ELAD - Easy Learning Autonomous Drive
Project status and SOTA for Self-Driving Vehicles

Gabriel Andersson Santiago
Nikhil Bohra Nirmal Kumar
Agnes Jernström
Fangjin Kuang
Henrik Ljunggren
Ramin Seyed Farshchi
Ylva Steffner

May 22, 2016

Abstract

Intelligent vehicles has become a major research theme in intelligent transportation systems. The goal is to augment vehicle autonomous driving either entirely or partly for the purpose of safety, comfort and energy saving. A self-driving vehicle independent of human control consists of four fundamental technologies; Observation, Orientation, Path planning and Decision making and Motion control.

ELAD - easy learning autonomous drive, is a project that aim at developing a learning platform for autonomous driving for the consultancy company ÅF. A small car in the size 1 to 10 of a real car is used. In order to decide direction for the ELAD project a State of the art of autonomous driving for both miniature vehicles and full-sized cars was made. The state of the art was focused around the first three fundamental technologies. This report gives a status report of the current status of the ELAD project and present the state of the art of self driving cars that was made.

The State of the art showed that all full-sized vehicles used LIDAR(s) and Radar(s) for observation. A camera was also a popular choice and was the most important sensor for observation for the smaller vehicles. When it came to orientation all vehicles used some sort of IMU to determine the vehicle direction. In some projects wheel encoders was used to determine distance traveled. Some sort of filter was used in most projects to get a more precise estimation of the vehicles position. For decision making all vehicles used a state machine to decide the behavior of the vehicle. Based on the state of the art is was decided that the Elad project should ....

Acknowledgements

We would like to thank our supervisors at KTH, Martin and Naveen and our supervisors at ÅF, Carl-Johan, Xavier, Jan, Peter and David for feedback and support throughout the project.

Gabriel Andersson Santiago, Nikhil Bohra Nirmal Kumar,
Agnes Jernström, Fangjin Kuang, Ylva Steffner,
Henrik Ljunggren and Ramin Seyed Farshchi

Stockholm, May 2016

Contents

1 Introduction

1.1 Background

The name automobile comes from the Greek words auto, meaning “self”, and mobile, meaning “moving” [1]. Back then, "self moving" referred to the car moving without a horse. Now, the term "self moving" brings to mind truly self driving cars, i.e. cars moving without any human input. The field of intelligent vehicles has become a major research theme in intelligent transportation systems, with companies like Google, GM, Scania and Volvo all having successful projects in the area. The goal is to augment vehicle autonomous driving either entirely or partly for the purpose of safety, comfort and energy saving.

A self-driving vehicle independent of human control consists of four fundamental technologies as shown in figure 1 and listed below:

1. Observation
2. Orientation
3. Path planning and decision-making
4. Motion control

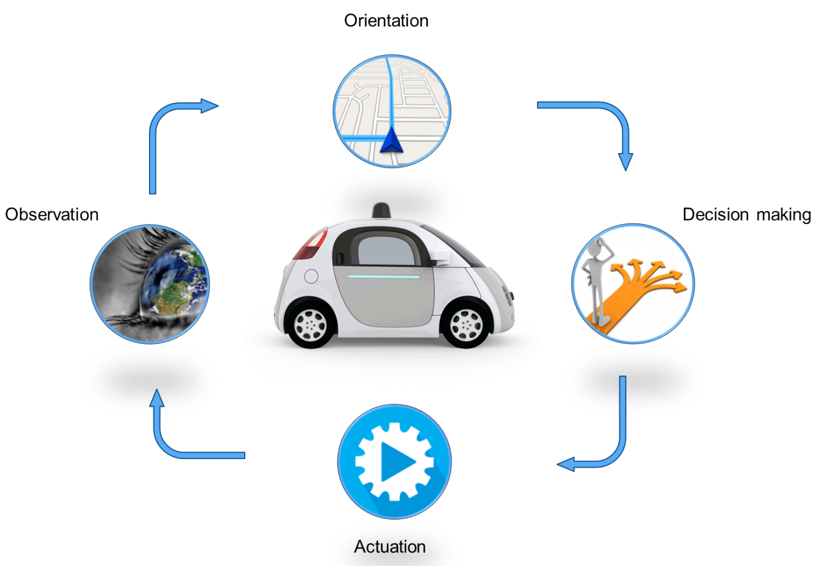

Figure 1: Fundamental areas for self-driving vehicles.

The observation module provides a model of the surrounding environment by sensing the environment structures in a multi-sensor way.

The second module, orientation, determines the vehicle's position by using geometric feature location estimations in the environment model. It also interprets sensor information to estimate the locations of geometric features and as a result yields a global map.

