect all the modules of the ARP system. The process of information exchange uses Base Station as a coordinator, even if it behaves in the same way as other modules from the network's point of view. It stores variables that need to be sent over the wireless network. In every cycle, all connected devices exchange their data with Base Station, therefore the suggested idea is to use it as the system's cloud. The main advantages of this solution are to keep consistency in the data and to prevent their loss when individual messages are not delivered. Communication iteration is shown in the next list. It works even for more ARPs in the system, which is the next advantage. If more ARP were used, it would not need any system modifications because all the communication goes through the cloud. That increases the modularity of the system as well:

1. Every ARP sends data to Base Station's cloud.
2. All Kill Switch Button devices send data to the cloud.
3. PC Application sends its data to the cloud.
4. Base Station saves all the data with current timestamp. It keeps defined history of data for future use if necessary.
5. Base Station broadcasts data from its cloud and other devices read what they need.

From the overall perspective, this solution decreases data loss and increases range of communication. Messages can be retransmitted over more devices which makes the probability of undelivered message lower. Then Base Station can be placed in the middle of the testing area which can increase the testing area. For example when the VUT where PC Application and Kill Switch Button are located needs to communicate with the ARP (either Control Unit or Kill Switch Relays Unit) it can send the message through Base Station. This means that the range of communication is nearly double than with P2P communication where no hop node can be used. Also, the ARP does not need to use spacious antennas that would make it harder to get overridden by car. On the other hand, this type of network reduces data throughput, because there must be space for retransmissions. This is not an issue because the throughput is more than sufficient for the amount of data that need to be exchanged. The improvements that this type of network brings to the system are shown in the table 6.1. The measurement was performed with the same parameters for both network types (distance of modules was 50 m for data loss measurement; the worst loss rate for maximum range was set to 10 %).

Tab. 6.1

Parameters	P2P network	IoT network
Data loss rate	4.27 %	2.18 %
Maximum range	113 m	185 m

Tab. 6.1: Comparison of parameters of different network types

6.3 Automatic Safety Mechanisms

The safety of the ARP is another key design feature that is being implemented from the beginning to every part of the system. The safety is important because in case of failure, the heavy moving ARP can cause damage to its surroundings. The different safety mechanisms are already described in the chapters 4 and 5 in the sections according to the modules where they are implemented. In the table 6.2, there is a summary of them giving an idea on how the overall safety is ensured.

The whole system is designed and tested to be as safe and reliable as possible. All the modules and sensors run auto diagnostics and can be reset automatically if error occurs. However, sometimes an error needs the test supervisor's attention to be fixed (e.g. loss of GPS signal, loss of communication or mechanical block of ARP). In this case, the system must safely stop until error's cause is eliminated.

Tab. 6.2

Safety mechanism	Implementation in	Resulting action
Battery low voltage detection (including PC battery)	Battery Pack, PC Application, Base Station, Kill Switch Button	Preceding warning ARP shutdown

Table continues on the next page . . .

Tab. 6.2 – continuation

Safety mechanism	Implementation in	Resulting action
High temperature detection	Control Module, Drivers Modules, Engines Modules	Preceding warning ARP shutdown
Hardware MCU watchdogs (MCU freezing detection)	Control Module, Base Station, Kill Switch Button	ARP stopping
Software watchdogs (wired/wireless communication loss)	Every MCU firmware	ARP stopping
Engines drivers watchdogs (communication from Control Module loss)	Drivers Modules	ARP stopping
Blocked movement detection (ARP should move but it does not)	Control Module	ARP stopping
Detection of leaving the testing area (using ARP's localization)	Control Module	ARP stopping
Manual kill switch in PC Application	PC Application	ARP stopping
Manual kill switch in Kill Switch Button	Kill Switch Button, Kill Switch Relays Unit (see section 5.4.3)	ARP stopping
Manual kill switch on top of the ARP	Main switch of the ARP located at the battery pack	ARP shutdown

Tab. 6.2: ARP safety mechanisms

These safety mechanisms work as a continuous diagnostics tool and they put the possibility of the false movement to the minimum (malfunction of any system means stopping the ARP movement or the ARP shutdown). If safety mechanisms fail, there is a possibility to use the Kill Switch Button as a wireless option to stop the movement. The procedure of how the ARP is stopped is shown in the chapter 5.4.3. Even if this kill switch fails together with the previous system fail at the same time which is highly improbable, there is always the last possibility in the platform's manual main switch (last item in the table). One of the ideas on how to improve this main switch is to implement it to a larger area on the top of the platform's lid so it is easier to be hit. It would be also triggered when a car overrides the platform. The process that leads to the ARP's shutdown through the power relays is shown in the chapter 4 in the figure 4.3.

In the future this area is also open for improvements. Overall long-term testing of platform's endurance, water resistance, vibration resistance, being driven over by car resistance or reliability should be performed.

6.4 Omnidirectional Movement

From the beginning of the research, one of the main ideas was to make the ARP able to perform an omnidirectional movement. Firstly, the prototype was designed with the straight wheels modules where turning was possible only by difference in velocities on the left and on the right side of the ARP – principle of vehicle with tracks. The computation of these velocities can be seen in the figure 6.2 and in the equations 6.1, 6.2 and 6.3 where r is radius of turning, $v_{x|req}$ is requested forward velocity, $\omega_{\psi|req}$ is requested angular velocity in yaw manner and d_y is lateral distance of the wheels. Turning point is located on the lateral axis intersecting the geometrical centre of ARP. This type of movement was sufficient to perform easy scenarios with the straight trajectories where difference in the velocities was used to compensate the small lateral deviation from this trajectory. This lateral deviation arises from geometrical imperfection of ARP as well as from influence of outer effects.

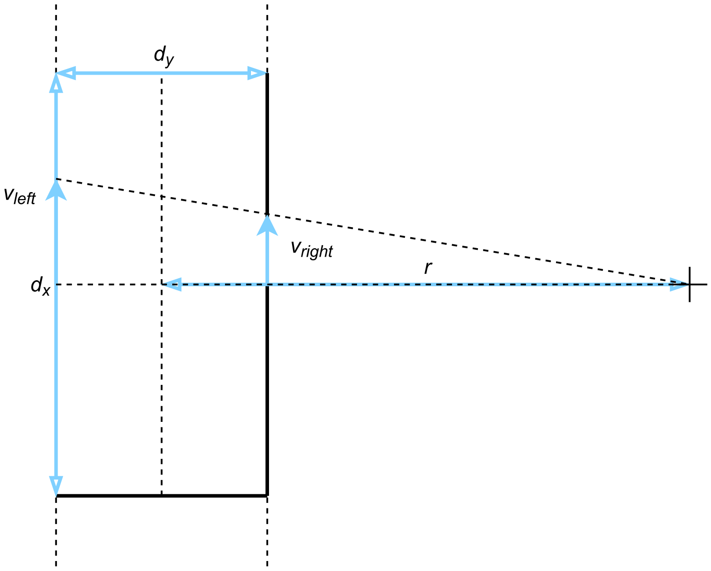


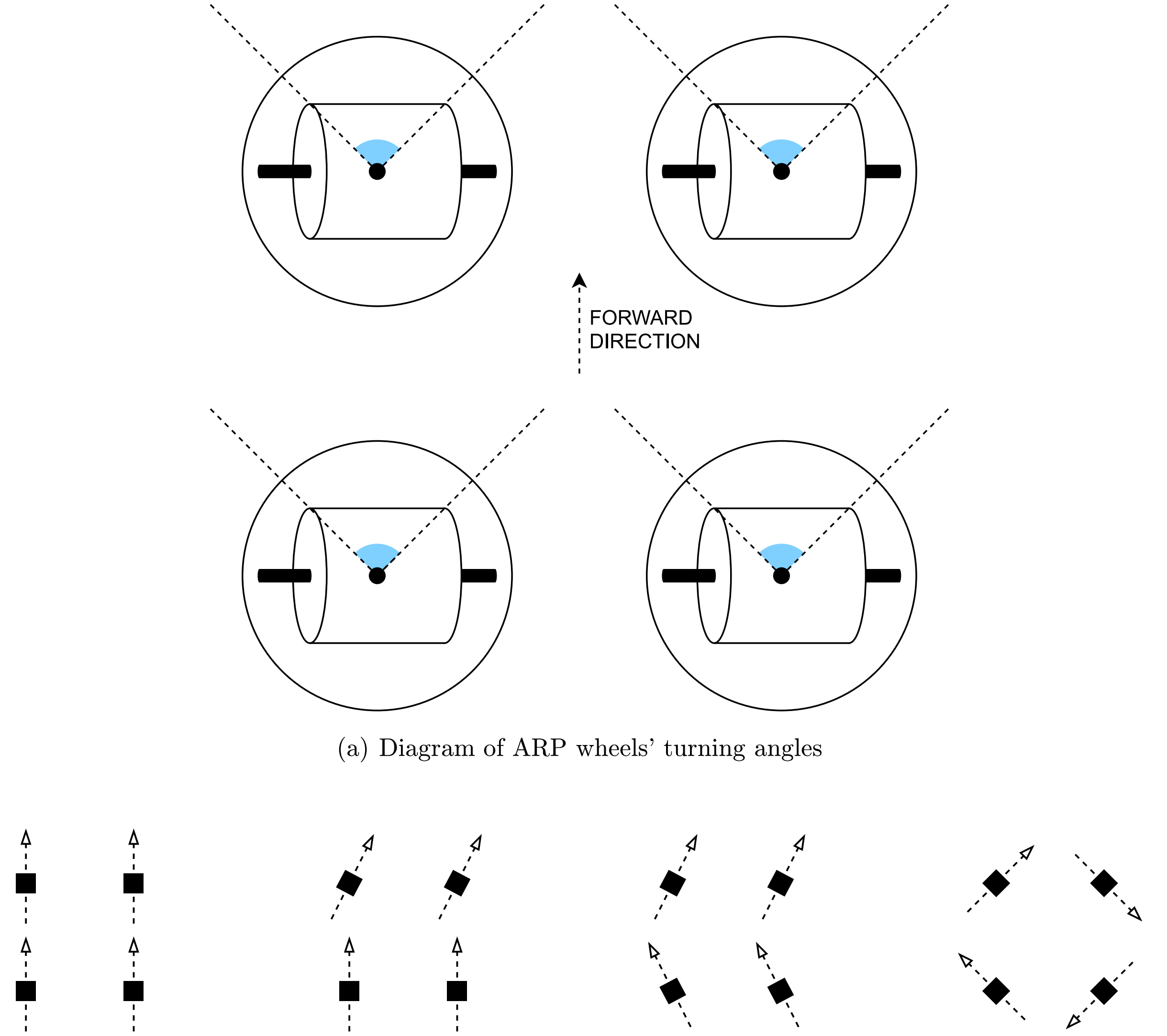
Fig. 6.2: Diagram of straight movement

$$r_{req} = \frac{v_{x|req}}{\omega_{\psi|req}} \tag{6.1}$$

$$v_{left|req} = \omega_{\psi|req}\left(r_{req} + \frac{d_y}{2}\right) \tag{6.2}$$

$$v_{right|req} = \omega_{\psi|req}\left(r_{req} - \frac{d_y}{2}\right) \tag{6.3}$$

However, it was not sufficient to fulfil the goals of this research to develop a more universal platform. Therefore, the omnidirectional movement was introduced. Traditional platforms usually move in the car's manner, which still makes some manoeuvres hard or impossible. On the other hand, this type of movement makes this new platform highly variable to do scenarios with all the different dummies (pedestrians, cyclists, balloon cars, etc.). The movement possibilities can be seen in the figure 6.3. The hardware design of modules and the selected engines are described in the chapter 4.4.6. The ARP can use electrical braking or mechanical braking for slowing down. Electrical one is preferred because it does not wear the wheels out and it also supports recuperation of power back into the main battery through Drivers Modules. But sometimes it is not sufficient for achieving the required deceleration by ADAS standards. This mechanical braking is controlled by braking servomechanisms connected to Control Unit of ARP. The same servos can also be used for emergency braking as it was explained in the section 5.4.3.



(a) Diagram of ARP wheels' turning angles

(b) Possibilities of ARP movement

Fig. 6.3: Omnidirectional movement explanation

A diagram of the omnidirectional movement in the figure 6.3 has two parts. In the part (a), there are four wheels modules with wheels using HUB BLDC engines (see section 4.4.6).

The wheels are attached to the module with a vertical spindle which makes the steering possible. The steering is controlled by a servomechanism with a gearing giving the turning angle of -45° to 45° to every wheel (blue angles drawn in the figure). In the part (b), there are movement options of the ARP with these modules (arrows indicate the directions of the wheels steering angles). ARP can drive forward, turn in the car's manner, turn with both front and back wheels, rotate around the centre vertical axis or even move to the side (this last option is not shown in the figure).

In the two following figures (6.4 and 6.5) and corresponding equations, there is the geometry of the different movement types explained. It is necessary to set different angles and velocities for inner and outer wheels so the ARP's movement is smooth. Symbols used in the equations are explained in the nomenclature and subscripts section.

- Equations for straight movement:

$$\omega_{\psi|req} = 0 \tag{6.4}$$

$$r_{req} = 0 \tag{6.5}$$

$$\alpha_{left|req} = 0 \tag{6.6}$$

$$\alpha_{right|req} = 0 \tag{6.7}$$

$$v_{left|req} = v_{x|req} \tag{6.8}$$

$$v_{right|req} = v_{x|req} \tag{6.9}$$

- Equations for movement in the car's manner (only front wheels are steerable). Turning point is located on the lateral axis intersecting rear wheels of the ARP:

$$r_{req} = \frac{v_{x|req}}{\omega_{\psi|req}} \tag{6.10}$$

$$\alpha_{left|req} = \frac{\pi}{2} - \arctan\left(\frac{r_{req} + \frac{d_y}{2}}{d_x}\right) \tag{6.11}$$

$$\alpha_{right|req} = \frac{\pi}{2} - \arctan\left(\frac{r_{req} - \frac{d_y}{2}}{d_x}\right) \tag{6.12}$$

$$r_{left|req} = \frac{r_{req} + \frac{d_y}{2}}{\sin\left(\frac{\pi}{2} - \alpha_{left|req}\right)} \tag{6.13}$$

$$r_{right|req} = \frac{r_{req} - \frac{d_y}{2}}{\sin\left(\frac{\pi}{2} - \alpha_{right|req}\right)} \tag{6.14}$$

$$v_{left|req} = \omega_{\psi|req} r_{left|req} \tag{6.15}$$

$$v_{right|req} = \omega_{\psi|req} r_{right|req} \tag{6.16}$$

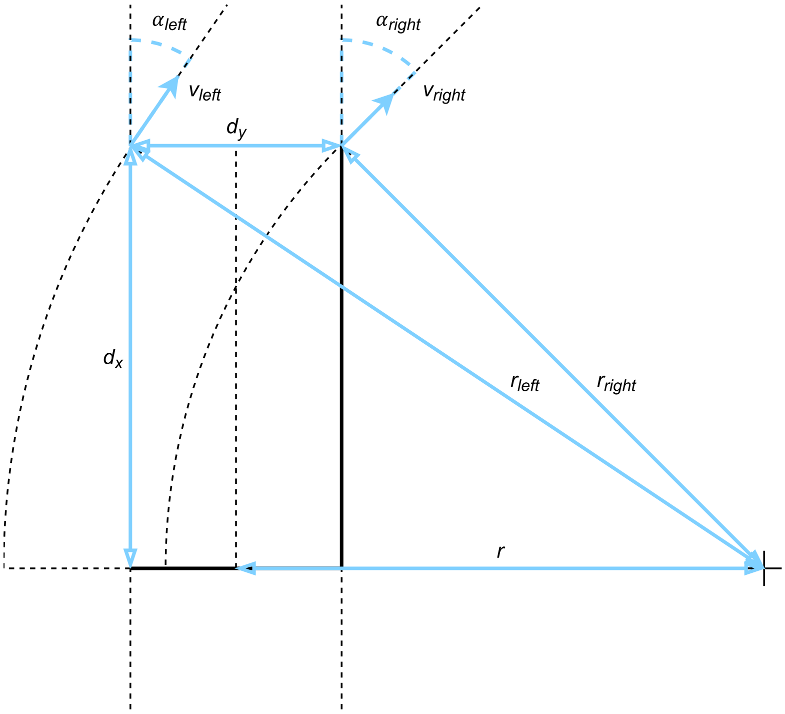


Fig. 6.4: Diagram of omnidirectional movement with front steerable wheels

- Equations for movement with all steerable wheels. Turning point is located on the lateral axis intersecting the geometrical centre of ARP:

$$r_{req} = \frac{v_{x|req}}{\omega_{\psi|req}} \tag{6.17}$$

$$\alpha_{left|req} = \frac{\pi}{2} - \arctan\left(\frac{2r_{req} + d_y}{d_x}\right) \tag{6.18}$$

$$\alpha_{right|req} = \frac{\pi}{2} - \arctan\left(\frac{2r_{req} - d_y}{d_x}\right) \tag{6.19}$$

$$r_{left|req} = \frac{r_{req} + \frac{d_y}{2}}{\sin\left(\frac{\pi}{2} - \alpha_{left|req}\right)} \tag{6.20}$$

$$r_{right|req} = \frac{r_{req} - \frac{d_y}{2}}{\sin\left(\frac{\pi}{2} - \alpha_{right|req}\right)} \tag{6.21}$$

$$v_{left|req} = \omega_{\psi|req} r_{left|req} \tag{6.22}$$

$$v_{right|req} = \omega_{\psi|req} r_{right|req} \tag{6.23}$$

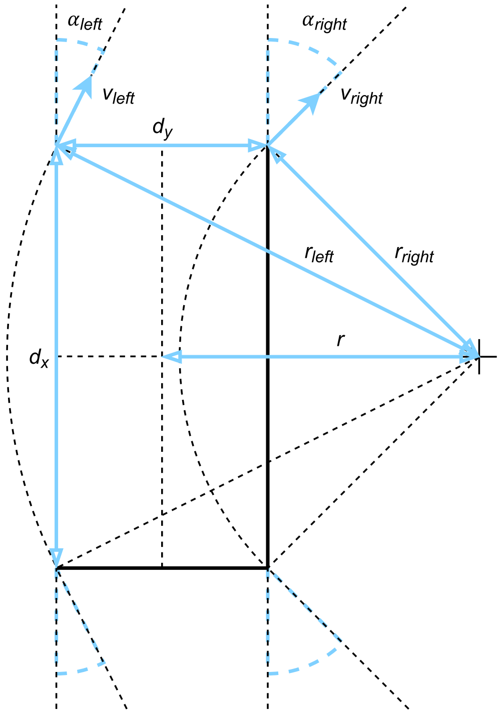


Fig. 6.5: Diagram of omnidirectional movement with all steerable wheels

- Equations for spot rotation movement (sign of steering angle and velocity for each wheel must be set according to the direction of rotation):

$$v_{x|req} = 0 \tag{6.24}$$

$$r_{req} = 0 \tag{6.25}$$

$$\alpha_{left|req} = \frac{\pi}{2} - \arctan\left(\frac{d_y}{d_x}\right) \tag{6.26}$$

$$\alpha_{right|req} = \arctan\left(\frac{d_y}{d_x}\right) - \frac{\pi}{2} \tag{6.27}$$

$$v_{left|req} = \omega_{\psi|req}\sqrt{\frac{d_x^2}{4} + \frac{d_y^2}{4}} \tag{6.28}$$

$$v_{right|req} = -\omega_{\psi|req}\sqrt{\frac{d_x^2}{4} + \frac{d_y^2}{4}} \tag{6.29}$$

This solution brings advantages for the movement options. On the other hand, the control mechanisms are more challenging. It must ensure correct turning of the wheels to follow the requested trajectory as well as not to allow the wheels to be turned in a wrong way which could result to Engines Modules damage. That will be described more deeply in the chapter 8.

From the mechanical point of view, the interesting feature is the suspension of the wheels as well. This is done by the torsion rods (see [26]). They enable to set the platform to carry the requested weight and also to hide the wheels inside the platform if the weight gets too heavy (usually when there is a car overriding the platform). They also save the space so the platform can preserve the overall low height.

7

Localization Possibilities

This chapter together with the subsequent one (8) explains one of the main goals of the research – to make the ADAS testing more efficient (by reducing the testing time and the required human supervision) or even to make it fully autonomous. After the features that ensure safety and reliability of the prototype that were described before, localization of the ARP is the‘next possibility where there is a huge space for improvements and applying new ideas into the field. Together with the ARP's control and car synchronization (see 8) it is also the core of this doctoral thesis.

This research aims to explore different ways of localization of the ARP and to try different approaches of position acquiring. They can enable testing in various areas by using non-standard sensors combined with software filtering to get the ARP's position with sufficient precision. Standard sensors that are usually used for the robot's localization are odometry, IMU and GPS. The ARP also uses these sensors plus it implements and tests different sensors such as mouse camera or ultrasonic distance measurement. There are papers that were written by the author of this doctoral thesis using the content of this chapter and their references can be found under the entries [Pub. 9], [Pub. 10] and [Pub. 11] among the list of author's publications.

Many solutions for ADAS testing (see chapter 2) use precise but very expensive differential GPS systems as said before. Also, they must be equipped with large and spacious antennas to make them work that makes the ability of the ARP to be overridden by car more difficult. This thesis' goal from the localization point of view is to study different possibilities of navigation for robots capable of performing the testing scenarios. It focuses on commonly available sensors and a software data fusion. The results of several different methods are compared in this chapter too.

7.1 Odometry, IMU and GPS Sensors

Firstly, the ARP uses these standard sensors to get its position. The aim is not to use industrial differential GPS units that must use spacious antennas. These antennas make the ability of the platform to be overridden by car much more difficult. The list of the sensors is described here, followed by the figures with graphs showing the results (7.1, 7.2, 7.3, 7.4, 7.5 and 7.6). Differential GPS system is used as a reference for these graphs. Namely, it is the Tersus BX305 solution ([35]). The movement was controlled manually by the RC transmitter to collect these data; whilst the ARP performed movement in elliptical manner during this measurement.

7.1.1 Odometry

The odometry means a procedure where the position is computed from movement measured on each wheel of a robot. Here it is taken from the pulses measured by Hall Effect sensors (3 per each wheel). The resulting distance (d_{meas}) and velocity (v_{meas}) for each side (left and right) is computed from the measured pulse count ($pulses_{meas}$), diameter of the ARP's wheels ($diameter$) and time of measurement (Δt_{meas}) – equations 7.1 to 7.4:

$$d_{left|meas} = \frac{pulses_{left|meas}}{3} \times \pi \times diameter \tag{7.1}$$

$$d_{right|meas} = \frac{pulses_{right|meas}}{3} \times \pi \times diameter \tag{7.2}$$

$$v_{left|meas} = \frac{d_{left|meas}}{\Delta t_{meas}} \tag{7.3}$$

$$v_{right|meas} = \frac{d_{right|meas}}{\Delta t_{meas}} \tag{7.4}$$

Forward velocity ($v_{x|meas}$) and angular velocity ($\omega_{\psi|meas}$) of the platform are then computed from geometry of the ARP, where steering angles and left and right velocities of wheels are the inputs. Geometry is different for various types of movement that are described in the chapter 6 in the section concerning the omnidirectional movement. When the ARP uses straight wheels modules, the computation is much easier and it is shown in the following equations 7.5 and 7.6:

$$v_{x|meas} = \frac{v_{left|meas} + v_{right|meas}}{2} \tag{7.5}$$

$$\omega_{\psi|meas} = \frac{v_{left|meas} - v_{right|meas}}{\frac{d_y}{2}} \tag{7.6}$$

Reference velocity ($v_{x|ref}$) and reference angular velocity ($\omega_{\psi|ref}$) used in the graphs are delayed because they were filtered to make the graphs more readable. They were computed by deriving the position from the reference differential GPS which makes them noisy. Computed forward velocity ($v_{x|meas}$) in the figure 7.1 has nice course and it is a good input for Kalman filter; however angular velocity ($\omega_{\psi|meas}$) computed from odometry in the figure 7.2 is quite noisy and it is better to use angular velocity from IMU as it will be shown later. Angular velocity is shown in [°/s] instead of [rad/s] (which is the standard SI unit for this quantity) for better graph readability. Values given by the odometry are in the reference frame of the platform's movement (longitudinal and lateral direction of its current heading).

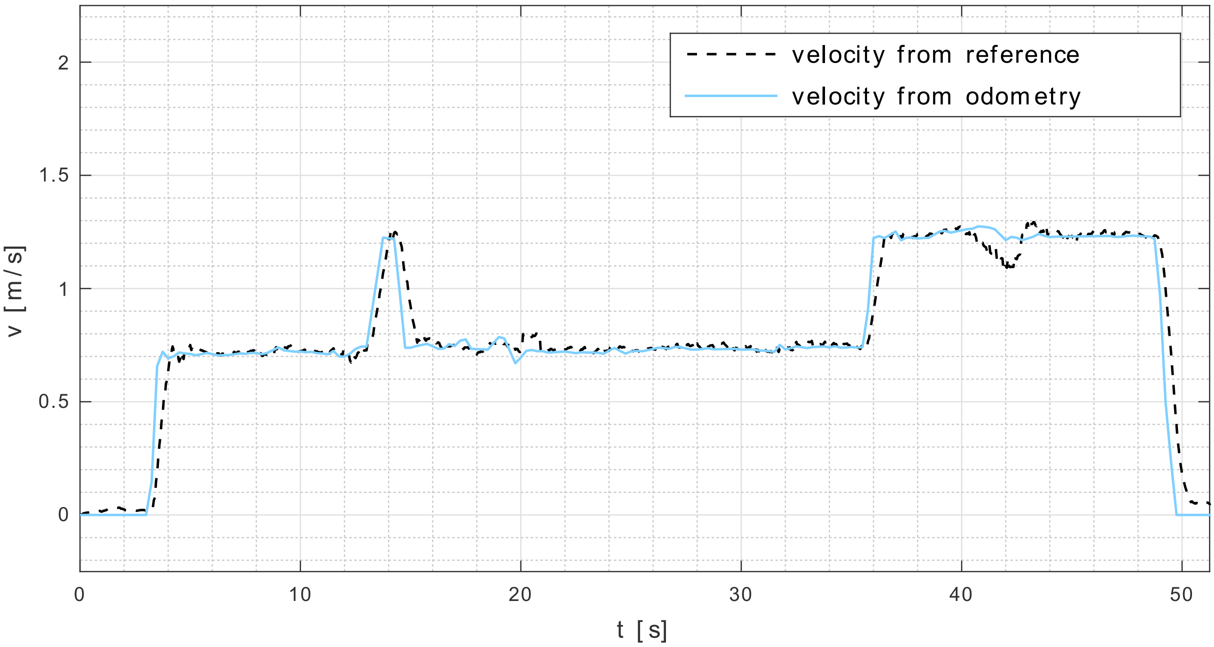


Fig. 7.1: Graph of odometry velocity measurement

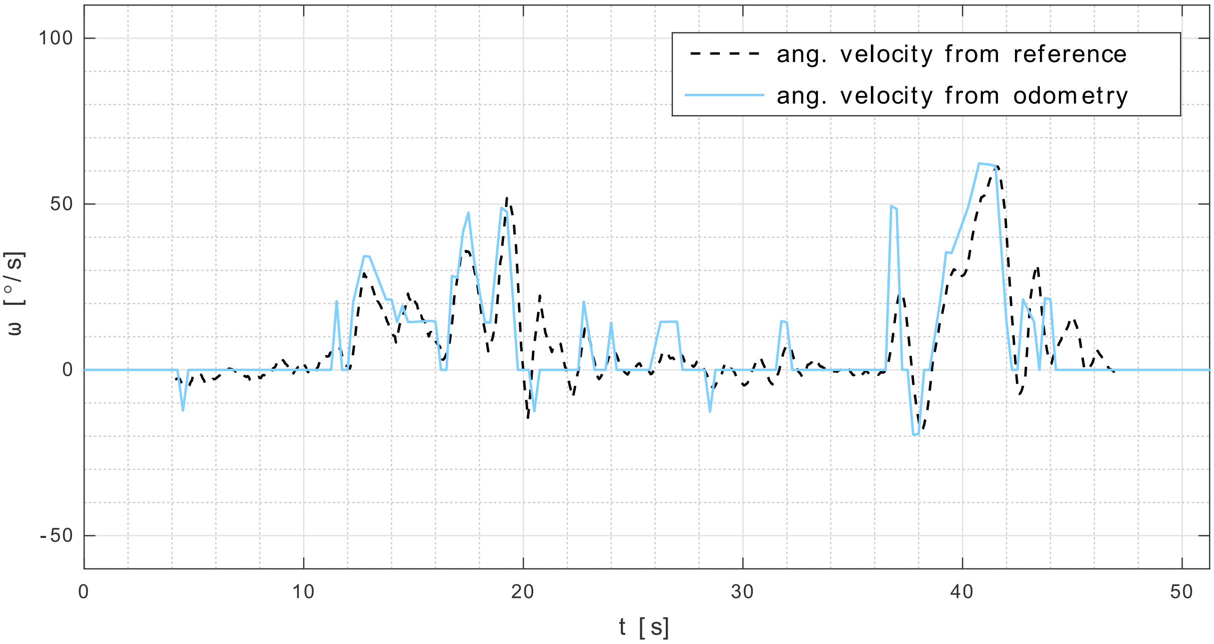


Fig. 7.2: Graph of odometry angular velocity measurement

7.1.2 IMU

The IMU stands for Inertial Measurement Unit and it is collecting data from sensors good for measuring acceleration, angular velocity and heading. The sensors that are used for that are accelerometer, gyroscope and magnetometer. ARP uses a device called BNO055 ([8]) that combines all of these sensors. There are heading in yaw manner ($\Phi_{\psi|meas}$), angular velocity in yaw manner ($\omega_{\psi|meas}$) and acceleration in longitudinal direction ($a_{x|meas}$) that are the outputs from this measurement good for the localization of ARP.

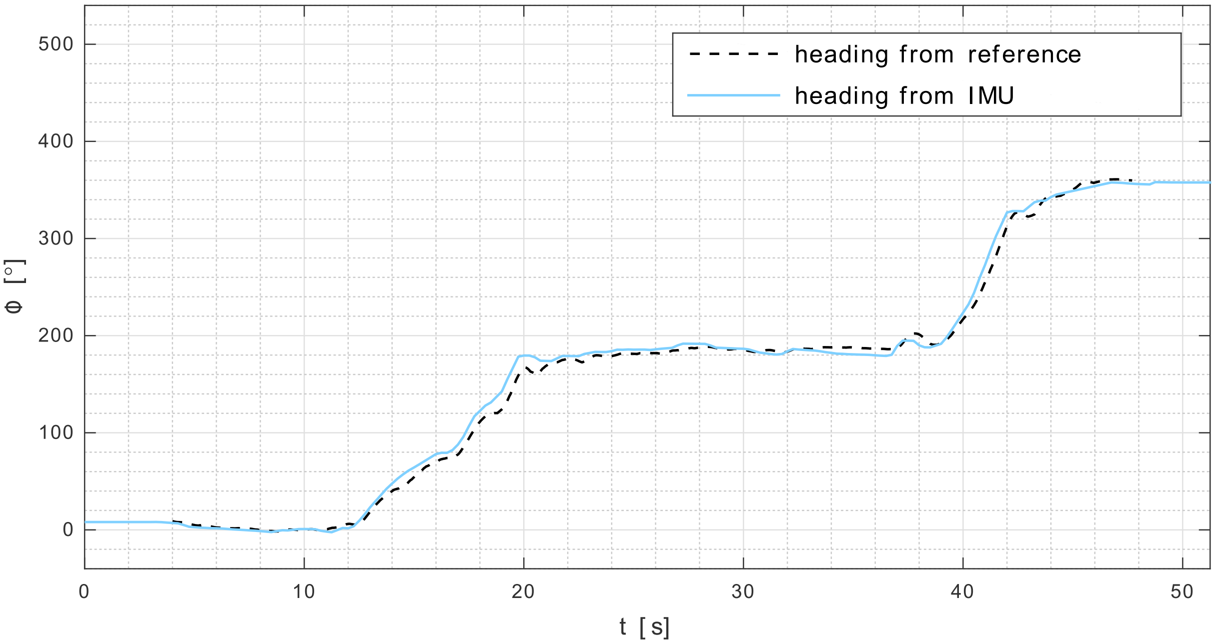


Fig. 7.3: Graph of IMU heading measurement

Reference heading ($\Phi_{\psi|ref}$) and reference angular velocity ($\omega_{\psi|ref}$) used in the graphs were computed from the reference differential GPS as it was done in the case of odometry. Measured heading in the figure 7.3 and angular velocity in the figure 7.4 are later used as inputs for Kalman filter. Heading is shown in [°] instead of [rad], angular velocity is shown in [°/s] instead of [rad/s] for better graphs readability. Acceleration in longitudinal direction that is also an input for data fusion is not so important for localization and its graph can be found in the attachment E in the figure E.1 if needed. Heading is referenced to absolute Earth's coordinates (latitude and longitude), other values are referenced to the platform's movement. There is one more important remark which is to take into account the difference between direction to the magnetic north pole of Earth and direction to the Geographic North Pole. It is different in the various locations on Earth and this compensation is necessary for the correct IMU operation.

These two localization methods – odometry and IMU – accumulate error in measured distance growing with time so their data cannot be used for more difficult trajectories without correction from other sensors that measure absolute position (such as GPS).

Nevertheless, it works for short straight distances where the position is reset after each scenario and can be corrected manually by a test supervisor if needed.

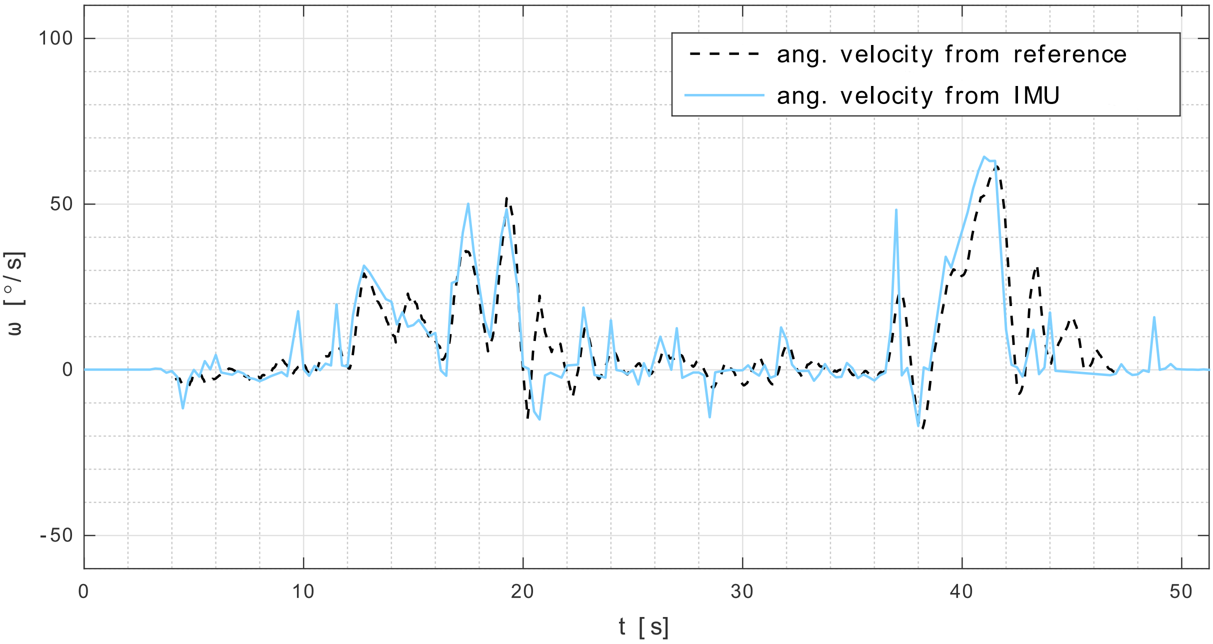


Fig. 7.4: Graph of IMU angular velocity measurement

7.1.3 GPS

GPS unit works as a standard sensor to give information about the absolute position of ARP. Without this sensor, the position is acquiring bigger error with increasing time as it was mentioned in odometry and IMU description. Devices for ADAS testing usually use differential GPS systems. During the research of this ARP, basic GPS unit was used. Its error is compensated with another GPS located on the static Base Station. It turned out that this solution can work when these two GPS are not too far from each other. The static one sends its current offset from its initial measured position to ARP and this offset is used to compensate the error offset of the GPS on the platform. The GPS units that were used for this testing are Adafruit Ultimate GPS devices with MTK3339 chipset ([2]).

The position that is given by the cooperation of these GPS sensors is shown in the figures 7.5 and 7.6. There are reference position (d_{ref}), position without the compensation from Base Station and position with the compensation (d_{meas}). There are these measured quantities that are used as inputs for Kalman filter from the GPS measurement: distance in latitude (south-to-north) direction ($d_{lat|meas}$) and distance in longitude (west-to-east) direction ($d_{long|meas}$). Obviously, the outputs from GPS sensors are given in degrees of latitude/longitude; therefore, they are converted to values in meters relative to the initial Base Station position so they can be combined with the values from other sensors. Values measured by GPS are referenced to the Earth's coordination system (latitude and longitude).

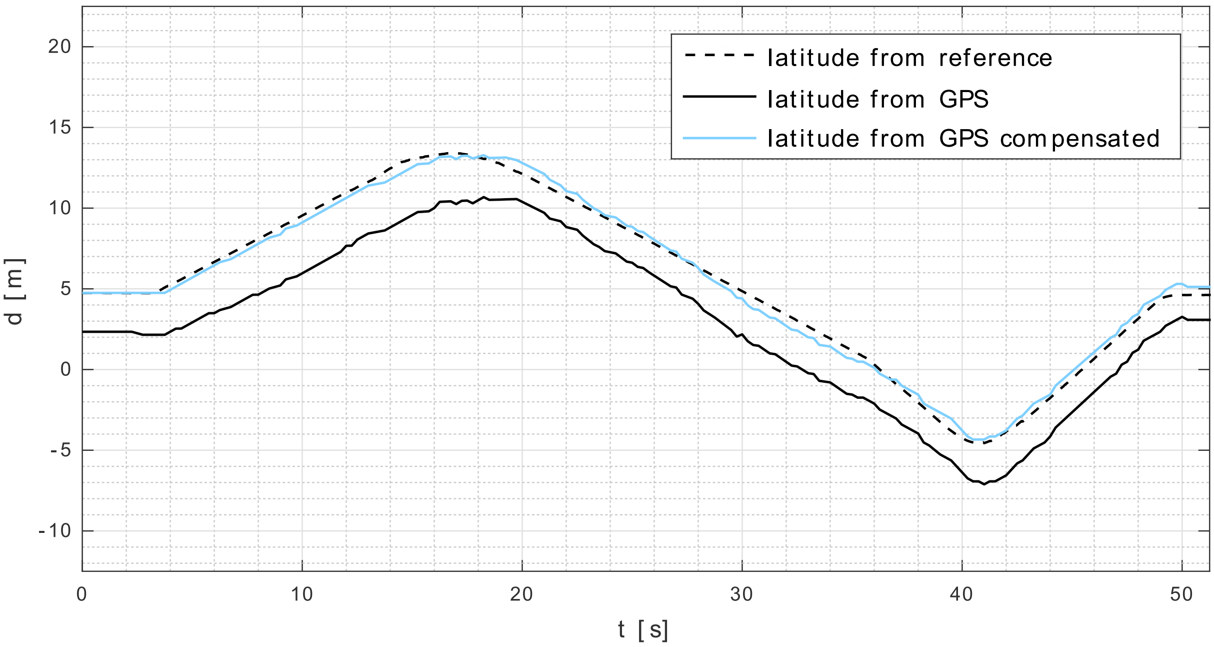


Fig. 7.5: Graph of GPS latitude position measurement

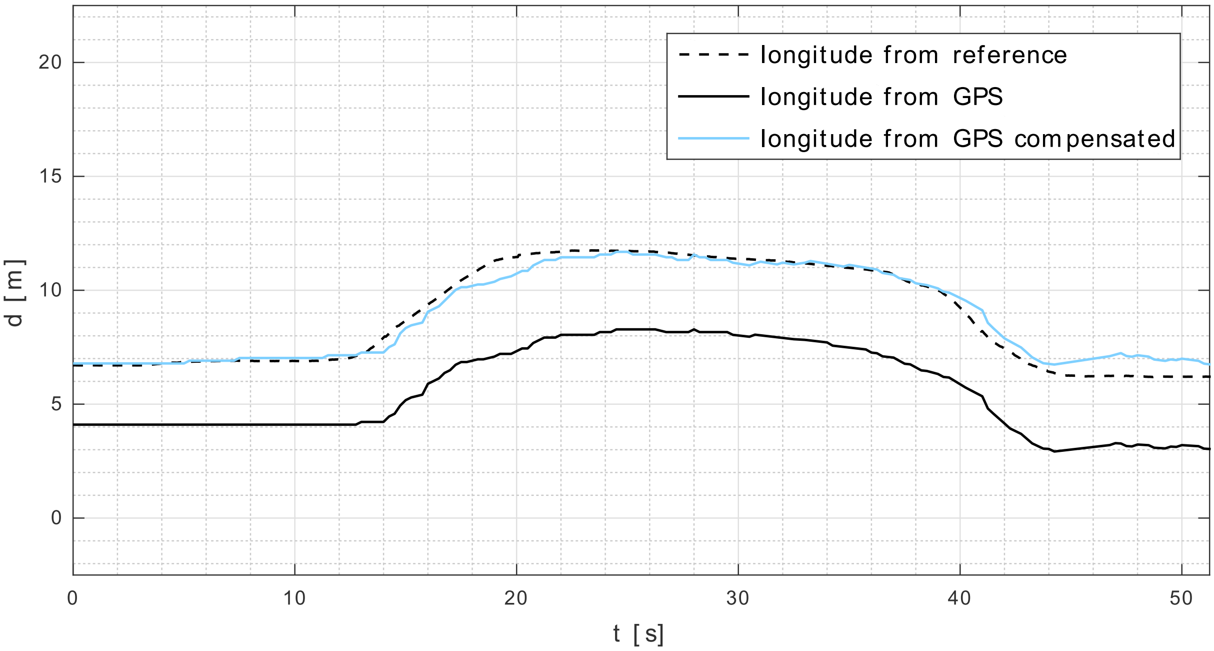


Fig. 7.6: Graph of GPS longitude position measurement

This section gives opportunities for the future research, where different GPS solutions can be tested and these results can be compared. There are connectors on Base Station and on the ARP for connection of external GPS sensors that make these tests easy to perform. One of the devices that is going to be tested in the near future is the u-blox GPS system (see [36]). It gives more precise position than the basic Adafruit GPS units but it can still work with non-spacious antennas that are required by more sophisticated industrial systems.

Overall performance of all these standard sensors and how they work with the Kalman filter is shown in the section 7.4.

7.2 Mouse Camera Positioning

Apart from the standard sensors that were described so far in this chapter, the ideas were to use different approaches to improve variability of the platform's localization. The mouse camera sensor (specifically ADNS-3080 – [4]) is the first tested option. The sensor works on the principle of computer mouse, where the camera is watching the ground beneath the ARP and it can compute the motion of the platform from the picture shift. Calibration of the sensor is needed and additionally, if the sensor is located beneath the platform, there has to be a light source mounted too. It is also necessary that the surface has some kind of pattern on it to minimize the lags in sensor's measurement. See [18] for more information about the sensor – how it works with different surfaces and how to perform its calibration.

This sensor can give additional useful inputs for the Kalman filter or alternatively, it can be used for short distance scenarios performed in space inside buildings where the GPS signal is not an option. It is not a replacement for absolute position sensors because error is accumulated over time but it can work for limited duration. The position must be then corrected in regular intervals (manually or by different sensor). Its performance vs. the filtered position can be seen in the figure 7.7.

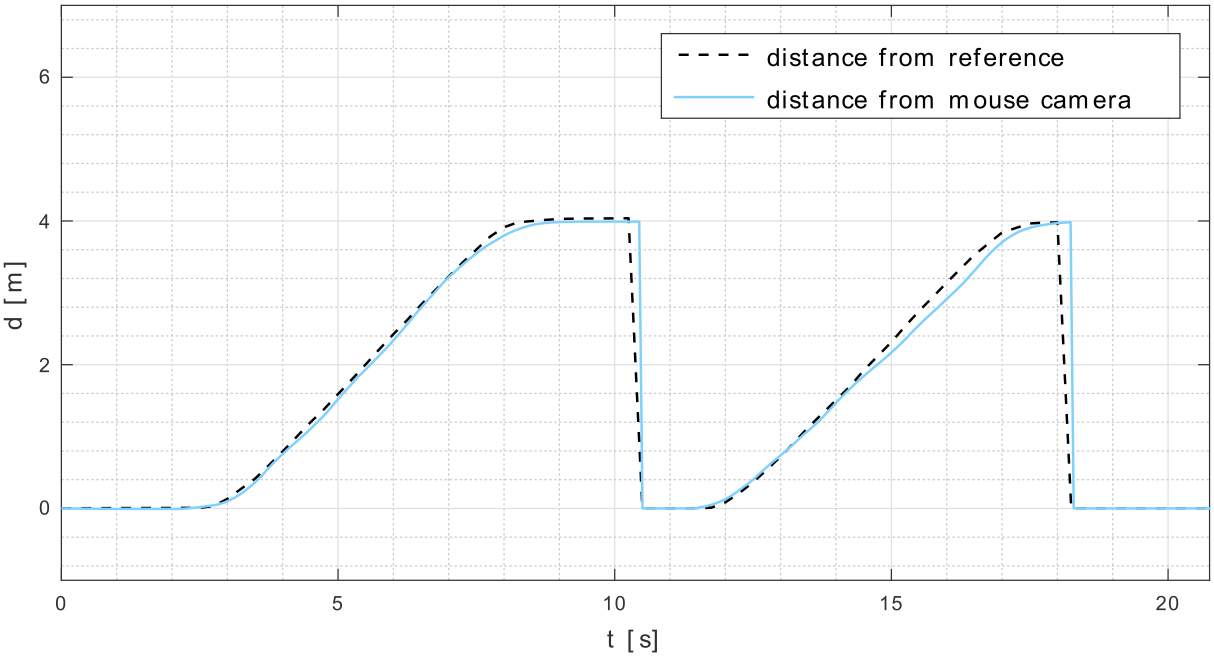


Fig. 7.7: Graph of mouse camera measurement

This sensor gives values of distance change per time duration referenced to the movement of ARP ($\Delta d_{x|meas}$ and $\Delta d_{y|meas}$). The distances are shown in the graph as the absolute travelled distances combined in both directions so it can be compared to the reference distance given by GPS which, on the other hand, gives result referenced to the Earth's coordinates. The graph shows a scenario where the ARP moves forward and backward whilst it is controlled by the navigation controller that is described in the chapter 8. The evaluation of multiple 4 m long scenario measurements showed that the distance error is kept in range of 5 % for individual scenario trajectory.

7.3 Ultrasonic P2P Distance Measurement

There is the second non-standard sensor that has been tested during the research described here. It is the P2P (Point-to-Point) ultrasonic distance measurement device. The ultrasonic distance measurement principle is well known, however the device developed during this research is built on the P2P principle. There are 4 papers written by the author of this doctoral thesis related to this topic: [Pub. 1], [Pub. 2], [Pub. 3] and [Pub. 4].

First, it was necessary to create units containing ultrasonic transmitter and receiver that are capable to communicate between themselves and to measure time of communication that can be used to compute the distance. In the figure 7.8, there is an explanation of the principle. One unit behaves as a master (initiates the communication) and the second one behaves as a slave (responds to requests). With this idea, it is possible to use one master unit which periodically transmits requests to several slave units and measures its distance to all of them. It is obvious that the requests and responses must contain data communication (ID) so the master unit can resolve which slave unit participates in the communication. ID is also needed for the slave unit to know whether to respond to the request or not.

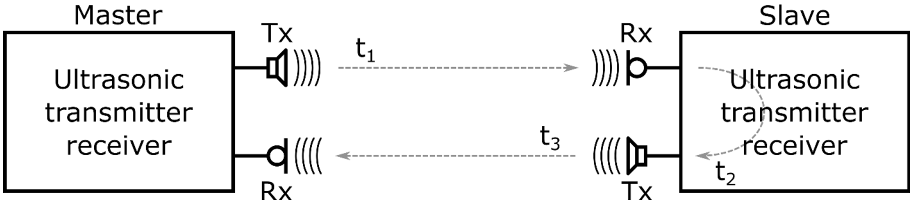

Master
Ultrasonic transmitter receiver
Tx
Rx
t1
t3
Rx
Tx
Slave
Ultrasonic transmitter receiver
t2

Fig. 7.8: Diagram of P2P ultrasonic distance measurement

The speed of sound in air is not constant and that can be compensated with the temperature sensor built into every unit. In the following equations 7.7, 7.8 and 7.9, distance computation and temperature compensation are shown. The quantities are as follows: t_{meas} is total measured time by master unit, t_2 is time for slave unit to respond (must be known by master), v_{sound} is speed of sound, T is temperature of air in [°C] and d_{meas} is measured distance.

$$t_{meas} = t_1 + t_2 + t_3 \tag{7.7}$$

$$v_{sound} = 20.0457\sqrt{273.15 + T} \tag{7.8}$$

$$d_{meas} = \frac{v_{sound}\left(t_{meas} - t_2\right)}{2} \tag{7.9}$$

The easiest way to code data information into communication without using any modulation is to divide the message to bit windows with period T_B. This principle is shown in the figure 7.9. Every message starts with Start Bit (S) of the defined length. Logical "1" is then coded as presence of 40 kHz pulses in the bit window; whilst logical "0" is coded as non-presence of pulses. Time of pulses burst (t_P) in one bit window (and so number of pulses N_P) was set experimentally. It must be possible to transmit the message over the maximum distance (lower number of pulses leads to the shorter maximum distance because of the capacitive character of the ultrasonic transducer). The current concept can operate up to the distance of 5 m and uses 20 pulses for one bit of logical "1". Then the bit period must have the sufficient length so no echo of one bit can be received in the next bit. The prototype works with the T_B equal to 5 ms; thus the maximum bit rate is 200 bit/s. Period of one word in the message (T_W) depends on the number of bits. In this case there are five bits including the start bit. One communication from the master unit to the slave unit and back then takes 50 ms. Time of flight of the signal can get up to 30 ms so the repetitive rate of the measurement is about 80 ms.

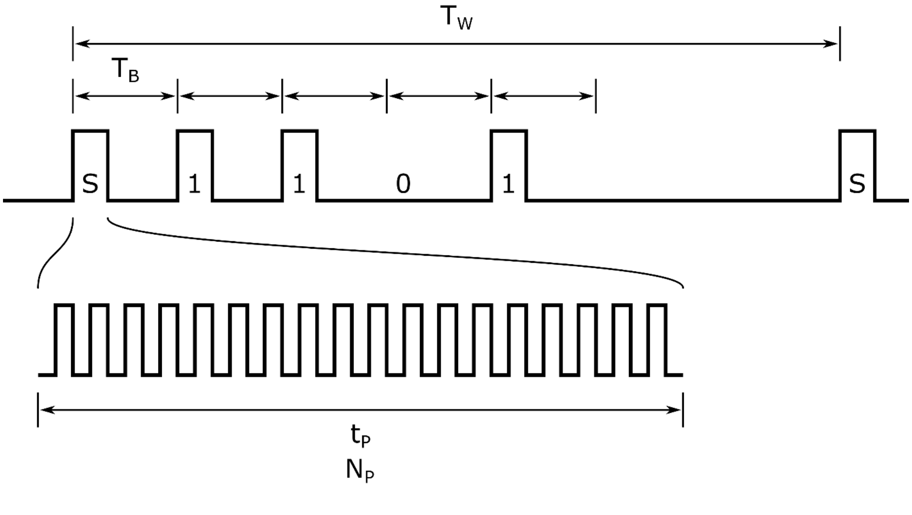


Fig. 7.9: Timing of P2P ultrasonic communication

Tab. 7.1

Distance [m]	Success rate	Distance error
1	96 %	0.8 %
3	78 %	1.1 %
5	35 %	1.8 %

Tab. 7.1: P2P ultrasonic distance measurement results

The measurement was performed between two units using the master slave model. A prototype hardware in the laboratory conditions was used to explore error of measured distance and success rate of communication. Measured distances were 1, 3 and 5 m. Each measurement contains 10000 attempts of transmission between one master unit and one slave unit. The results are shown in the table 7.1.

It is apparent from the table that success rate of communication drops significantly with growing distance. Reason is that the received signal disappears under the noise level. Therefore, simulations to understand this process were performed and repair codes were implemented. Plus, evaluation of received bit can be done in frequency domain instead of time domain to further improve the decision whether the signal is noise or communication. In the figure 7.10, there is a result of measurement performed on the receiver side. Schematics of transmitter and receiver or results of measurements and simulations can be found in the attachment F. More detailed information about these methods can be obtained from the papers mentioned in the introduction to this section.

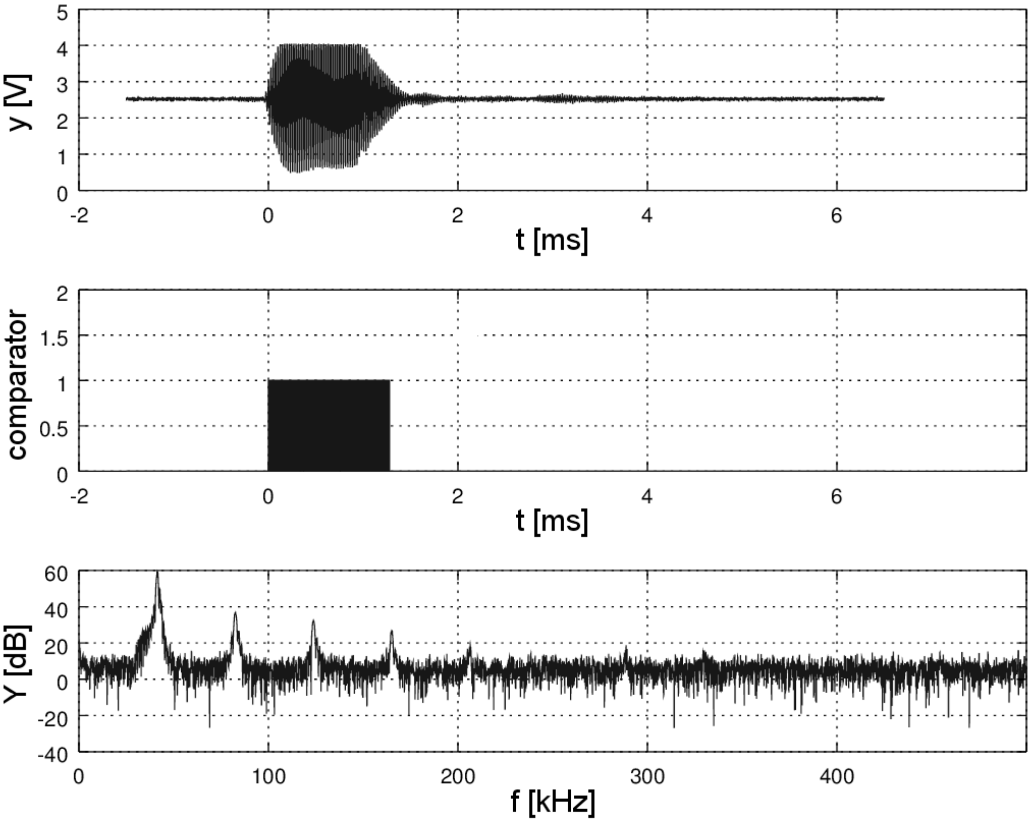


Fig. 7.10: Detection of ultrasonic signal on receiver side

This sensor has not been tested directly with ARP yet, however its performance was verified by the stand-alone experiment described above. Implementation of this sensor to the platform will be part of the future research. Again, the measurements of this sensor can be added to the Kalman filter, or it can be used individually. Good example of its usage is to correct the starting and finishing positions of the scenario in space inside buildings where there is no GPS signal. These positions are each equipped with at least two beacons (slave units) while ARP is equipped with one master unit to enable this measurement. It can be combined with the mouse camera sensor effectively.

7.4 Kalman Filtering

After all the sensors were described, the chapter talks about possibilities of combining values from these sensors in the form of Kalman filtering. There are positions from GPS in latitude and longitude direction ($d_{lat|meas}$ and $d_{long|meas}$) referenced to Base Station; velocity from odometry in platform's movement direction ($v_{x|meas}$); heading ($\Phi_{\psi|meas}$), angular velocity ($\omega_{\psi|meas}$) and acceleration in platform's movement direction ($a_{x|meas}$) from IMU sensor. They might be distorted individually but they are combined by the Kalman filter to get the position of ARP with sufficient precision. The algorithm is a standard discrete Kalman filter plus the mathematical model is updated with new measured values every iteration (e.g. heading to recompute relative velocity to absolute position must be updated from IMU for every new iteration). In the next sections, theoretical background, algorithm and filter outputs are described. Theory is outlined only briefly, however more detailed information can be obtained from [23] or [33].

7.4.1 Theoretical Background

First, theoretical background of Kalman is described. Filter works with a mathematical model and then the algorithm is built on the equations that follows. Symbols are explained in the nomenclature and subscripts section. Matrices and vectors are in bold font; hat operator ($\hat{\mathbf{x}}$) stands for estimate of a variable. Mathematical model was acquired from ARP's geometry with respect to the ARP movement. Movement is available in 2D space (latitude and longitude directions). State vector ($\mathbf{x}$) combines the physical quantities mentioned above.

- Mathematical model matrices and measurement vector:
  - Mathematical model: 7.10 and 7.11
  - State vector: 7.12
  - Input/control vector: 7.13
  - Observation/measurement vector: 7.14
  - Matrices of mathematical model: 7.15, 7.16, and 7.17
- Time update (prediction) for one iteration:
  - Predicted (a priori) state estimate: 7.18
  - Predicted (a priori) estimate covariance: 7.19
- Measurement update (correction) for one iteration:
  - Kalman gain: 7.20
  - Updated (a posteriori) state estimate: 7.21

– Updated (a posteriori) estimate covariance: 7.22

$$\mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \tag{7.10}$$

$$\mathbf{z} = \mathbf{H}_k\mathbf{x}_k + \mathbf{n}_k \tag{7.11}$$

$$\mathbf{x} = \begin{bmatrix} d_{lat|filt} \\ d_{long|filt} \\ v_{x|filt} \\ a_{x|filt} \\ \Phi_{\psi|filt} \\ \omega_{\psi|filt} \end{bmatrix} \tag{7.12}$$

$$\mathbf{u} = \begin{bmatrix} 0 \end{bmatrix};\ \mathbf{u}\, is\ zero\ vector\ because\ there\ is\ no\ input \tag{7.13}$$

$$\mathbf{z} = \begin{bmatrix} d_{lat|meas} \\ d_{long|meas} \\ v_{x|meas} \\ a_{x|meas} \\ \Phi_{\psi|meas} \\ \omega_{\psi|meas} \end{bmatrix} \tag{7.14}$$

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & \Delta t\sin(\Phi) & \frac{1}{2}(\Delta t)^2\sin(\Phi) & 0 & 0 \\ 0 & 1 & \Delta t\cos(\Phi) & \frac{1}{2}(\Delta t)^2\cos(\Phi) & 0 & 0 \\ 0 & 0 & 1 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.15}$$

$$\mathbf{G} = \begin{bmatrix} 0 \end{bmatrix};\ \mathbf{G}\, is\ zero\ matrix\ because\ there\ is\ no\ input \tag{7.16}$$

$$\mathbf{H} = \begin{bmatrix} I \end{bmatrix};\ \mathbf{H}\, is\ identity\ matrix \tag{7.17}$$

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{G}_k\mathbf{u}_k \tag{7.18}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k\mathbf{P}_{k-1|k-1}\mathbf{F}_k^T + \mathbf{Q}_k \tag{7.19}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^T \left(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k\right)^{-1} \tag{7.20}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\left(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}\right) \tag{7.21}$$

$$\mathbf{P}_{k|k} = \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{P}_{k|k-1} \tag{7.22}$$

7.4.2 Algorithm Code

The Kalman filter algorithm is then written in Matlab software to test its functionality. Then it is also written in C language so it can be implemented into the MCU in Control Unit of ARP. This algorithm processes measured quantities in every processing iteration and its result is vector of filtered values ($\mathbf{x}$). Position from this vector can be used for navigation which will be explained in the chapter 8.

In the following example, there is a part of the code which performs one iteration of the Kalman filter. The Matlab version is shown for its better readability. Matrix operations are not so straightforward in C language. Therefore, other libraries must be used there. Whole algorithm can be seen in the attachment D.

Filter uses heading to convert quantities which are relative to the platform into absolute coordinates quantities (latitude and longitude).

```
% Set current observation (measurement) matrix (it is the same all the time)
H(k) = H(1);

% Set current observation (measurement) vector from current measured values
z(k) = [d_lat_meas(k) ;
        d_long_meas(k);
        v_x_meas(k)   ;
        a_x_meas(k)   ;
        Phi_meas(k)   ;
        omega_meas(k) ];

% Set current covariance matrix for observation (measurement) noise
R(k) = [r1, 0 , 0 , 0 , 0 , 0 ;
        0 , r2, 0 , 0 , 0 , 0 ;
        0 , 0 , r3, 0 , 0 , 0 ;
        0 , 0 , 0 , r4, 0 , 0 ;
        0 , 0 , 0 , 0 , r5, 0 ;
        0 , 0 , 0 , 0 , 0 , r6; % r1-6 ... current error of each value from vector z

% Prediction of the Kalman filter - time update
x_apriori(k) = F(k-1) * x(k-1);
P_apriori(k) = F(k-1) * P(k-1) * F(k-1)' + Q(k-1);

% Correction of the Kalman filter - observation (measurement) update
K(k) = P_apriori(k) * H(k)' * inv(H(k) * P_apriori(k) * H(k)' + R(k));
x(k) = x_apriori(k) + K(k) * (z(k) - H(k) * x_apriori(k));
P(k) = (eye(6) - K(k) * H(k)) * P_apriori(k);

% Save current state vector to the filtered values vectors
[d_lat_filt(k); d_long_filt(k); v_x_filt(k); a_x_filt(k); Phi_filt(k); omega_filt(k)] = x(k);

% System model update for the next iteration (uses current filtered heading)
F(k) = [1, 0, sin(Phi_filt(k))*delta_t, 0.5*sin(Phi_filt(k))*delta_t*delta_t, 0, 0      ;
        0, 1, cos(Phi_filt(k))*delta_t, 0.5*cos(Phi_filt(k))*delta_t*delta_t, 0, 0      ;
        0, 0, 1                       , delta_t                             , 0, 0      ;
        0, 0, 0                       , 1                                   , 0, 0      ;
        0, 0, 0                       , 0                                   , 1, delta_t;
        0, 0, 0                       , 0                                   , 0, 1     ];
```

```
% State vector covariance matrix update for next iteration
Q(k) = [q , 0 , 0 , 0 , 0 , 0 ;
        0 , q , 0 , 0 , 0 , 0 ;
        0 , 0 , q , 0 , 0 , 0 ;
        0 , 0 , 0 , q , 0 , 0 ;
        0 , 0 , 0 , 0 , q , 0 ;
        0 , 0 , 0 , 0 , 0 , q ]; % q ... current error of the system
```

Last but not least, after the algorithm functionality is verified, error covariance matrices have to be tuned according to the current state of the system. The algorithm itself does not give acceptable results. To achieve proper functionality, the measurement covariance matrix **R** has to be set according to current errors of sensors that give the corresponding quantity. For example, error of GPS unit is returned in its output and this value can be used as the error value for position (latitude and longitude) measurement. Similar approach works for odometry and IMU measurement. Then the state vector covariance matrix **Q** has to be set as well according to current error in the system. The smaller the values, the more trust is in the prediction. More information about Kalman filter tuning can be found in [31] and [34].

7.4.3 Measurement Results

There is a real measurement shown in this section to evaluate the filter performance. ARP was performing elliptical movement during this test. It can be seen in the figure 7.15. There is a dependence of the latitude position on the longitude position shown there; it is the best visualization of the movement. Both positions are referenced to Base Station. The movement was controlled manually by the RC transmitter, therefore, the requested values that are shown in the graphs are forward velocity ($v_{x|req}$) and angular velocity ($\omega_{\psi|req}$).

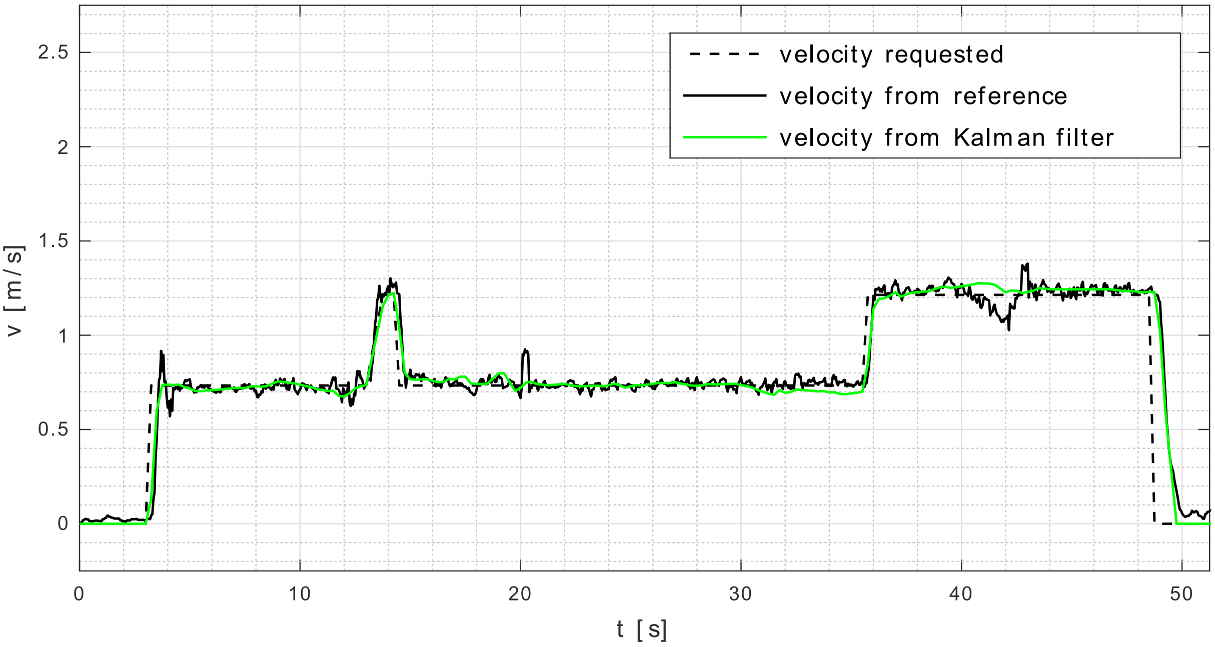


Fig. 7.11: Graph of Kalman filter velocity result

The filter performance can be then seen in the figures 7.11, 7.12, 7.13 and 7.14 – velocity, position (in latitude and longitude direction) and heading dependencies on time. Angular velocity graph is not so important and it is shown in the figure E.3 in the attachments.

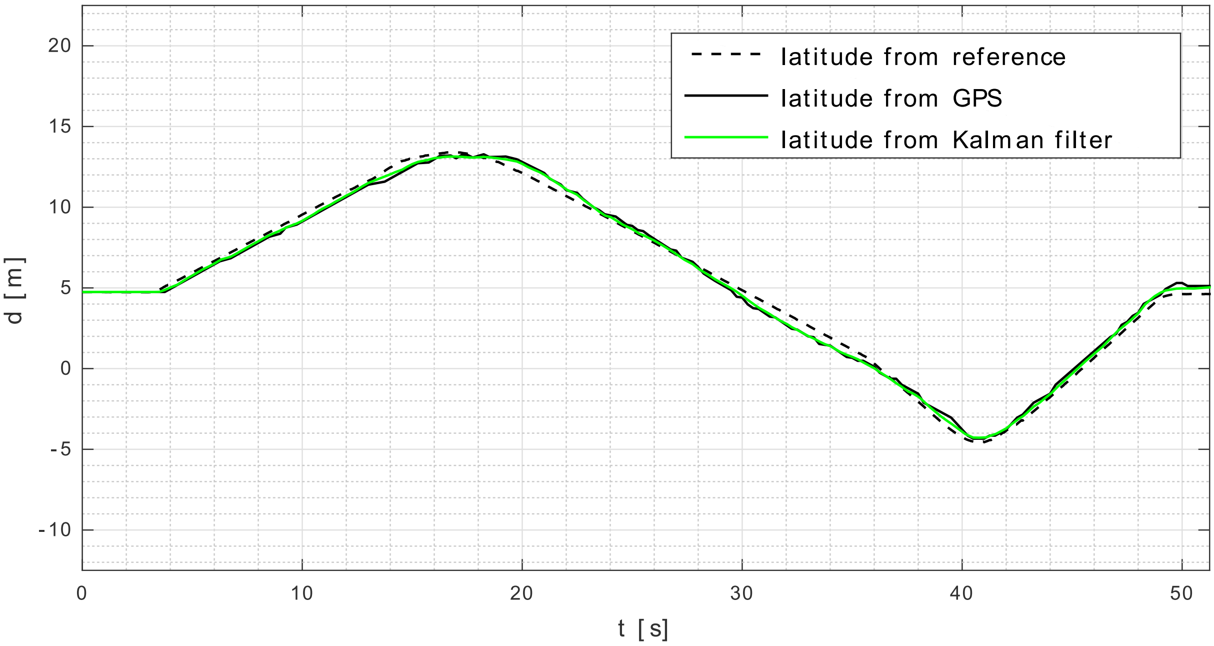


Fig. 7.12: Graph of Kalman filter latitude position result

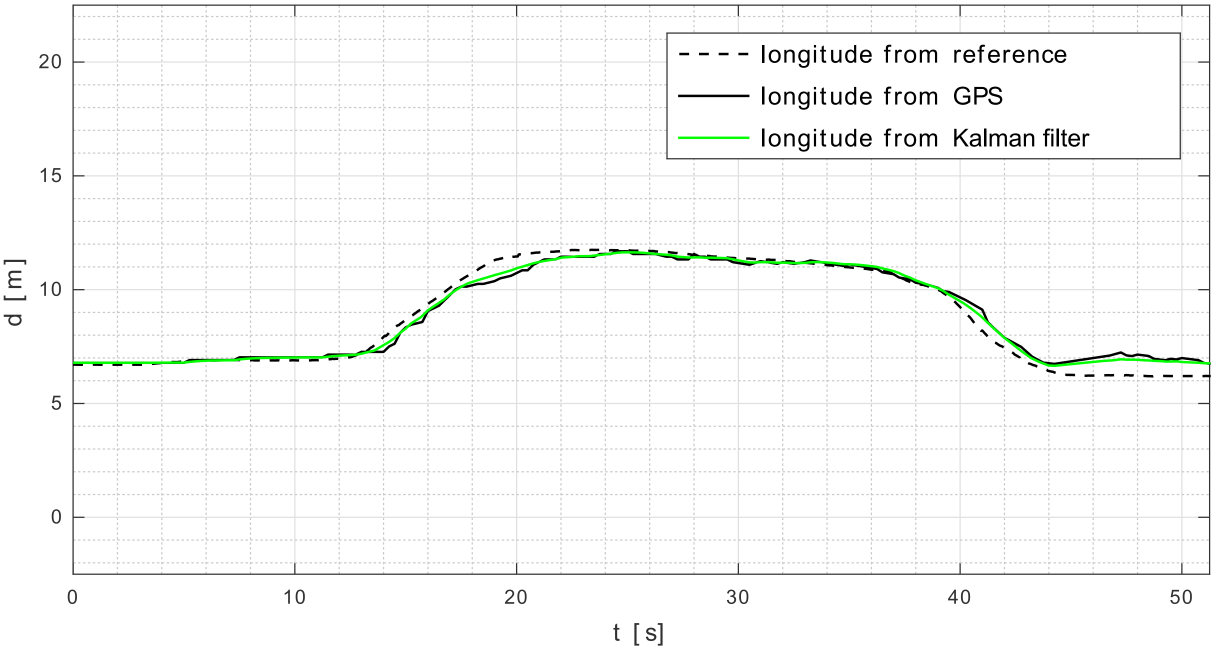


Fig. 7.13: Graph of Kalman filter longitude position result

From measurement of different scenarios and comparison of the results with reference position (given by differential GPS), it has been concluded that the platform can be used

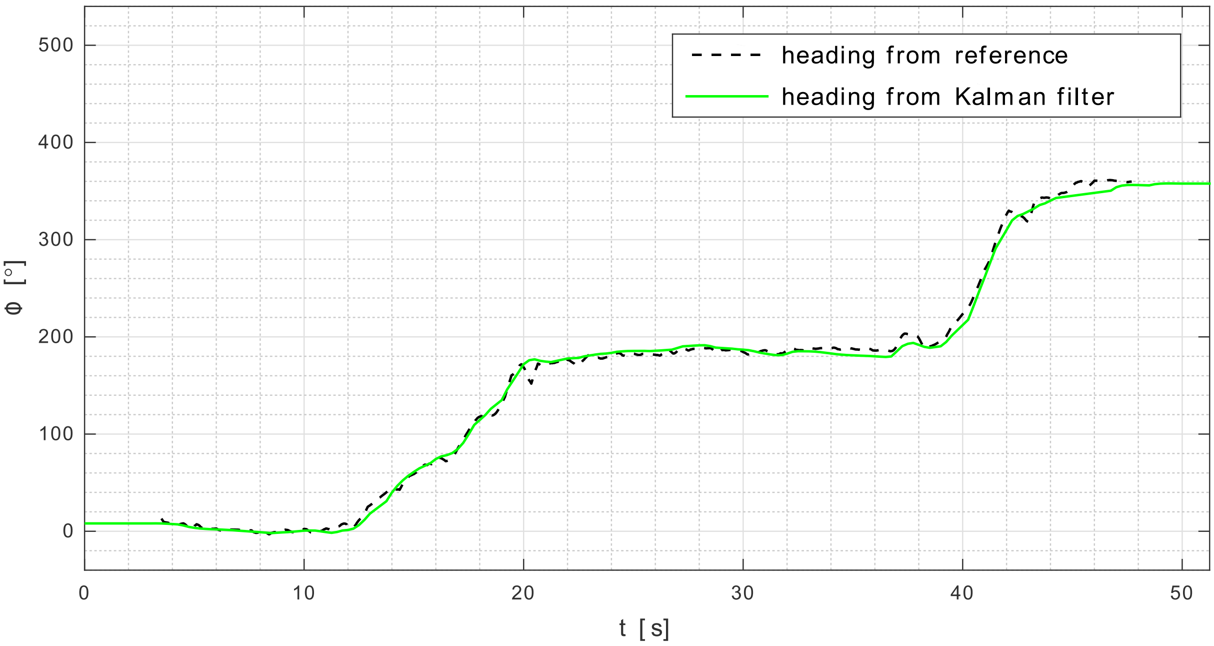


Fig. 7.14: Graph of Kalman filter heading result

for ADAS tests even with standard sensors combined with Kalman filtering instead of implementing industrial differential GPS systems. The results of measurement are shown in the table 7.2. The measurement consists of one hundred samples of elliptical movement performed on the test track.

Tab. 7.2

Sensors used for localization	Error in position measurement (*)	Total distance error
Odometry	2.64 m	6.58 %
GPS	0.86 m	2.12 %
Kalman filter	0.61 m	1.51 %

Tab. 7.2: Localization measurement results

$$\sigma[\Delta(d)] = \sqrt{\frac{1}{K}\sum_{k=1}^{K}[\Delta(d_k)]^2;\ where\ K\ is\ number\ of\ samples} \tag{7.23}$$

$$\Delta(d) = \sqrt{(d_{lat|ref} - d_{lat|meas})^2 + (d_{long|ref} - d_{long|meas})^2} \tag{7.24}$$

*) The error is computed as standard deviation (σ) of each sample in every measurement (equation 7.23). Deviation is a value of the difference between position given by the tested sensor and position given by the reference (equation 7.24).

Measurement trajectory is shown in the graph 7.15 (position from Kalman filter) and for comparison in the graph 7.16 (position only from odometry). Arrows indicate the current heading of ARP. It can be seen that odometry is gaining increasing error over time; while Kalman filter result is kept closer to the reference during the whole scenario – it is combined from absolute position (GPS) which keeps the long-term position in range and from relative position (odometry plus IMU which corrects short-term errors). Average total distance that was travelled during this test is about 41 m and maximum velocity is 5 km/h.

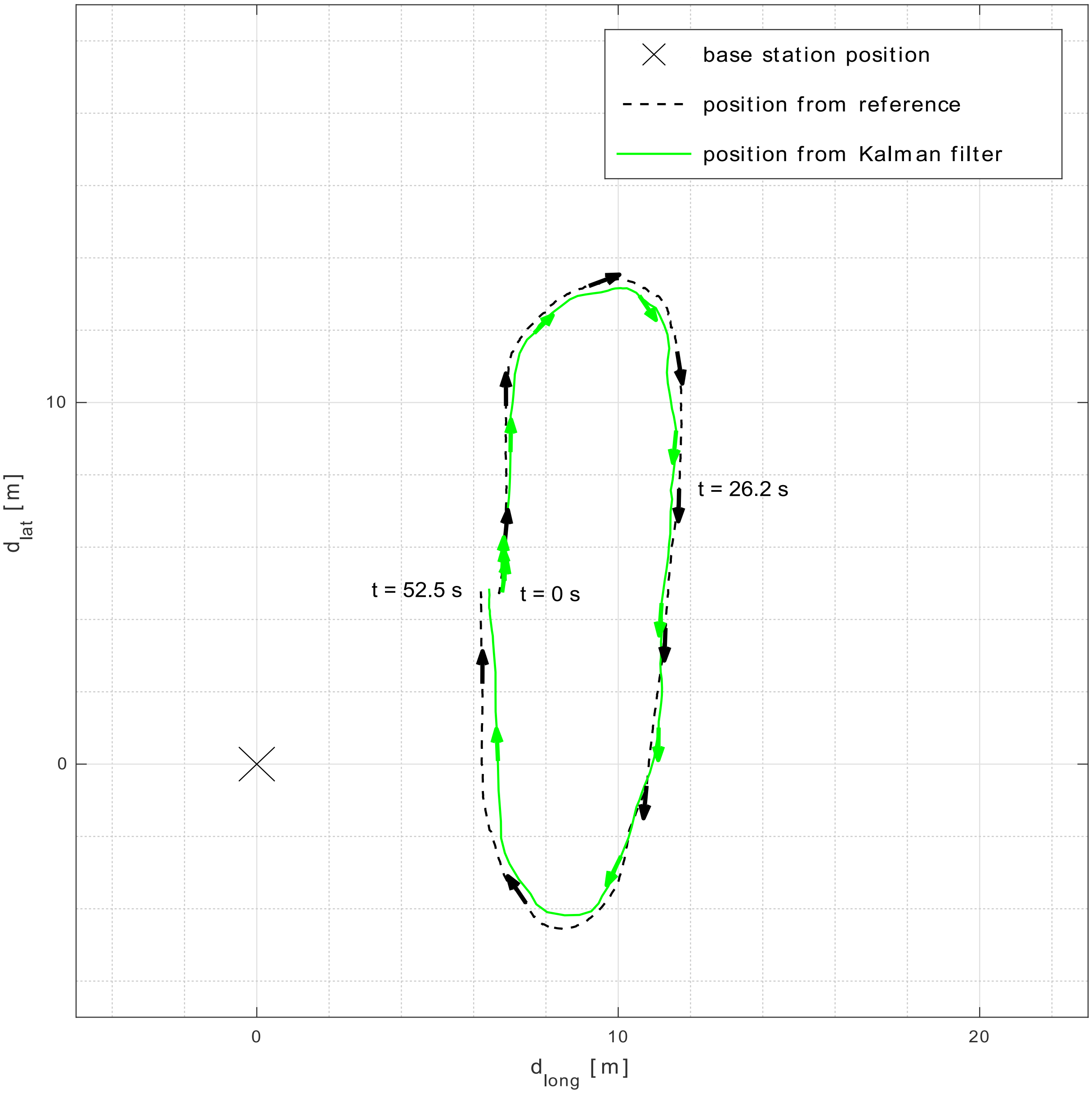


Fig. 7.15: Graph of Kalman filter position result

The results of mouse camera measurement were mentioned in the section 7.2 and they are not included in this table. They were collected during different type of scenario testing because special surface is needed for this sensor to work. For comparison, the error in travelled distance for 4 m long scenarios was kept in range of 5 %. The ultrasonic distance measurement was not yet tested for the ARP localization so the results are not known. It was tested only in the stand-alone experiment described in the section 7.3 and the error was 1.8 % for distances up to 5 m. To include these sensors to Kalman filter and to perform measurements that compare all the results can be part of future experiments.

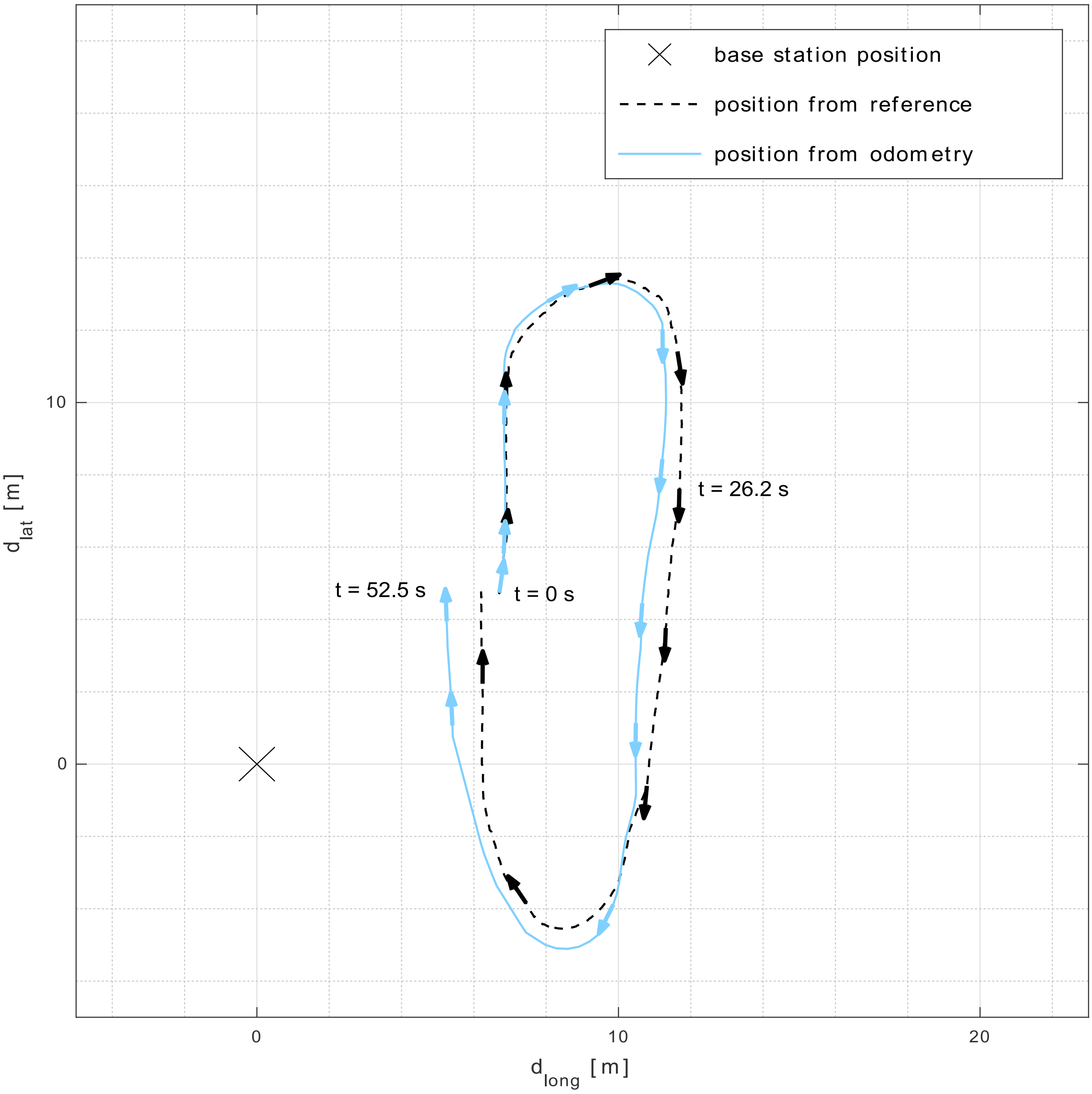


Fig. 7.16: Graph of odometry position result

Additional measurements were performed for speeds up to 50 km/h with sensors from table 7.2. In conclusion, the deviation of position of scenarios on a test track was kept in range of 1 m for all these tests using the Kalman filter localization method. Some ADAS tests require better precision (in range of centimetres), however it is sufficient for majority of tests or internal company research. Internal tests are test cases, when a new car's equipment is being developed and tests with high precision can be performed later with different equipment or platform. Nonetheless, the goal of the research is to continue with these measurements and to try different sensors and filter tuning to achieve even better precision results.

Implementation of all these different sensors can improve autonomy of ADAS testing in various testing areas without high increases in the design's costs. Also with the option of omnidirectional movement, one ARP can be used for more different scenarios which makes it more variable as well.

7.5 Future development

Last but not least, ideas for future research of localization methods are outlined here.

- Comparison of different GPS modules (e.g. u-blox GPS system [36]).
- Possibility of position computation from camera image shifting ([9]).
- Solutions for local positioning systems ([20]). Problems with range of covered area and accuracy would have to be solved.
- Implementing the short range local positioning using ultrasonic sensors (section 7.3).
- Implementing sensors measuring displacement against the ground into the Kalman filter (section 7.2).
- Tuning of Kalman filter covariance matrices **Q** and **R** based on different methods ([31] and [34]).
- Performing and evaluating more measurements to compare results for different sensors.

8

Navigation and Synchronization of ARPs

This chapter talks about autonomous control (i.e. navigation) of ARP. For now, PID (Proportional Integral Derivative) controllers for the scenario trajectory control are used. First PID regulates travelled distance (to get navigate the platform in specified time to specified distance) and the second one compensates deviation from the straight trajectory. There are PID regulators for velocity and steering angle of each wheel as well. In the future, the goal is to implement solutions based on the feed-forward state feedback controllers with the state estimation (see [3]). Content of this chapter was also used for preparation of the papers [Pub. 9] and [Pub. 10] which can be found among publications of this doctoral thesis' author.

8.1 PID Control

When position is obtained with sufficient precision which was explained in the chapter 7, the next part of research can lead to speed and position control of the platform. As mentioned in the chapter's introduction, ARP uses PID (Proportional Integral Derivative) controllers. There are four PID controllers for RPM control of each wheel (embedded directly inside the engines' drivers). These are sufficient and they are tuned finely for the used engines and ARP's weight. Different tuning is also pre-set for dummies that can be carried by the platform because dynamics of the platform changes with different load. The step response of these regulators can be seen in the figure 8.1. Requested RPM for each wheel is an input for each controller. Error is then computed as deviation between requested and real RPM. Output is a current driven to the engine.

Then there are PID controllers for each wheel's steering and those will be kept in the design as well. Their goal is to keep the wheels in the requested steering angle. When turning the platform, one must remember to recompute steering angles for inner and outer wheels separately according to ARP's geometry (see chapter 6.4). These controllers must react

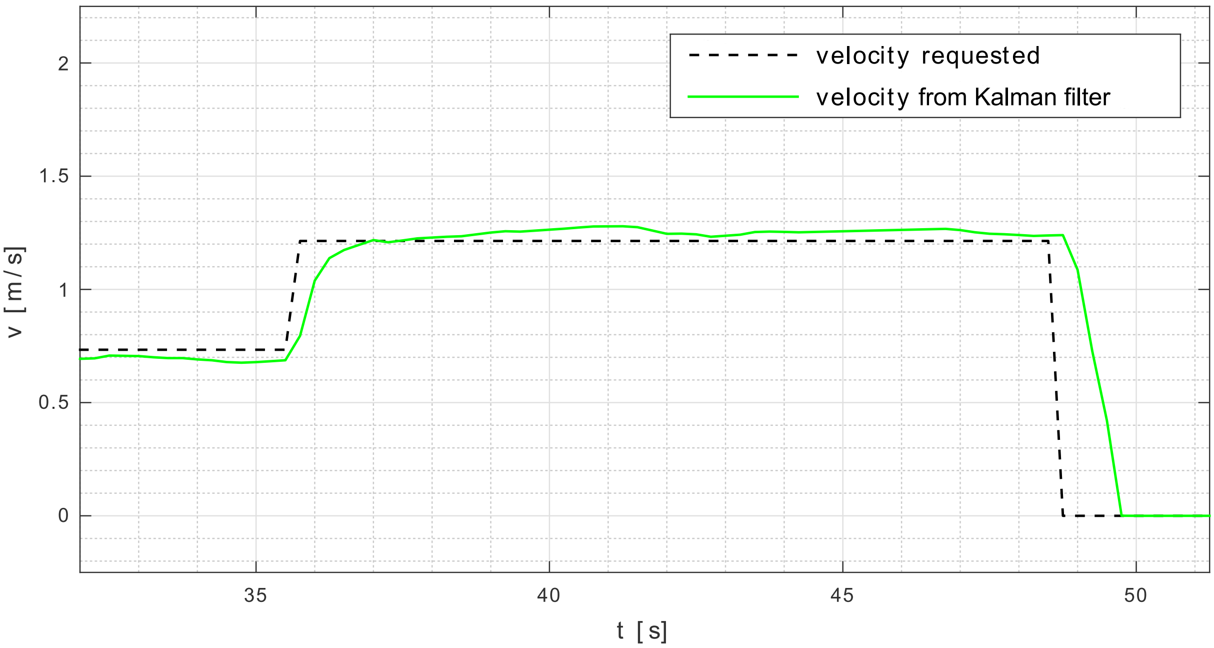


Fig. 8.1: Step response of engine drivers PID controllers

very precisely, otherwise the platform does not turn properly and it can be even damaged. The reason is that the omnidirectional movement is based on four steerable wheels. When wheels are set to different angles the movement cannot be predicted and wheels are damaged due to friction against the surface.

Every PID controller is based on the equations 8.1 and 8.2 where $u(t)$ is control signal, $e(t)$ is error between requested signal $r(t)$ and measured signal $y(t)$. Constants K_p, K_i and K_d are proportional, integral, and derivative terms respectively. They set the time response of the controller and they must be tuned to achieve satisfactory results. In the C language code, every PID is implemented in the discrete time form (see the following code example where one iteration of PID output computation is shown). Its block diagram is shown in the figure 8.2.

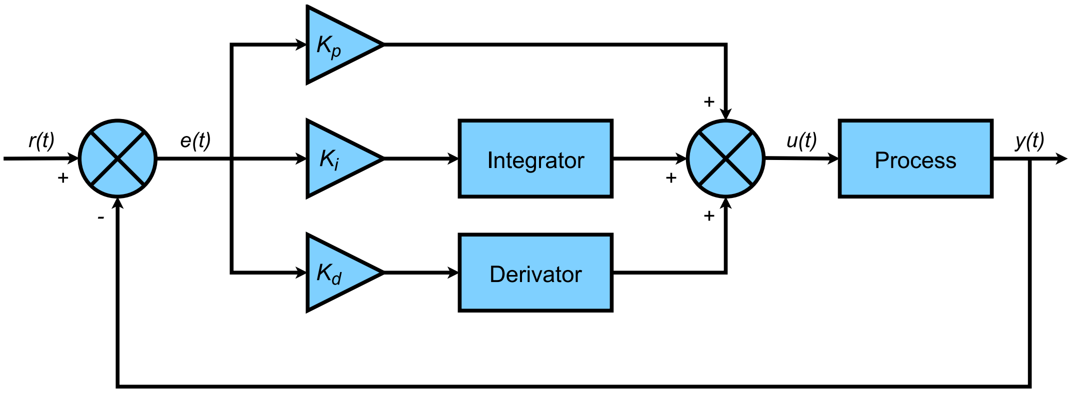


Fig. 8.2: Step response of engine drivers PID controllers

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d(\tau) + K_d \frac{de(t)}{dt} \tag{8.1}$$

$$e(t) = r(t) - y(t) \tag{8.2}$$

```
/* Compute error */
double error = set_point - input;

/* Instead of remembering error sum - remember it with K_i together (inside integral) */
/* - this prevents bumps when changing constants on the run */
ki_err_sum += K_i * error;

/* Compute derivative */
double d_input = input - last_input;

/* Compute PID output */
/* - use the derivative on input instead of error to avoid derivative spikes */
output = K_p * error + K_i_err_sum + K_d * d_input;

/* Prevent error sum and output to exceed the limits */
/* - if PID can exceed limits it sends value to output that is not possible which makes */
/*   random lags when returning back into the limits */
if (output > out_max)
{
        K_i_err_sum -= (output - out_max);
        output = out_max;
}
else if (output < out_min)
{
        K_i_err_sum += (out_min - output);
        output = out_min;
}

/* Remember last input for the next iteration */
last_input = input;
```

The most difficult part of the controller design is its tuning. In case of RPM and steering PID controllers, a heuristic method based on step response measurement is used:

1. Open loop step response is measured to get the system's response delay time.
2. Start with proportional gain set according to response delay time and no integral or derivative.
3. Double proportional gain until it starts oscillating and then halve it.
4. Implement small integral gain.
5. Double integral gain until it starts oscillating and then halve it.
6. Add/increase derivative gain to reduce oscillations.
7. Repeat from step 3. If oscillations cannot be reduced by increasing derivative anymore, last stable setup is the result.

Above this set of RPM and steering controllers, there must be a controller that sets overall velocity and angular velocity of the platform according to its real versus required position (given by the scenario's trajectory description). For now, two PID controllers are used. One controls the position in the direction of movement, the other one controls the lateral deviation from the scenario's trajectory. The performance of the first one can be seen in the figure 8.3. Result of this controller is the requested velocity that is sent to RPM controllers. As said before, it must be recomputed for every wheel according to the platform's geometry. Its course can be seen in the figure 8.4. ARP performed straight scenario with return to the starting position with length of 4 m in this measurement.

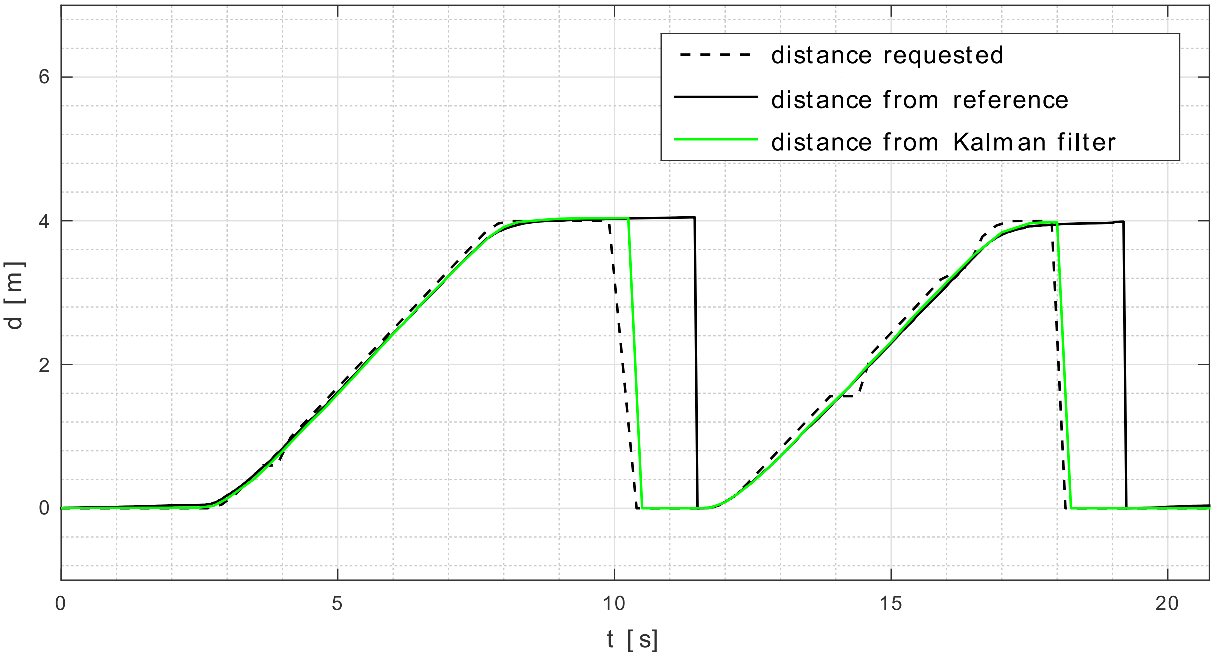


Fig. 8.3: Graph of distance PID control

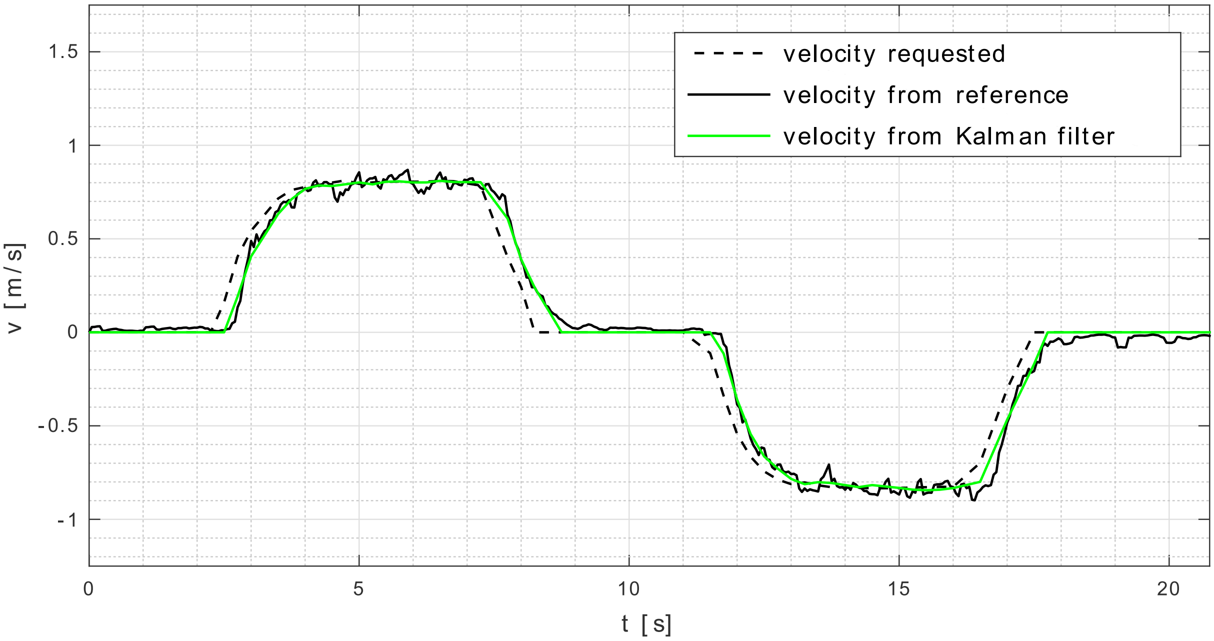


Fig. 8.4: Graph of velocity PID control

The second controller (the one for lateral deviation) behaves very similarly. Its result is the requested angular velocity of the platform which is recomputed to steering angles for each wheel and applied as input for steering PID controller. Both of these controllers are set as PD (no integral gain) because when error is zero there is no need for movement. Their tuning was performed with the same process as RPM and steering controllers; integral gain related steps were skipped. Measurement of one hundred samples with this type of scenario was performed. The results show that the measured position of ARP was kept in range of 4 % error from the requested position.

Another interesting graph in the attachments is voltage and current courses during the scenario (figure E.2). It shows battery behaviour during the movement. It can be seen that voltage drops significantly when platform accelerates and then slowly returns back as current drops down. Current course shows that its peak can get up to 30 A when accelerating to 5 km/h. On the other hand, when the platform is decelerating there is a negative peak which indicates recuperation of power back into the battery pack.

Parameters of this scenario can be set from PC Application from the scenario tab (see figure C.2 in the attachments). User can choose distance, velocity and time. Calculation of feasibility of selected parameters is done directly in the application, it is displayed on the graph and if they are correct they are sent to the platform. Scenario can then be started by the trigger button on Kill Switch Button unit or by the photocell connected to Base Station. Other possibilities of synchronization with VUT (Vehicle under Test) are described in the section 8.3. For now, the application supports straight trajectory scenarios, however, future implementation will include scenarios with custom trajectories as well.

8.2 State Feedback Controller

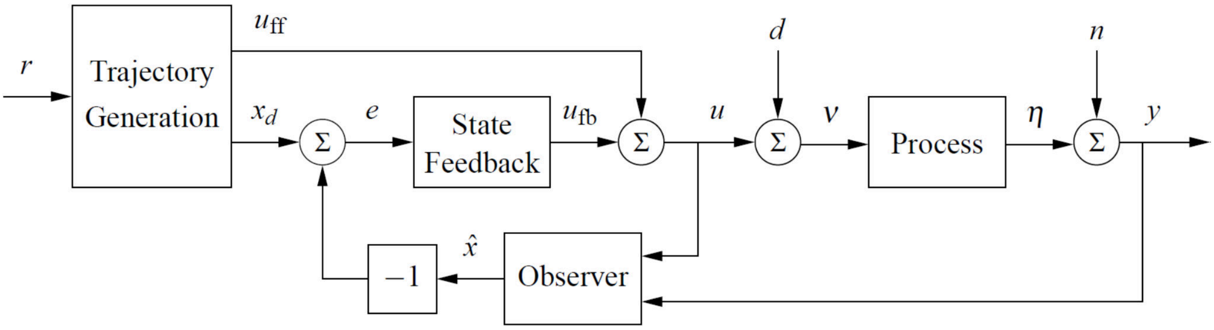


Fig. 8.5: Block diagram of the feed-forward state feedback controller |Borrowed from [3]|

There is a lot of space for future improvements and a part of the research will be focused on implementation of more sophisticated controllers, such as a feed-forward state feedback controller. Block diagram of such regulator can be seen in the figure 8.5; where r is reference signal, x_d is actual signal to track, u_{ff} is feed-forward control signal, e is error of input signal, u_{fb} is feedback control signal, u is combined control signal, v is control signal u disturbed with noise d, η is output signal, y is output signal disturbed with noise

n, $\hat{x}$ is observer's state vector estimate. A good publication to study this field can be found under the item [3] in the references.

The following research will also deal with comparison of different regulators and their accuracy. The PC Application will be improved by the section of scenario management and monitoring. There is also a new idea to be explored which concerns random scenarios – they can be useful for testing the cars with higher level of autonomous driving.

8.3 Synchronization Possibilities of ARP

Second part that is discussed in this chapter focuses on the idea of platforms cooperation and synchronization. Later when the autonomous driving gets to higher levels, one platform does not need to be sufficient for ADAS testing. Therefore, from the beginning, the platform is designed to have an ability of synchronization with more platforms or vehicles.

8.3.1 ARP and VUT

The easiest car-to-platform synchronization is performed by the photocell scenario trigger which is already implemented (see section 4.2). Once the VUT goes through the photocell, the platform performs a scenario of the defined parameters. The meeting point is pre-set by the scenario parameters and is located at constant point. The VUT must optimize its trajectory according to this meeting point.

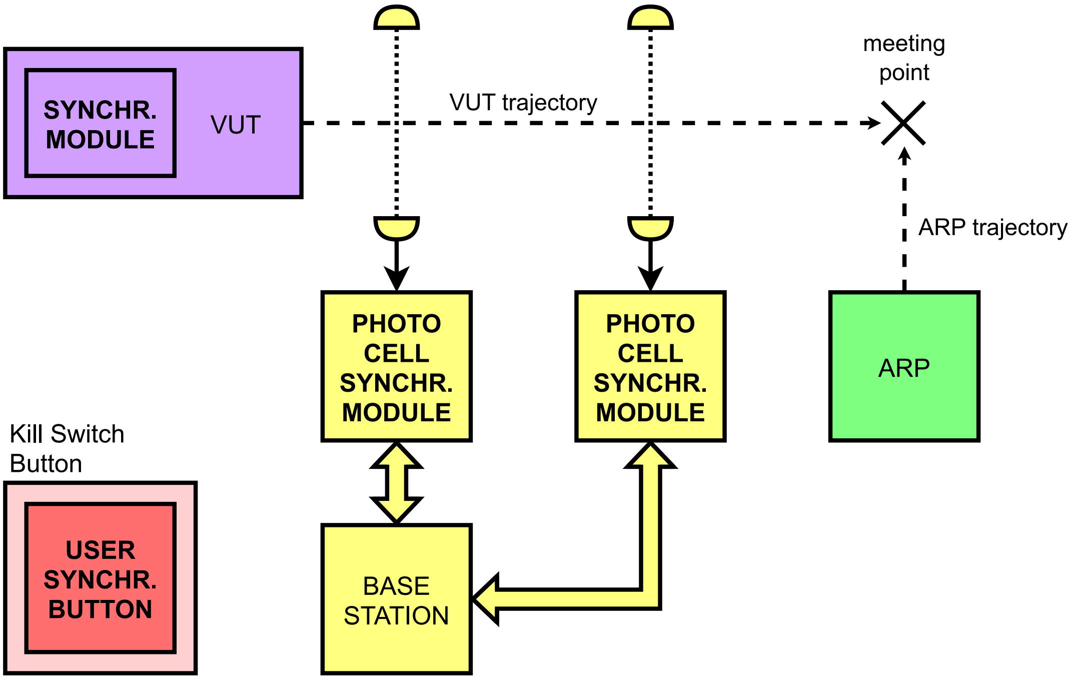


Fig. 8.6: Synchronization of ARP and VUT

There is an option to improve this procedure which has to implement two photocells.

This time, the velocity of VUT can be measured and sent online to ARP, which can give ARP the information to optimize its scenario trajectory parameters according to the VUT speed. ARP can then keep the meeting point not at constant point in absolute time but at constant point according to VUT's speed.

The next option that is currently in development is to use a so called virtual photocell. It adds another module to the ARP system. This module can have its own position measurement or it could use position of the VUT if it is known. Then it can communicate through wireless network with ARP which can be synchronized directly to the car's position. The scenario is then performed independently on the car's position, because it can compensate its position and time deviations.

Both of these options (photocells and synchronization module) are represented by the figure 8.6. There is also a third option there, which is to use the trigger button on Kill Switch Button module. This option can be used for manual start of the scenario and it is good to test the ARP movement before a real ADAS test is performed.

8.3.2 Multiple ARPs

As said before, one ARP does not need to be sufficient for more complex ADAS tests that are likely to be required in the future by Euro NCAP standards ([15]). Synchronization of more ARPs is enabled by its design from the beginning of the research. The idea is to use Base Station as a cloud connecting the network of possibly more platforms and tested car (VUT) together (see chapter 5). This cloud is based on the DigiMesh protocol designed by the XBee manufacturer ([11]) that makes the communication robust and safe. Base station works as a coordinator to connect the devices together as well. More information about this network can be found in the chapter 4. This idea opens new research possibilities of the robots' cooperation in the field of ADAS testing.

8.3.3 Multiple ARPs with Autonomous VUT

Another part of the autonomous ADAS testing research aims to replace even VUT by a platform which is going to carry the same sensors and which is going to have the same geometry. ARP is going to be synchronized with this new autonomous VUT, which makes the testing fully autonomous without the need of test drivers. This project is part of another study in the company Valeo Autoklimatizace k.s. in Prague. Furthermore, as it was mentioned in the previous section, future ADAS tests will more likely require usage of more testing targets, so this idea leads to synchronization of more platforms that carry the targets with the automatic VUT together. The synchronization with the tested car is possible by communication with its localization unit usually based on the differential GPS. This is the main area for future research which is going to enable fully autonomous ADAS tests implementing multiple Euro NCAP targets.

9

Conclusion

This doctoral thesis is focused on the research of autonomous testing possibilities of Advanced Driver Assistance Systems (ADAS) in automotive. Design of the Autonomous Robotic Platform (ARP) is covered with implementation of several innovative ideas and proposals of possible future research. The main goal is to improve the autonomy of ADAS testing which is achieved by innovative features described later throughout the paper. Such a platform must be capable of mounting the normalized targets and perform the Euro NCAP (European New Car Assessment Programme, [17]) scenarios as it was outlined in the introduction.

There already are several platforms in development which are described in the chapter 2 that deals with the background research of this field. The proposed prototype takes inspiration in them. Then the thesis explains improvements that can be used for this concept and it introduces new ideas to the field that are related to the platform's modularity, reliability, safety, localization and navigation. It also talks about its movement possibilities (e.g. omnidirectional movement) and synchronization between the platform and a car or even among more platforms together if necessary. The concept of the prototype's design is introduced in the chapter 3 as well because this prototype provides a tool to validate new and innovative ideas that are discussed theoretically throughout the thesis. This chapter also deals with the introduction to the ADAS systems and their testing in general to give the overview of the field. Euro NCAP standards that regulate the ADAS testing are explained as well.

In the two following chapters 4 and 5, the paper focuses on the ARP prototype's design. Since this thesis is discussing the subject from the embedded systems point of view, there are chapters about the prototype's hardware and software solution. Several innovative features are already described by these chapters, i.e. the ARP's modularity, reliability and safety. Block diagrams and flowcharts are used to give illustrative examples of the design. The mechanical construction is part of another research paper which can be found in the references section under the item [26]. The research led to a successful completion of the prototype which was possible due to the cooperation with the company Valeo Autoklimatizace k.s. in Prague and its project Opportunities for Students – Valeo

R&D Program. The prototype is a suitable tool to show how the implemented innovative features work in the real scenarios and how they were used to improve the field of ADAS testing. It is already used for the real ADAS testing in several Valeo sites worldwide (e.g. in Germany, Ireland or South Korea) as well. Its top speed can get up to 50 km/h and during the future research it is going to be extended to 80 km/h to increase coverage of ADAS tests. Prototype is described in these two chapters and then it is used to get real measurement data that do support the innovative features description in the chapters 6, 7 and 8.

These chapters (6, 7 and 8) talk about the key innovative features that were used during the research. They are the core of this thesis and together they improve autonomy of ADAS tests by using this platform. They show how these testing platforms can be improved in the ways of movement, localization and navigation. The first chapter summarizes the features concerning modularity and safety. In general, ARP is designed to be as modular and safe as possible, many automatic and manual safety mechanisms are proposed. The prototype implements mechanisms to check and fix possible errors automatically; furthermore, when the error that cannot be fixed occurs, ARP's operation is stopped to avoid more dangerous faulty behaviour. The last option if everything else fails is to use the manual kill switch that is provided as well. Then the cloud based communication is proposed, which leads to an increase in the testing area coverage together with an increase in the communication's reliability (lower data loss) with maintaining the same wireless modules. The final feature described in the chapter 6 focuses on the idea of omnidirectional movement which improves ARP's variability. Its advantages and design difficulties arising from this option are explained there.

The next chapter (7) focuses on the possibilities of ARP's localization. Here is a great space to improve the field and it is also one of the main goals of this paper. Different sensors that aim to improve the localization precision without high costs are listed and their performance is shown in the attached graphs. There are standard sensors such as odometry, IMU and GPS. On the other hand, the examples of newly used sensors are mouse camera sensor or point-to-point ultrasonic distance measurement. Different sensors can be combined by using the Kalman filtering which increases overall precision of localization. The filter's theoretical background together with its implementation is described and results of different methods are compared by performing measurements on the ARP prototype.

Finally, there are the navigation and synchronization possibilities of ARP explained in the chapter 8. The paper focuses on the controllers that are used now (PID) and that are going to be tested in the future (state feedback controller). There is theoretical background and their implementation described here. The synchronization with VUT (Vehicle under Test) can be performed by a photocell connected to Base Station module or by stand-alone module embedded to VUT. Last but not least, the possibility to make the testing fully autonomous is described. The idea is to use autonomous VUT (platform with the same sensors and geometry as real VUT) that is synchronized with ARP carrying the target. The scenarios can be repeated over and over and results logged automatically without the need of test drivers. Finally, even more than one ARP and VUT could be synchronized together to perform more complicated ADAS tests. This idea is going to be part of the future research and it can build on the results achieved in this doctoral thesis.

In conclusion, there are many innovative features that this doctoral thesis introduces into the field of autonomous ADAS testing which highlights its contribution to the scientific research. They aim to improve autonomy and variability of ADAS testing using one ARP device. On top of that its research led to the successful completion of ARP prototype that is already used for ADAS testing worldwide. It was enabled due to the cooperation with the company Valeo Autoklimatizace k.s. in Prague and with other team members working on the project there; as well as in the cooperation with the Faculty of Electrical Engineering at the University of West Bohemia by means of the scientific projects SGS (Student Grant Competition) mentioned in the acknowledgment. Author of this doctoral thesis provided embedded design of the prototype and all the ideas presented in this thesis are solely work of its author. The content of this thesis was also used to prepare several scientific papers with other colleagues from the company Valeo Autoklimatizace k.s. or from the University of West Bohemia whose main author is the author of this thesis and they can be found in the list of author's publications. Three of them were presented at international scientific conferences ([Pub. 1], [Pub. 3] and [Pub. 8] – this one also won the Best Paper Award certificate), one is in review for another international conference ([Pub. 11]) and one of them ([Pub. 9]) is in review for a scientific journal with academic impact factor (IF). The content described in this thesis can also be used as a background for future research which is its other accomplished quality.

References

[1] *4activeSystems* [online]. Available at: http://www.4activesystems.at/en

[2] Adafruit. *Adafruit Ultimate GPS Breakout – 66 channel w/10 Hz updates* [online]. 2021. Available at: https://www.adafruit.com/product/746

[3] K. J. Astrom and R. M. Murray. *Feedback Systems. An Introduction for Scientists and Engineers.* Princeton University Press, Princeton, 2012. Copyright © 2008 by Princeton University Press. ISBN-13: 978-0-691-13576-2

[4] Avago Technologies. *ADNS-3080 High-Performance Optical Mouse Sensor* [online]. Copyright © Avago Technologies, 2005-2009. Available at: https://datasheet.octopart.com/ADNS-3080-Avago-datasheet-10310392.pdf

[5] *ABDynamics* [online]. Copyright © Anthony Best Dynamics Limited, 2018. Available at: https://www.abdynamics.com/en

[6] Benjamin's robotics. *VESC – Open Source ESC* [online]. 2015. Available at: http://vedder.se/2015/01/vesc-open-source-esc/

[7] J. M. Blosseville. *Driver assistance systems, a long way to AHS.* 2006 IEEE Intelligent Vehicles Symposium, Tokyo, 2006, pp. 237-237. doi: 10.1109/IVS.2006.1689634

[8] Bosch Sensortec. *BNO055 Intelligent 9-axis absolute orientation sensor* [online]. Copyright © Bosch Sensortec, 2020. Available at: https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bno055-ds000.pdf

[9] L. Cheng and Y. Wang. *Localization of the autonomous mobile robot based on sensor fusion.* 2003 IEEE International Symposium on Intelligent Control, Houston, 2003, pp. 822-826. doi: 10.1109/ISIC.2003.1254742

[10] *Dewetron* [online]. Copyright © DEWETRON GmbH, 2018. Available at: https://www.dewetron.com/solutions/automotive/advanced-driver-assistance-systems

[11] DIGI. *DIGI XBEE S2C 802.15.4 RF MODULES* [online]. Copyright © Digi International Inc., 1996-2018. Available at: https://www.digi.com/pdf/ds_xbee-s2c-802-15-4.pdf

[12] *DSD* [online]. Copyright © DSD, Dr. Steffan Datentechnik Ges.m.b.H., 2014. Available at: http://www.dsd.at/index.php?lang=en

[13] *dSPACE* [online]. Copyright © dSPACE GmbH, 2018. Available at: https://www.dspace.com/en/inc/home/applicationfields/our_solutions_for/driver_assistance_systems.cfm

[14] e-Power Hobby. *HUB brushless motor for longboard 2 X 83mm* [online]. Copyright © E-Power Hobby, 2018. Available at: https://epowerhobby.com/product/e-power-hobby-hub-brushless-motor-for-longboard-2-x-83mm-truck-used

[15] Euro NCAP. *Euro NCAP 2025 Roadmap* [online]. Copyright © Euro NCAP, 2017. Available at: https://cdn.euroncap.com/media/30701/euroncap-roadmap-2025-v4-print.pdf

[16] Euro NCAP. *Test Protocol – AEB systems* [online]. Copyright © Euro NCAP, 2015. Available at: https://cdn.euroncap.com/media/17719/euro-ncap-aeb-test-protocol-v11.pdf

[17] Euro NCAP. *The Official Site of the European New Car Assessment Programme* [online]. Copyright © Euro NCAP, 2018. Available at: https://www.euroncap.com/en

[18] N. Gageik, M. Strohmeier and S. Montenegro. *An autonomous UAV with an optical flow sensor for positioning and navigation.* Int. J. of Advanced Robotic Systems, 2013, volume 10, issue 10. doi: 10.5772/56813

[19] P. Giusto, S. Ramesh and M. Sudhakaran. *Modeling and analysis of automotive systems: Current approaches and future trends.* 2016 4th International Conference on Model-Driven Engineering and Software Development (MODELSWARD), Rome, 2016, pp. 704-710

[20] D. Gualda, J. Urena, J. C. García, E. García, D. Ruiz and A. Lindo. *Fusion of data from ultrasonic LPS and isolated beacons for improving MR navigation.* 2014 International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings, Montevideo, 2014, pp. 1552-1555. doi: 10.1109/I2MTC.2014.6861006

[21] IEEE STANDARDS ASSOCIATION. *IEEE 802.15.4-2015 - IEEE Standard for Low-Rate Wireless Networks* [online]. Copyright © IEEE, 2018. Available at: https://standards.ieee.org/standard/802_15_4-2015.html

[22] C. Laugier. *Some key technologies for the next car generation.* 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV), Singapore, 2014, pp. 1-2. doi: 10.1109/ICARCV.2014.7064482

[23] D. Lee, S. Son, K. Yang, J. Park and H. Lee. *Sensor fusion localization system for outdoor mobile robot.* 2009 ICCAS-SICE, Fukuoka, 2009, pp. 1384-1387

[24] Nanotec. *DB59C024035-A – Brushless DC Motor* [online]. Copyright © Nanotec Electronic GmbH & Co KG, 1995-2018. Available at: https://en.nanotec.com/products/1833-db59c024035-a

[25] *National Marine Electronics Association* [online]. Copyright © NMEA, 2021. Available at: https://www.nmea.org

[26] J. Német. *Dynamická platforma pro nesení lehkého cíle pro testování systémů ADAS.* Prague, 2018. Master thesis (in Czech). Czech Technical University in Prague. Faculty of Mechanical Engineering

[27] *OxTS* [online]. Copyright © OxTS Ltd. Available at: https://www.oxts.com/what-is-adas-testing

[28] K. Park, J. Kwahk, S. H. Han, M. Song, D. G. Choi, H. Jang, D. Kim, Y. D. Won and I. S. Jeong. *Modelling the Intrusive Feelings of Advanced Driver Assistance Systems Based on Vehicle Activity Log Data: Case Study for the Lane Keeping Assistance System.* Int. J. Automotive Technology 20(3), 2019, pp. 455-463

[29] J. Pinker. *Mikroprocesory a mikropočítače.* BEN – Technická literatura, Prague, 2004. 160p. ISBN: 80-7300-110-1

[30] SiRF Technology, Inc. *NMEA Reference Manual* [online]. Copyright © SiRF Technology, Inc., 2005. Available at: https://www.sparkfun.com/datasheets/GPS/NMEA%20Reference%20Manual1.pdf

[31] S. Soltani, M. Kordestani and P. Karimaghaee. *New estimation methodologies for well logging problems via a combination of fuzzy Kalman filter and different smoothers.* Journal of Petroleum Science and Engineering, June 2016

[32] ST. *STM32F765xx STM32F767xx STM32F768Ax STM32F769xx* [online]. Copyright © STMicroelectronics, 2017. Available at: https://www.st.com/resource/en/datasheet/stm32f767zi.pdf

[33] G. A. Terejanu. *Discrete Kalman Filter Tutorial* [online]. Department of Computer Science and Engineering, University at Buffalo, Buffalo, 2021. Available at: https://cse.sc.edu/~terejanu/files/tutorialKF.pdf

[34] V. Tereshkov. *A Simple Observer for Gyro and Accelerometer Biases in Land Navigation Systems.* Journal of Navigation 68, 2015. pp. 635-645. doi: 10.1017/S0373463315000016.

[35] Tersus GNSS. *BX305 GNSS Kit (HRS)* [online]. Copyright © Tersus GNSS, Inc., 2021. Available at: https://www.tersus-gnss.com/product/gnss-receiver-bx305-kit-high-end-radio

[36] u-blox. *Positioning chips and modules* [online]. Copyright © u-blox AG, 2018. Available at: https://www.u-blox.com/en/positioning-chips-and-modules

[37] *Unmanned Solutions* [online]. Copyright © Unmanned Solutions. Available at: http://www.unmansol.com

[38] Q. Xia, J. Duan, F. Gao, Q. Hu and Y. He. *Test Scenario Design for Intelligent Driving System Ensuring Coverage and Effectiveness.* Int. J. Automotive Technology 19(4), 2018, pp. 751-758

*) The publications in this section are listed alphabetically by the author's last name.

Author's Publications Related to the Doctoral Thesis

[Pub. 1] O. Lufinka. *Ultrasonic Transceiver with the Possibilities of the Data Communication and the Two-Point Distance Measurement.* International Conference on Applied Electronics (AE 2016), Pilsen, 2016, pp. 153-156. doi: 10.1109/AE.2016.7577262 (*1 citation on Web of Science*)

[Pub. 2] O. Lufinka. *Ultrazvukový vysílač a přijímač.* Funkční vzorek, RIV/49777513:23220/16:43930089, Pilsen, 2016

[Pub. 3] O. Lufinka. *Multiple-Point Ultrasonic Distance Measurement and Communication with Simulations.* 24th Telecommunications Forum (TELFOR), Belgrade, 2016, pp. 651-654. doi: 10.1109/TELFOR.2016.78188

[Pub. 4] O. Lufinka. *Analýza vlivu Hammingova kódu na přenosový kanál mezi ultrazvukovým vysílačem a přijímačem.* Elektrotechnika a informatika 2017. Elektrotechnika, elektronika, elektroenergetika, Pilsen, 2017, pp. 133-136

[Pub. 5] O. Lufinka. *Princip autonomních robotických platforem (ARP) používaných pro testování ADAS systémů.* Elektrotechnika a informatika 2018. Elektrotechnika, elektronika, elektroenergetika, Pilsen, 2018, pp. 81-84

[Pub. 6] O. Lufinka. *Concept of HW for Autonomous Robotic Platform (ARP) System Suitable for ADAS Testing.* Elektrotechnika a informatika 2019. Elektrotechnika, elektronika, elektroenergetika, Pilsen, 2019, pp. 57-60

[Pub. 7] O. Lufinka. *Concept of SW for Autonomous Robotic Platform (ARP) System Suitable for ADAS Testing.* Elektrotechnika a informatika 2020. Elektrotechnika, elektronika, elektroenergetika, Pilsen, 2020, pp. 61-64

[Pub. 8] O. Lufinka, J. Kadeřábek, J. Prstek, J. Skála, K. Kosturik. *Embedded Hardware and Software Design of Omnidirectional Autonomous Robotic Platform Suitable for Advanced Driver Assistance Systems Testing with Focus on Modularity and Safety.* International Conference on Automotive Electronics and Engine Electronics (ICAEEE), Prague, 2021, pp. 296-299 (*Best paper award*)

[Pub. 9] O. Lufinka, J. Kadeřábek, J. Prstek, J. Skála, K. Kosturik. *Omnidirectional Autonomous Robotic Platform for Advanced Driver Assistance Systems Testing – Movement, Localization and Navigation Possibilities.* Int. J. of Automotive Technology, 2021 (*In review*)

[Pub. 10] O. Lufinka. *Localization and Navigation of Autonomous Robotic Platform with Omnidirectional Movement for ADAS Testing.* Elektrotechnika a informatika 2021. Elektrotechnika, elektronika, elektroenergetika, Pilsen, 2021 (*In review*)

[Pub. 11] O. Lufinka, J. Kadeřábek, J. Prstek, J. Skála, K. Kosturik. *Comparison of Different Sensors and Localization Methods Suitable for Autonomous Robotic Platform Used in the Field of ADAS Testing.* 29th Telecommunications Forum (TELFOR), Belgrade, 2021 (*In review*)

*) The publications in this section are listed chronologically by the year of publication.

Other Author's Publications

[Pub. 12] O. Lufinka. *Návrh řídicí aplikace pro mobilní platformu.* Elektrotechnika a informatika 2015. Elektrotechnika, elektronika, elektroenergetika, Pilsen, 2015, pp. 169-172

[Pub. 13] O. Lufinka. *Design of the New Advanced Modes for the Mobile Platform Control Application.* 2015 23rd Telecommunications Forum (TELFOR), Belgrade, 2015, pp. 677-680. doi: 10.1109/TELFOR.2015.7377558

[Pub. 14] O. Lufinka. *16 segmentový LED displej s SPI komunikací.* Funkční vzorek, Pilsen, 2015

*) The publications in this section are listed chronologically by the year of publication.

Appendix A

3D Models

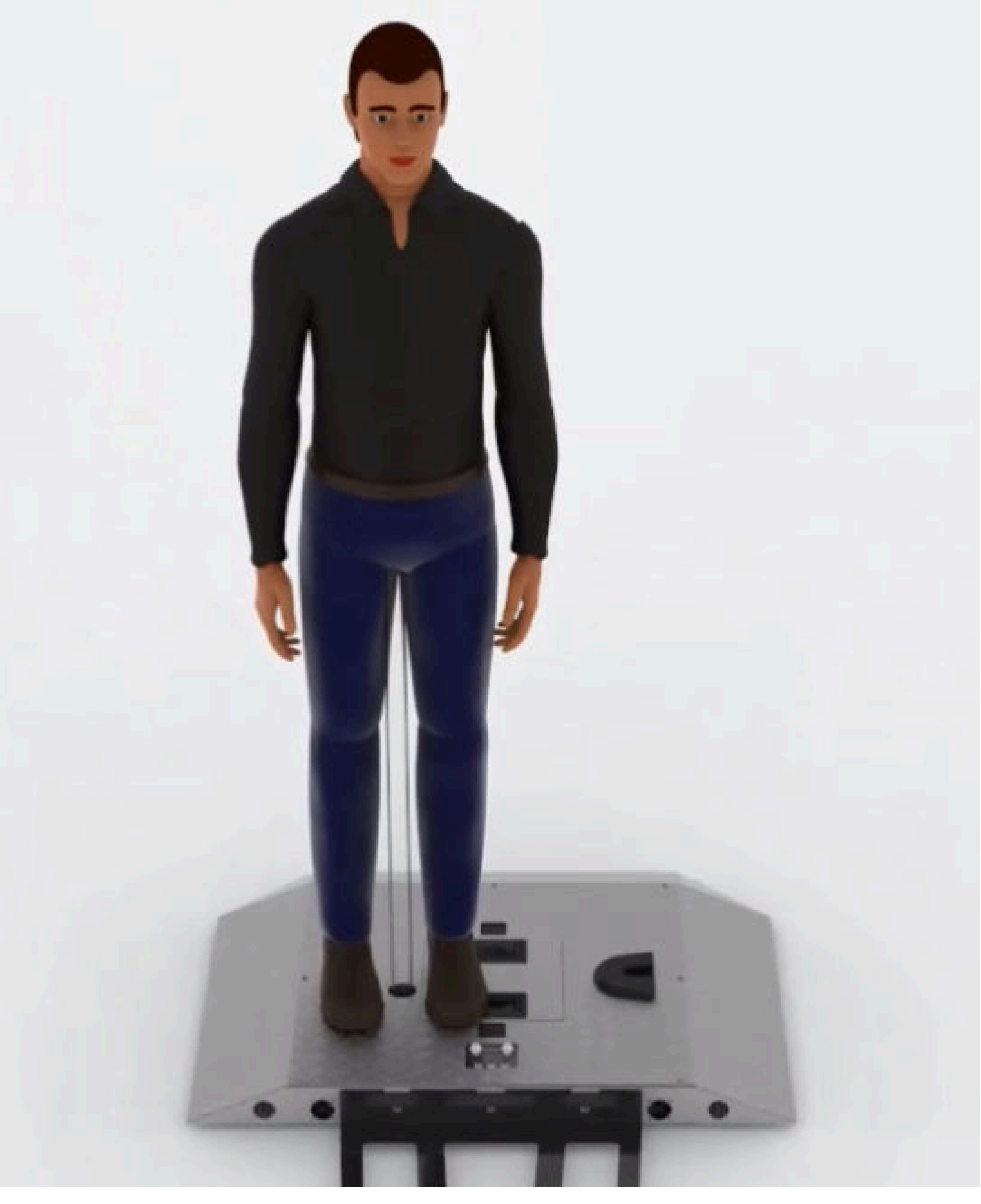
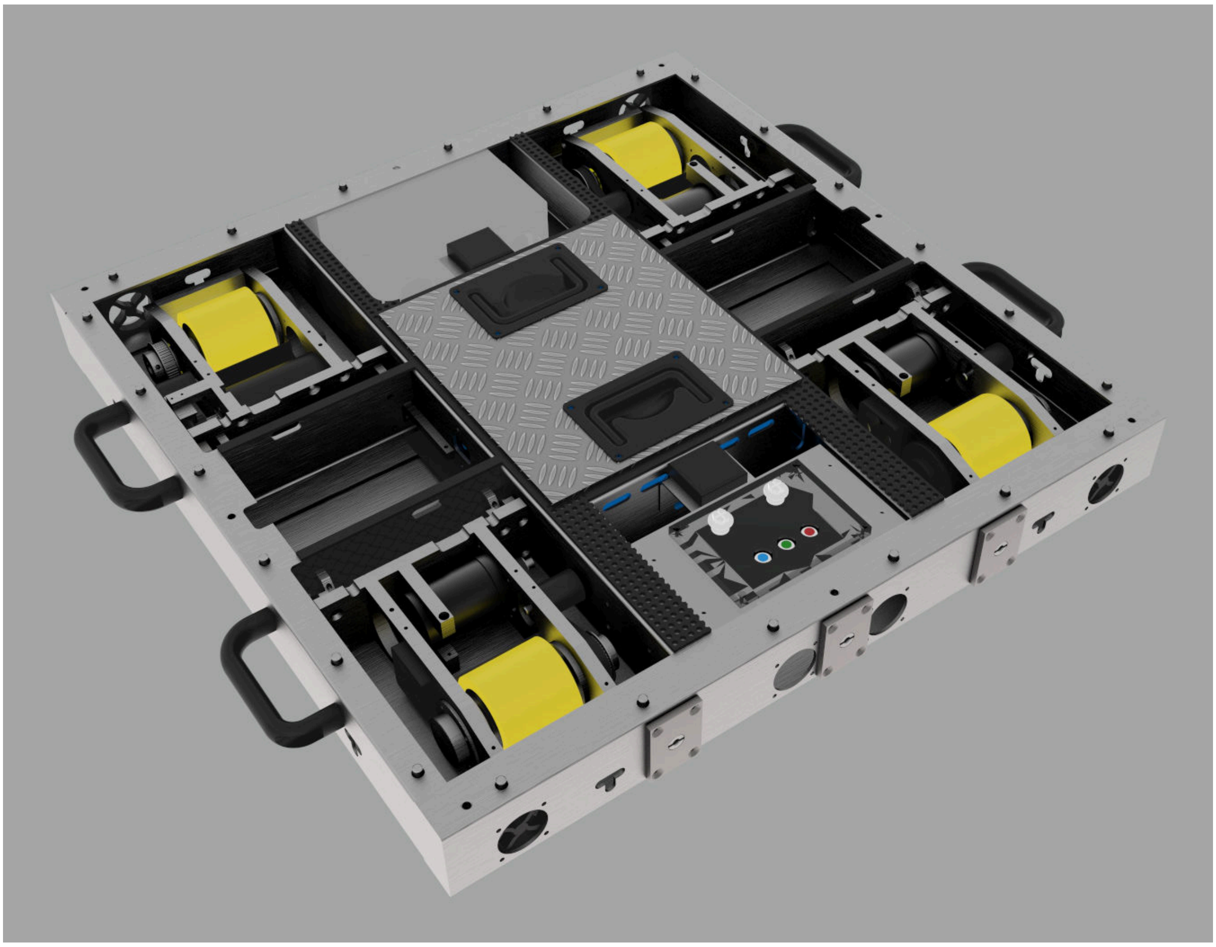
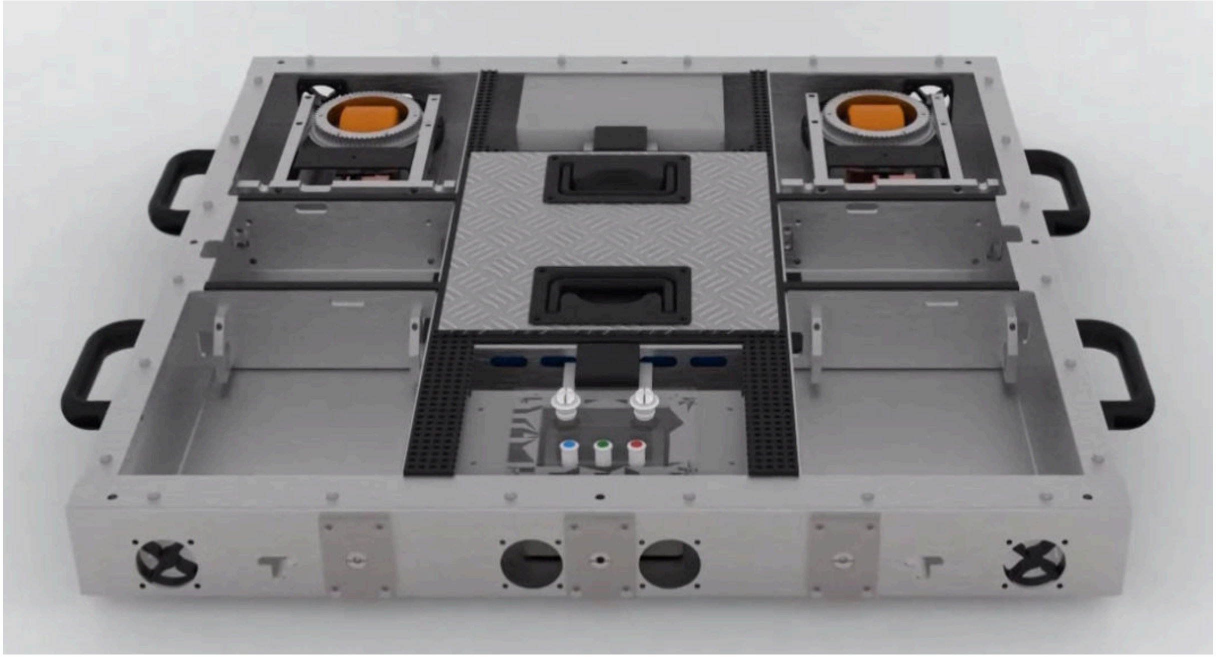


Fig. A.1: 3D model of the prototype



(a) Prototype with straight wheels



(b) Prototype with omnidirectional wheels

Fig. A.2: Detailed 3D models of the prototype

Appendix B

Photos

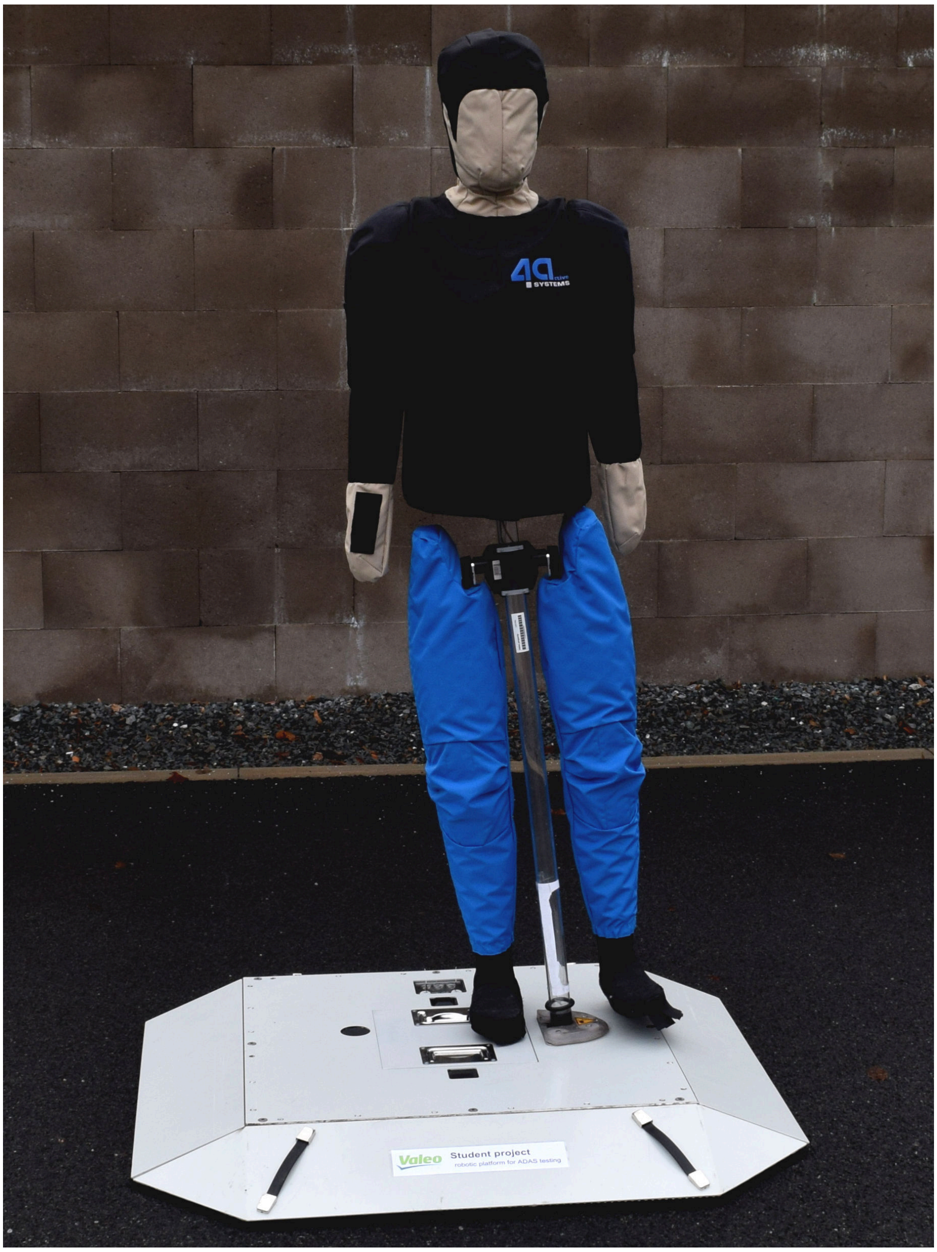


Fig. B.1: Photo of the prototype with a dummy pedestrian

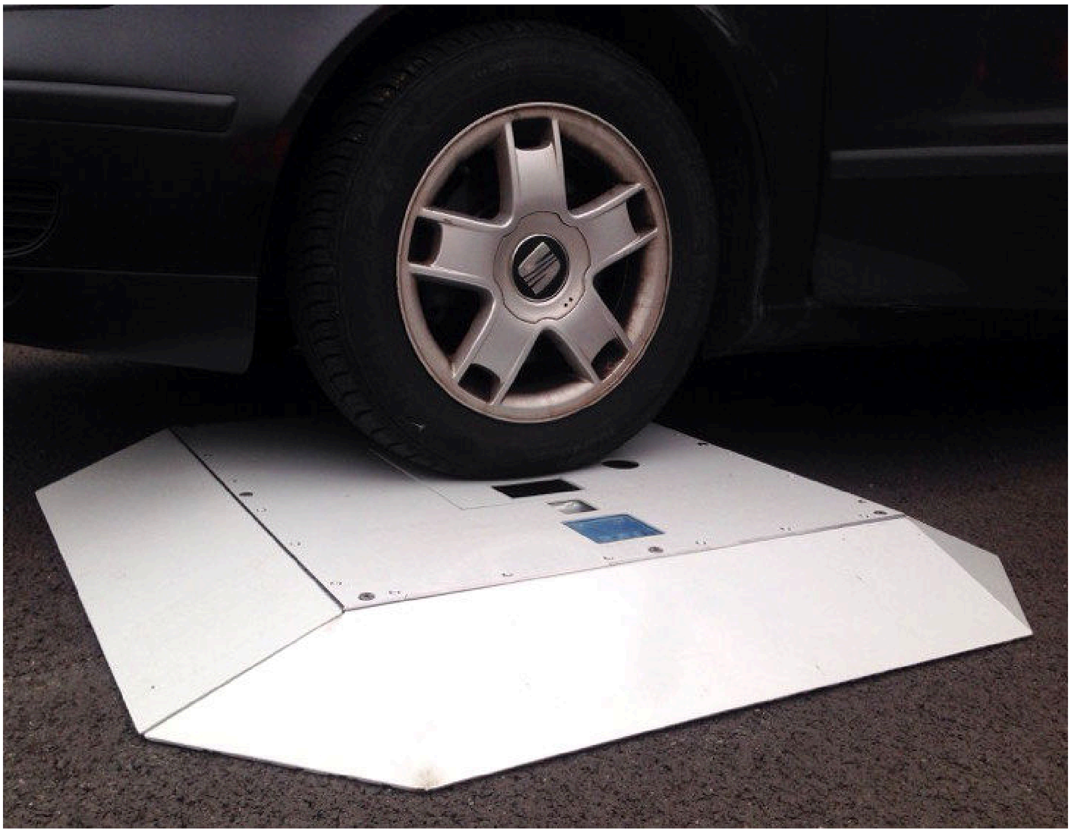


Fig. B.2: Photo of the prototype being driven over by car

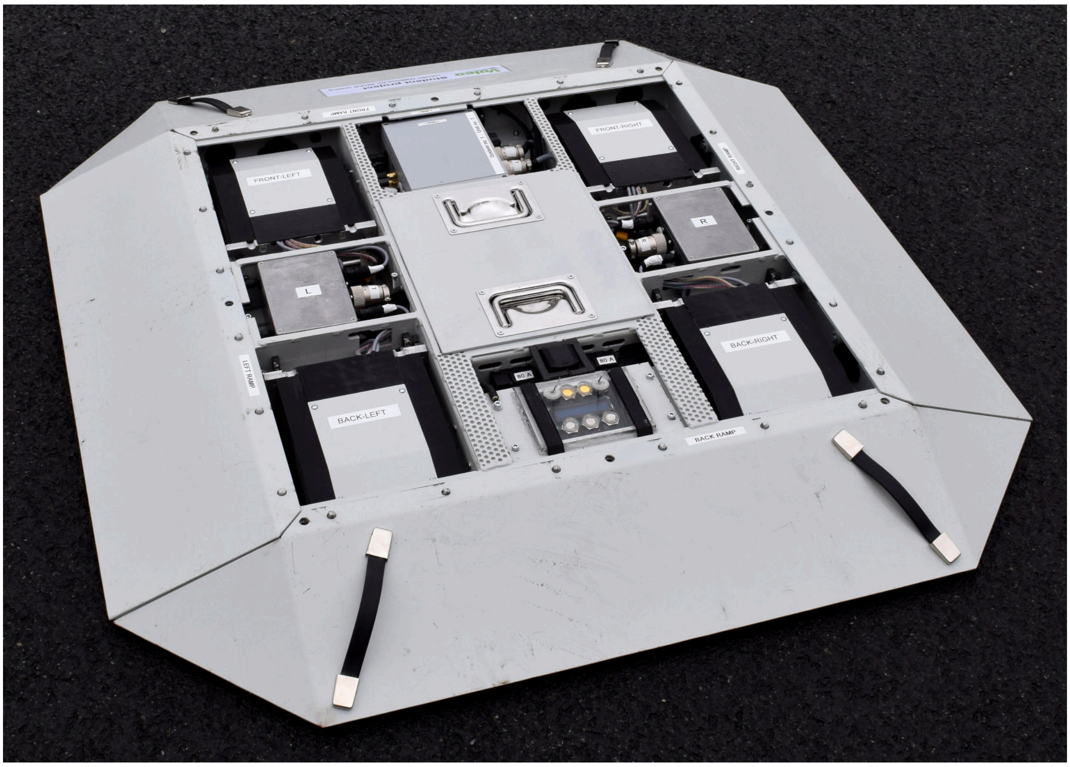


Fig. B.3: Photo of the prototype

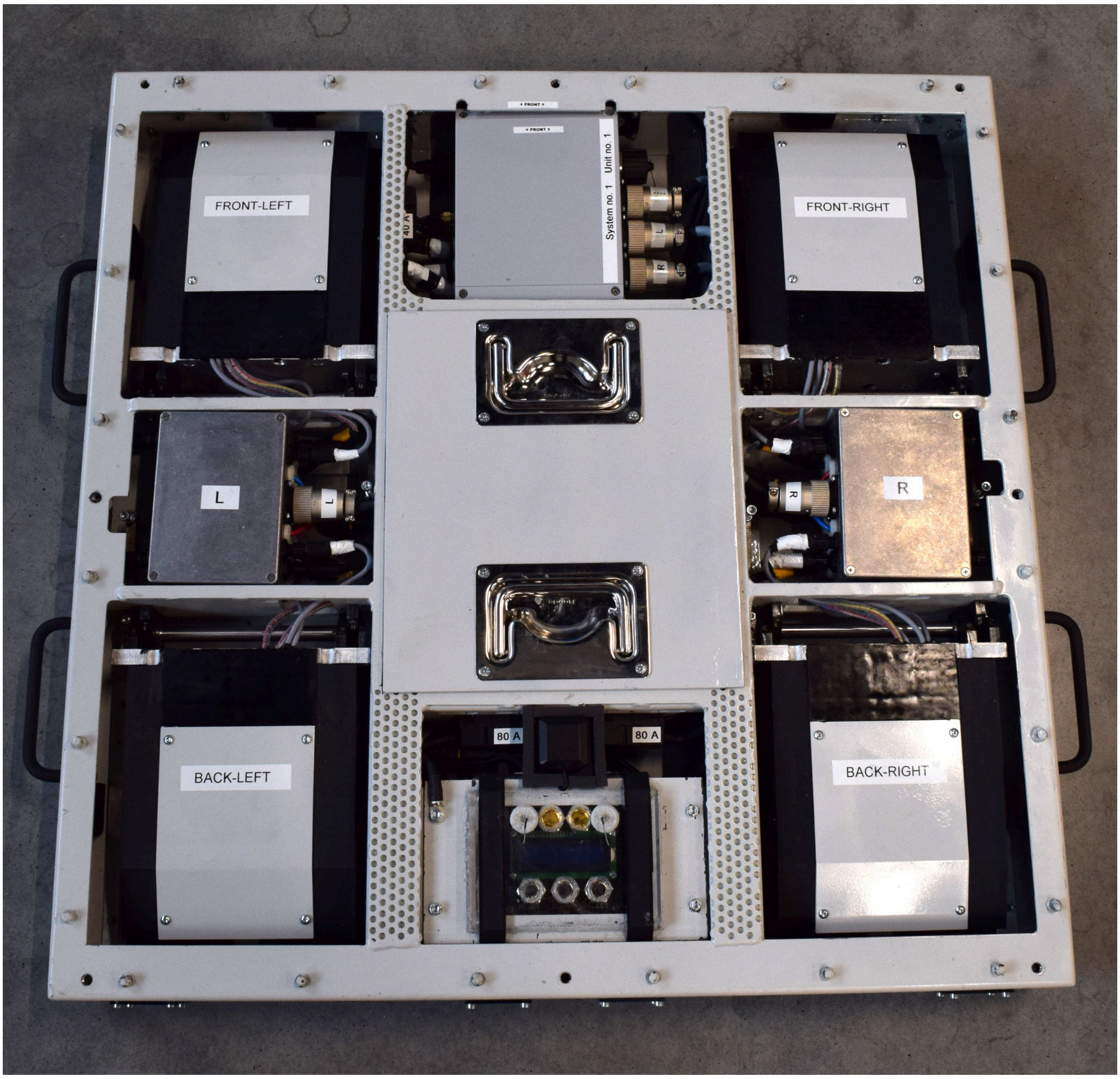
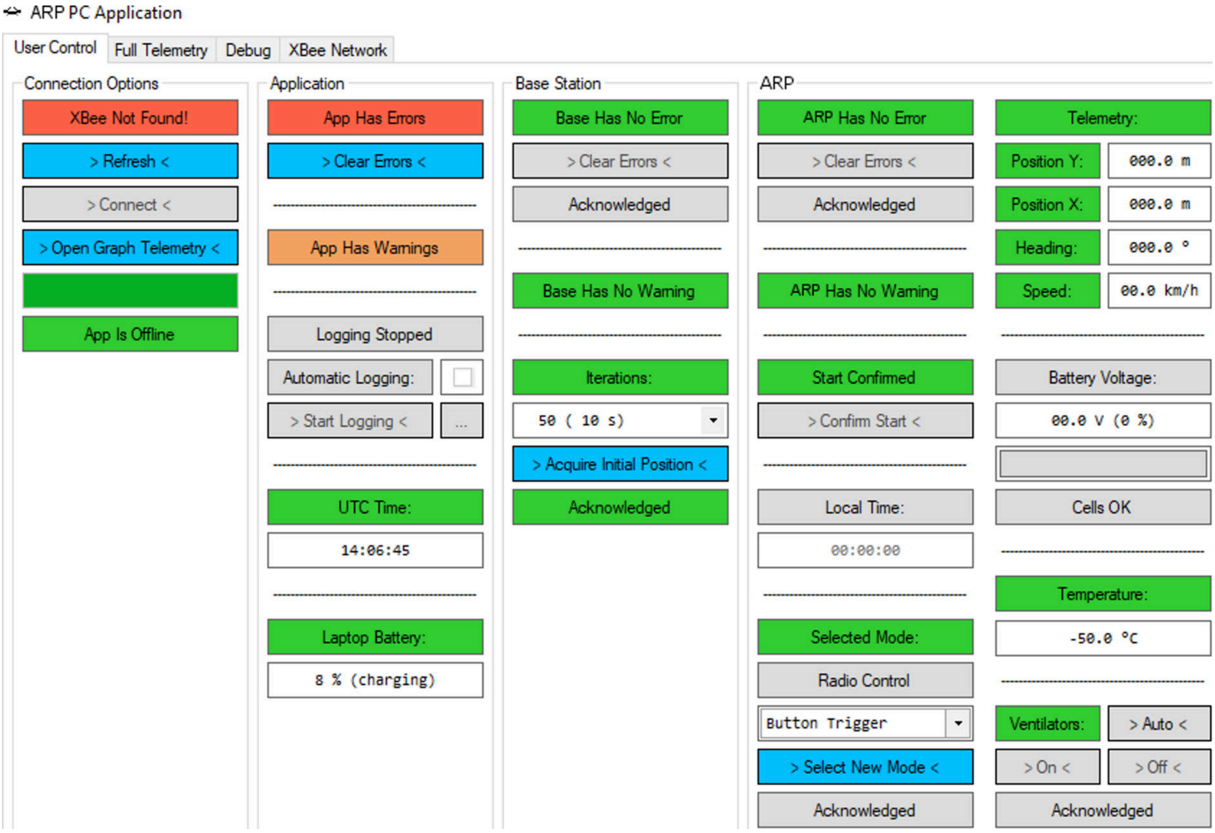


Fig. B.4: Detailed photo of the prototype

Appendix C

PC Application



Automatic Logging: □

UTC Time: 14:06:45

Laptop Battery: 8 % (charging)

Iterations: 50 (10 s)

Local Time: 00:00:00

Selected Mode: Radio Control

Button Trigger

Telemetry:

Position Y: 000.0 m

Position X: 000.0 m

Heading: 000.0 °

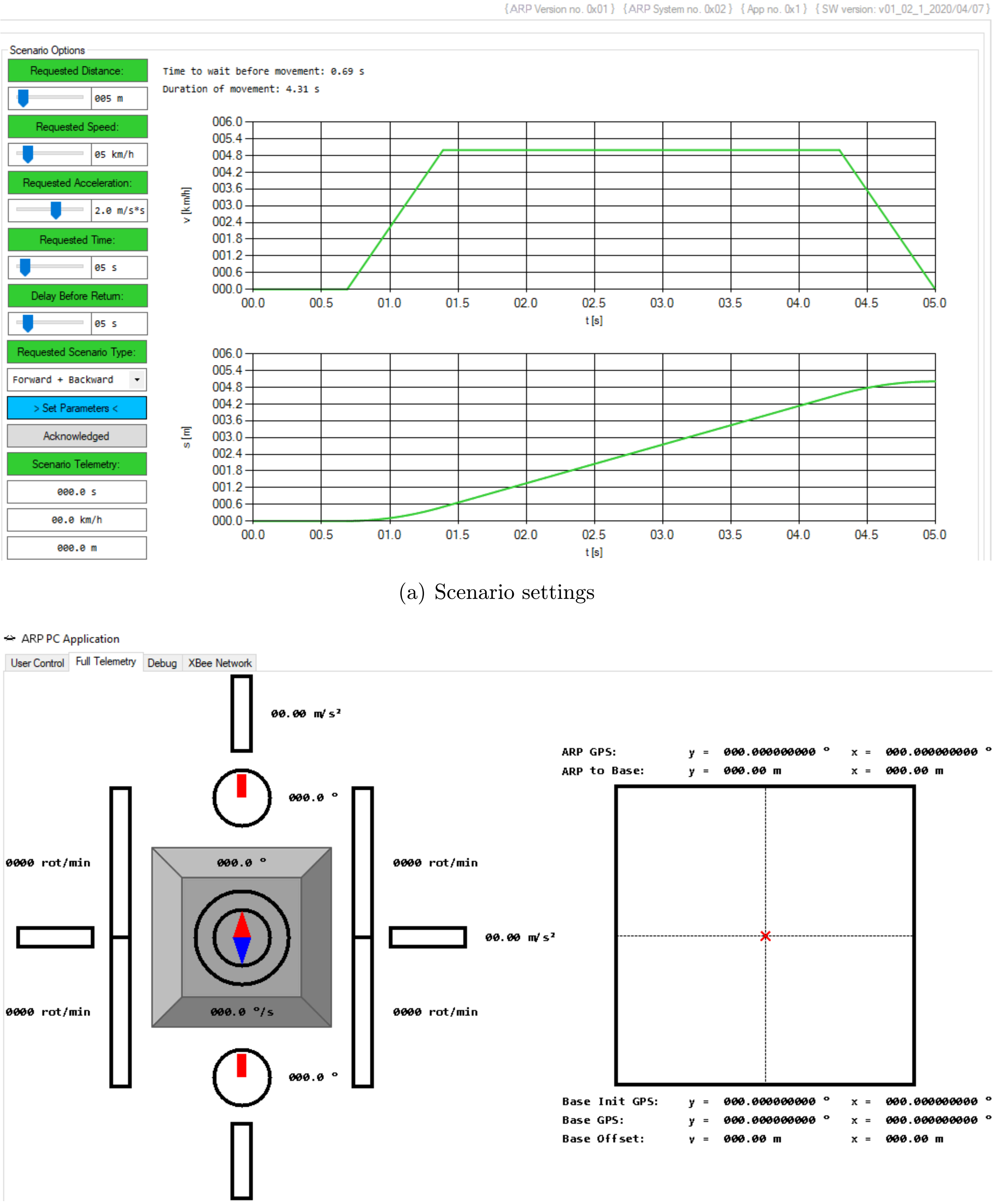
Speed: 00.0 km/h

Battery Voltage: 00.0 V (0 %)

Temperature: -50.0 °C

Ventilators:

Fig. C.1: Visualization of PC Application – user interface



{ARP Version no. 0x01} {ARP System no. 0x02} {App no. 0x1} {SW version: v01_02_1_2020/04/07}

Scenario Options

Requested Distance: 005 m

Requested Speed: 05 km/h

Requested Acceleration: 2.0 m/s*s

Requested Time: 05 s

Delay Before Return: 05 s

Requested Scenario Type: Forward + Backward

Scenario Telemetry: 000.0 s 00.0 km/h 000.0 m

```
Time to wait before movement: 0.69 s
Duration of movement: 4.31 s
```

(a) Scenario settings

ARP PC Application

```
ARP GPS:          y =  000.000000000 °     x =  000.000000000 °
ARP to Base:      y =  000.00 m            x =  000.00 m

Base Init GPS:    y =  000.000000000 °     x =  000.000000000 °
Base GPS:         y =  000.000000000 °     x =  000.000000000 °
Base Offset:      y =  000.00 m            x =  000.00 m
```

(b) Telemetry

Fig. C.2: Visualization of PC Application – scenario and telemetry

Appendix D

Kalman Filter Algorithm

There is the Kalman filter algorithm shown in this appendix. The example code is written in the Matlab software and it is used for the ARP localization. It combines values from different sensors according to the system model and error covariance matrices. The algorithm is explained in the section 7.4.

```
% System model ---------------------------------------------------------------------------- %
%                                                                                           %
%   x(k) = F(k-1) * x(k-1) + G(k-1) * u(k-1) + F(k-1) * w(k-1)                              %
%   z(k) = H(k)   * x(k)                     +          n(k)                                %
%                                                                                           %
%   F(k) = [1  0  sin(Phi(k))*delta_t  1/2*sin(Phi(k))*delta_t^2)  0  0      ]              %
%          [0  1  cos(Phi(k))*delta_t  1/2*cos(Phi(k))*delta_t^2)  0  0      ]              %
%          [0  0  1                    delta_t                     0  0      ]              %
%          [0  0  0                    1                           0  0      ]              %
%          [0  0  0                    0                           1  delta_t]              %
%          [0  0  0                    0                           0  1      ]              %
%                                                                                           %
%   G(k) = [0] ... input/control vector is has no effect (is always zero)                   %
%          [0]                                                                              %
%          [0]                                                                              %
%          [0]                                                                              %
%          [0]                                                                              %
%          [0]                                                                              %
%                                                                                           %
%   H(k) = [1  0  0  0  0  0] ... observation matrix is identity matrix                     %
%          [0  1  0  0  0  0]                                                               %
%          [0  0  1  0  0  0]                                                               %
%          [0  0  0  1  0  0]                                                               %
%          [0  0  0  0  1  0]                                                               %
%          [0  0  0  0  0  1]                                                               %
%                                                                                           %
%   x(k) = [d_lat_filt(k) ]    u(k) = [0]    z(k) = [d_lat_meas(k) ]                        %
%          [d_long_filt(k)]                         [d_long_meas(k)]                        %
%          [v_x_filt(k)   ]                         [v_x_meas(k)   ]                        %
%          [a_x_filt(k)   ]                         [a_x_meas(k)   ]                        %
%          [Phi_filt(k)   ]                         [Phi_meas(k)   ]                        %
%          [omega_filt(k) ]                         [omega_meas(k) ]                        %
%                                                                                           %
% ----------------------------------------------------------------------------------------- %

% Kalman filter initialization ------------------------------------------------------------ %

% Initial observation (measurement) matrix is identity matrix
H(1) = [1, 0, 0, 0, 0, 0;
        0, 1, 0, 0, 0, 0;
        0, 0, 1, 0, 0, 0;
        0, 0, 0, 1, 0, 0;
        0, 0, 0, 0, 1, 0;
        0, 0, 0, 0, 0, 1];

```

```
% Initial observation (measurement) vector is set from initial measured values
z(1) = [d_lat_meas(1) ;
        d_long_meas(1);
        v_y_meas(1)   ;
        a_y_meas(1)   ;
        Phi_meas(1)   ;
        omega_meas(1) ];

% Initial covariance matrix for observation (measurement) noise
R(1) = [r1, 0 , 0 , 0 , 0 , 0 ;
        0 , r2, 0 , 0 , 0 , 0 ;
        0 , 0 , r3, 0 , 0 , 0 ;
        0 , 0 , 0 , r4, 0 , 0 ;
        0 , 0 , 0 , 0 , r5, 0 ;
        0 , 0 , 0 , 0 , 0 , r6; % r1-6 ... initial error of each value from vector z
                                % (the bigger the error, the less trust to the value)

% Initial state transition matrix (measured value is used as initial heading)
F(1) = [1, 0, sin(Phi_meas(1))*delta_t, 0.5*sin(Phi_meas(1))*delta_t*delta_t, 0, 0      ;
        0, 1, cos(Phi_meas(1))*delta_t, 0.5*cos(Phi_meas(1))*delta_t*delta_t, 0, 0      ;
        0, 0, 1                       , delta_t                             , 0, 0      ;
        0, 0, 0                       , 1                                   , 0, 0      ;
        0, 0, 0                       , 0                                   , 1, delta_t;
        0, 0, 0                       , 0                                   , 0, 1     ];

% Initial observation (measurement) vector is used as initial state vector
x(1) = z(1);

% Save initial state vector to filtered values vectors
[d_lat_filt(1); d_long_filt(1); v_y_filt(1); Phi_filt(1); omega_filt(1)] = x(1);

% Initial covariance matrix of state vector
% (if p close to 0 ... system is trusted from the beginning - x(1) must be set correctly;
%  if p close to Inf ... system is not trusted - x(1) can be set to random values)
p = 0.1;
P(1) = [p, 0, 0, 0, 0, 0;
        0, p, 0, 0, 0, 0;
        0, 0, p, 0, 0, 0;
        0, 0, 0, p, 0, 0;
        0, 0, 0, 0, p, 0;
        0, 0, 0, 0, 0, p];

% Initial covariance matrix for process noise
Q(1) = [q , 0 , 0 , 0 , 0 , 0 ;
        0 , q , 0 , 0 , 0 , 0 ;
        0 , 0 , q , 0 , 0 , 0 ;
        0 , 0 , 0 , q , 0 , 0 ;
        0 , 0 , 0 , 0 , q , 0 ;
        0 , 0 , 0 , 0 , 0 , q ]; % q ... initial error of system
k = 1;                           % (the bigger the error, the less trust to the system)

% Kalman filter iterations --------------------------------------------------------------- %
while (k++)

  % Set current observation (measurement) matrix (it is the same all the time)
  H(k) = H(1);

  % Set current observation (measurement) vector from current measured values
  z(k) = [d_lat_meas(k) ;
          d_long_meas(k);
          v_x_meas(k)   ;
          a_x_meas(k)   ;
          Phi_meas(k)   ;
          omega_meas(k) ];

  % Set current covariance matrix for observation (measurement) noise
  R(k) = [r1, 0 , 0 , 0 , 0 , 0 ;
          0 , r2, 0 , 0 , 0 , 0 ;
          0 , 0 , r3, 0 , 0 , 0 ;
          0 , 0 , 0 , r4, 0 , 0 ;
          0 , 0 , 0 , 0 , r5, 0 ;
          0 , 0 , 0 , 0 , 0 , r6; % r1-6 ... current error of each value from vector z

  % Prediction of the Kalman filter - time update
  x_apriori(k) = F(k-1) * x(k-1);
  P_apriori(k) = F(k-1) * P(k-1) * F(k-1)' + Q(k-1);

  % Correction of the Kalman filter - observation (measurement) update
  K(k) = P_apriori(k) * H(k)' * inv(H(k) * P_apriori(k) * H(k)' + R(k));
  x(k) = x_apriori(k) + K(k) * (z(k) - H(k) * x_apriori(k));
  P(k) = (eye(6) - K(k) * H(k)) * P_apriori(k);

  % Save current state vector to the filtered values vectors
  [d_lat_filt(k); d_long_filt(k); v_x_filt(k); a_x_filt(k); Phi_filt(k); omega_filt(k)] = x(k);

  % System model update for the next iteration (uses current filtered heading)
  F(k) = [1, 0, sin(Phi_filt(k))*delta_t, 0.5*sin(Phi_filt(k))*delta_t*delta_t, 0, 0      ;
          0, 1, cos(Phi_filt(k))*delta_t, 0.5*cos(Phi_filt(k))*delta_t*delta_t, 0, 0      ;
          0, 0, 1                       , delta_t                             , 0, 0      ;
          0, 0, 0                       , 1                                   , 0, 0      ;
          0, 0, 0                       , 0                                   , 1, delta_t;
          0, 0, 0                       , 0                                   , 0, 1      ];

  % State vector covariance matrix update for next iteration
  Q(k) = [q , 0 , 0 , 0 , 0 , 0 ;
          0 , q , 0 , 0 , 0 , 0 ;
          0 , 0 , q , 0 , 0 , 0 ;
          0 , 0 , 0 , q , 0 , 0 ;
          0 , 0 , 0 , 0 , q , 0 ;
          0 , 0 , 0 , 0 , 0 , q ]; % q ... current error of the system

  % Check if k is not out of bounds of input vectors
  if (k > size(d_lat_meas)(2) ||
      k > size(d_long_meas)(2) ||
      k > size(v_x_meas)(2) ||
      k > size(a_x_meas)(2) ||
      k > size(Phi_meas)(2) ||
      k > size(omega_meas)(2))
    break;
  endif

end
```

Appendix E

Localization Graphs

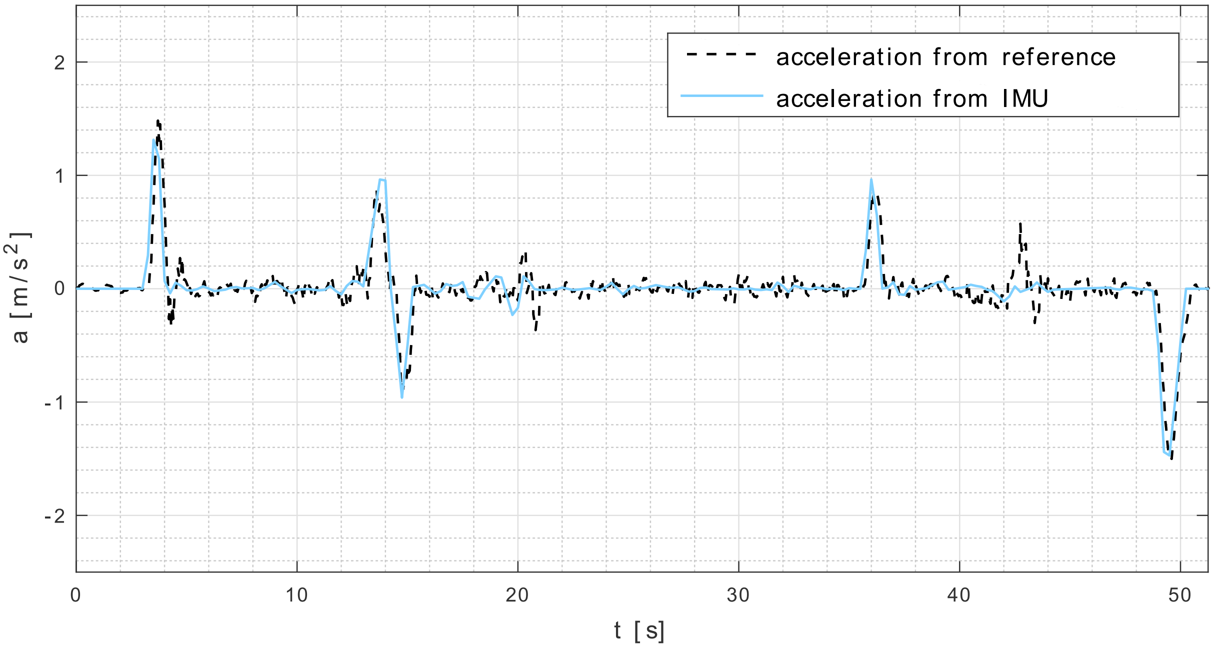


Fig. E.1: Graph of IMU acceleration measurement

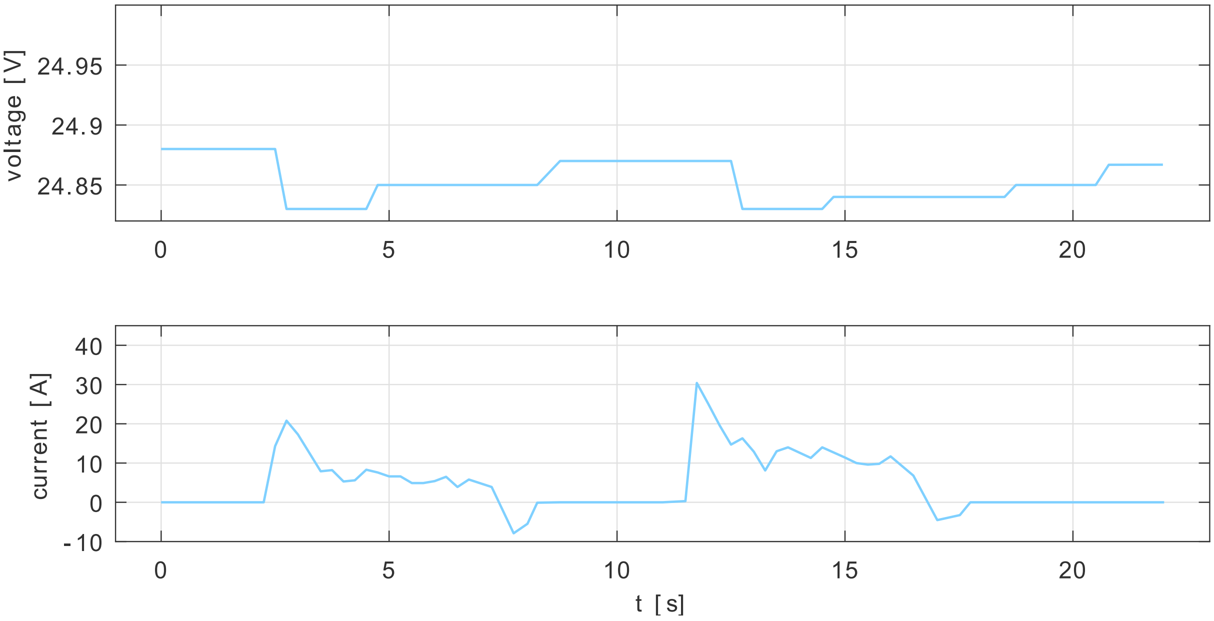


Fig. E.2: Graph of voltage and current measurement

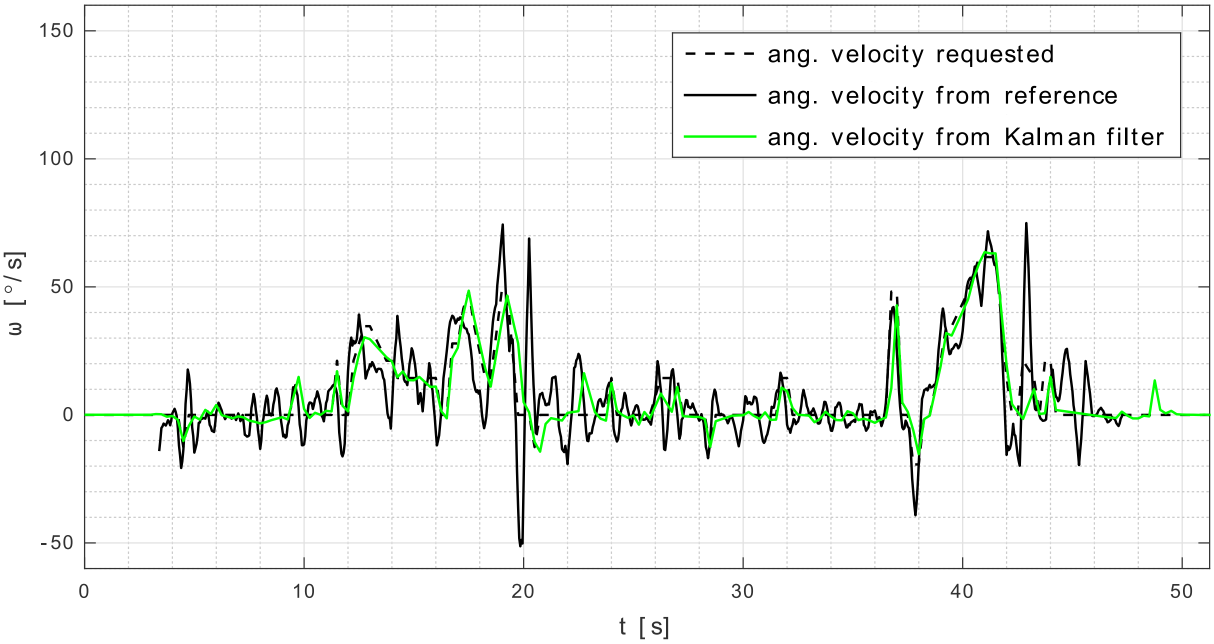


Fig. E.3: Graph of Kalman filter angular velocity result

Appendix F

P2P Ultrasonic Distance Measurement

F.1 Measurement Results

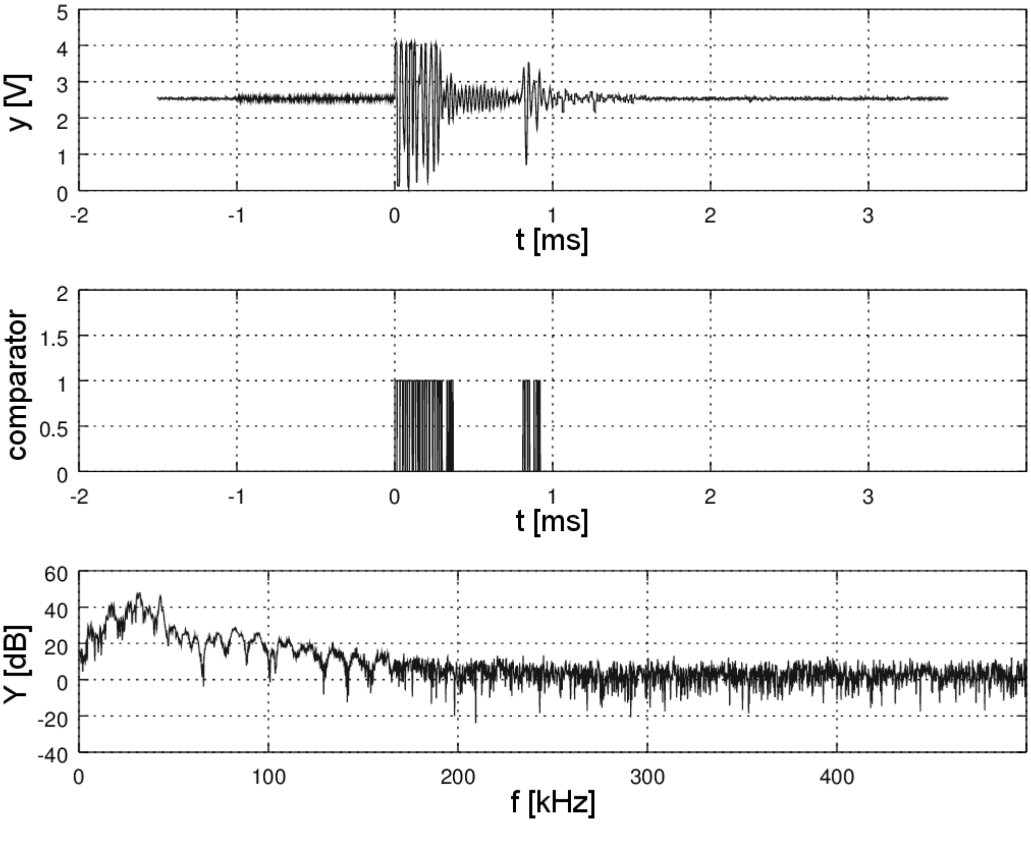


Fig. F.1: Detection of ultrasonic noise on receiver side

F.2 Schematics

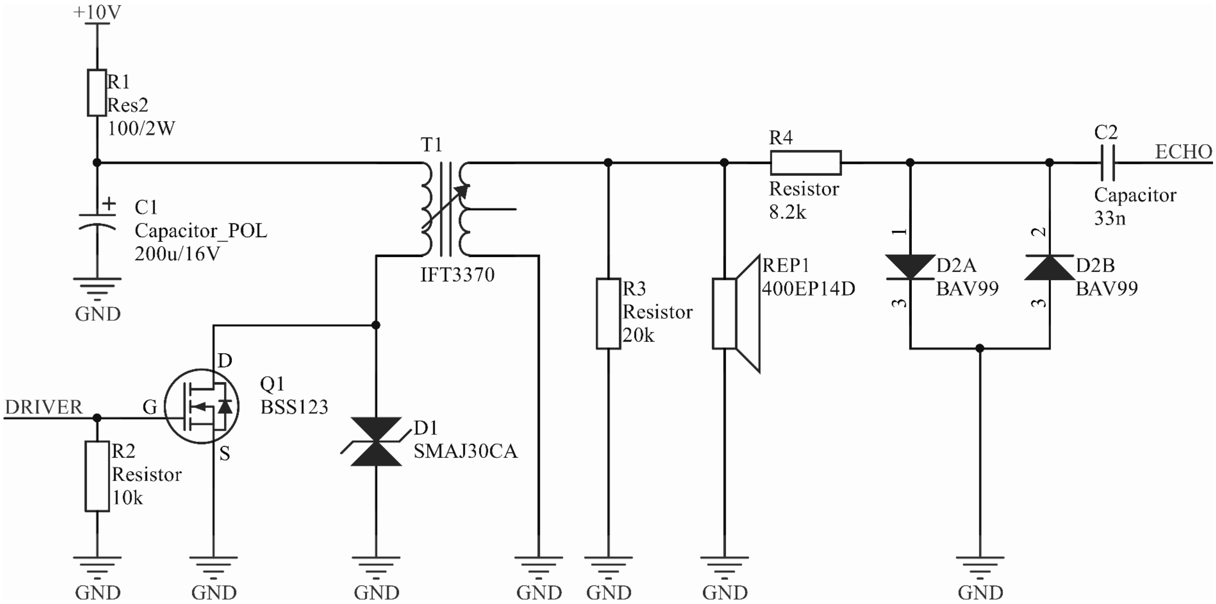


Fig. F.2: Schematic of ultrasonic transmitter

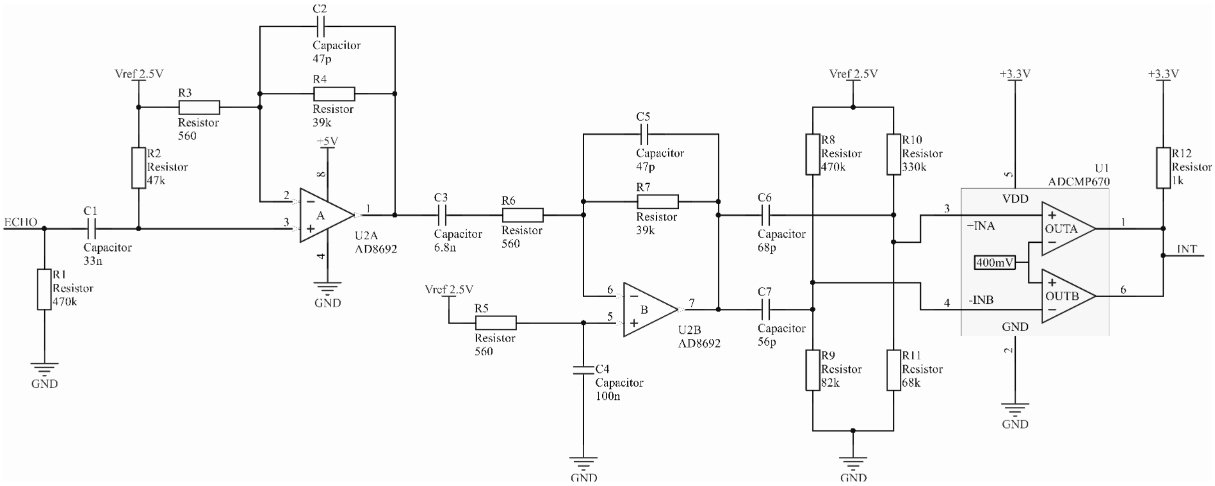


Fig. F.3: Schematic of ultrasonic receiver

F.3 Simulations

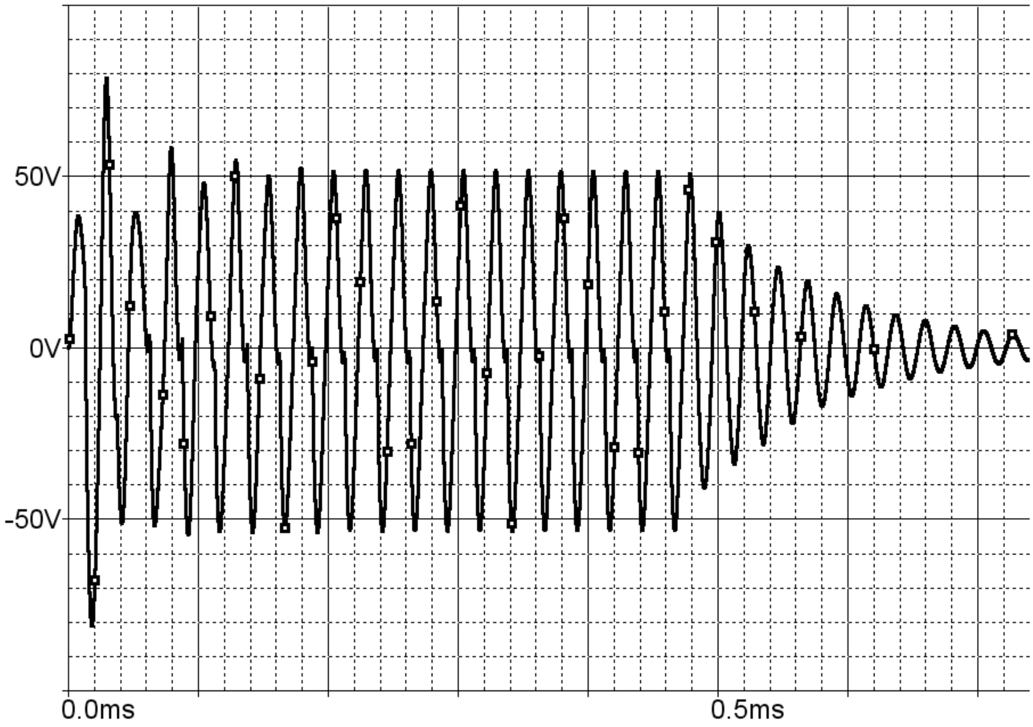


Fig. F.4: Simulation of non-optimized ultrasonic transmitter output

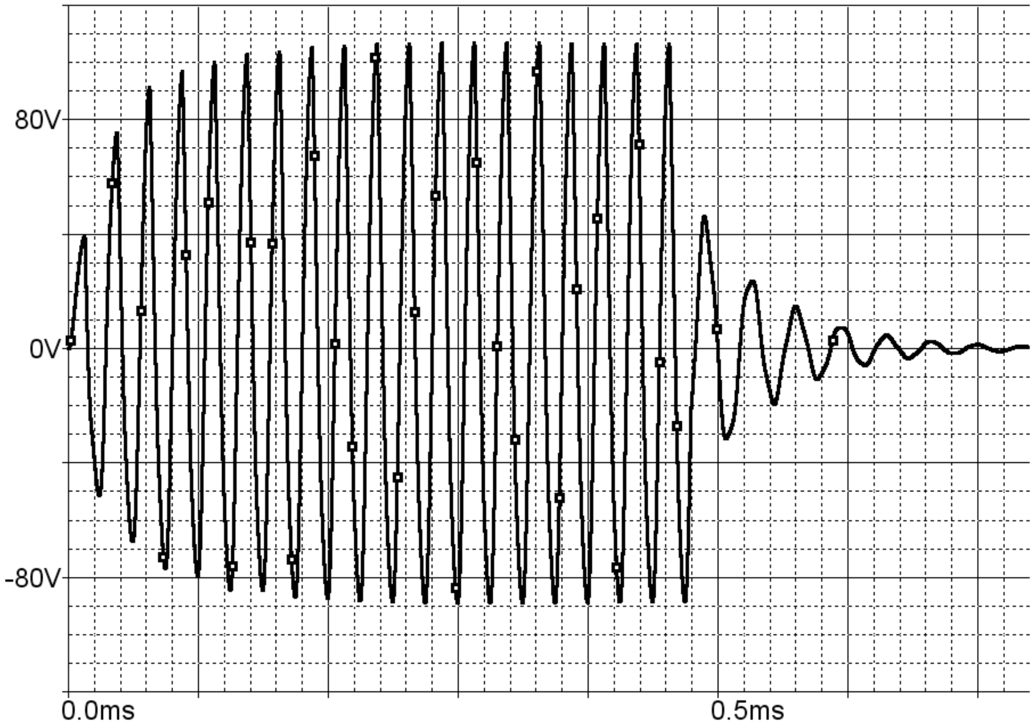


Fig. F.5: Simulation of optimized ultrasonic transmitter output

Index