The purpose of the path planning and decision-making module is to ensure that the vehicle follows the rules regarding safety, vehicle dynamics and environment contexts etc. This module generates the desired path.

The final module, motion control, executes the commands necessary to achieve the desired path. [3]

In order to research autonomous driving, miniature vehicles in the scale 1:10 of a real car can be used. This is what the ELAD - Easy Learning Autonomous Drive, project aims at. The ELAD project is a student project for the "Advanced Course in Mechatronics" (MF2058) at KTH. The project is issued by the consulting company ÅF. The aim of the project is to improve ÅF:s teaching platform about autonomous vehicles so that consultants at ÅF quickly can learn, test and research autonomous drive in order to be prepared for customer projects in that area.

1.2 Purpose

The purpose of this report is to give a brief overview of the state of the art of self-driving vehicles in order to determine the direction of the ELAD project. The purpose is also to give a status report about the progress in the ELAD project.

1.3 Scope

In this report "self-driving" denotes a vehicle that is able to:

- Follow a lane
- Avoid obstacles

without the help of any human input. The term "miniature vehicle" denotes a vehicle with the approximate length of 40 cm. The term "full-sized vehicle" denotes a vehicle that has the capacity of transporting an adult human passenger.

The state of the art investigation of the vehicles will be about the first three of the four fundamental technologies for self-driving vehicles; observation, orientation and path planning and decision-making, as described in the background.

The fourth fundamental technology, motion control (or actuation), will not be evaluated.

For the miniature vehicles, only projects developed in the last four years have been chosen in order to make sure that they accurately reflect the state of the art of the subject. For the full-sized vehicles that time is 10 years, due to the limited amount of available source material. Except projects from DARPA and Carolo Cup some similiar projects to these will also be evaluated.

1.4 Method

The research has been done through literature study of self-driving vehicles in general and the technologies behind them. Five different self-driving vehicles were more closely examined, three miniature vehicles and two full-sized ones.

2 Status report

ELAD stands for easy learning autonomous drive. The team consists of seven members. The team have had two days a week to work on and Scrum has been used to manage the project process. This chapter present the work done during the first seven weeks of the project, divided into three different sprints.

2.1 Sprint one

In the first sprint the project was kick-started with a visit to ÅF where the team got an introduction to the scrum tool Jira and picked up hardware to use in the project. The hardware was a miniature vehicle, a motherboard, encoders, a LIDAR (Light detection and ranging) and a camera.

Figure 2: The miniature toy vehicle obtained from ÅF.

This sprint was focused on researching hardware typically used in autonomous driving. The hardware researched was LIDARs, cameras, radars and GPS. See Appendix A for information about LIDAR. Research about fundamental technologies in autonomous drive (see for example the information in Appendix A), artificial intelligence (Appendix B), the history of autonomous drive (Appendix C) and mapping (Appendix E) was also made. We also began to investigate how to build the real life test track.

2.2 Sprint two

The second sprint was more concentrated on practical work. The car was disassembled and the different hardware such as LIDAR and camera were put in use with the purpose to gather some test data from them, see figure

3 and 4. The motherboard was also connected and a simple 'Hello world' program was performed on it.

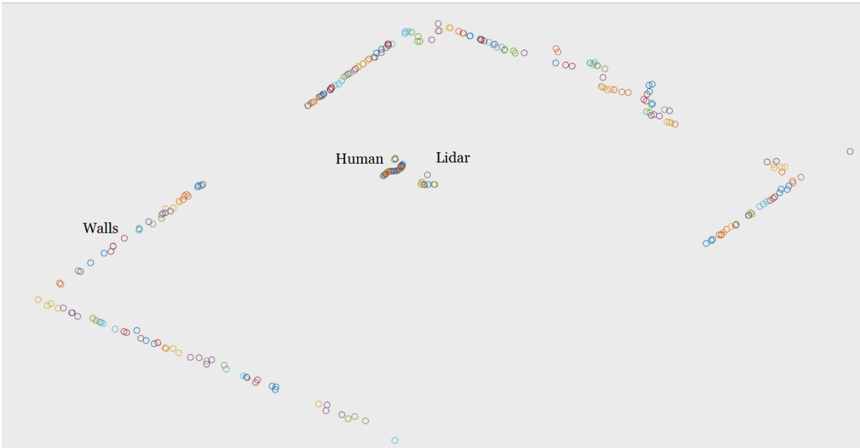

Figure 3: The LIDAR was connected and tried out in the HK-room. The figures shows the human holding the LIDAR, the LIDAR and the walls of the HK-room.

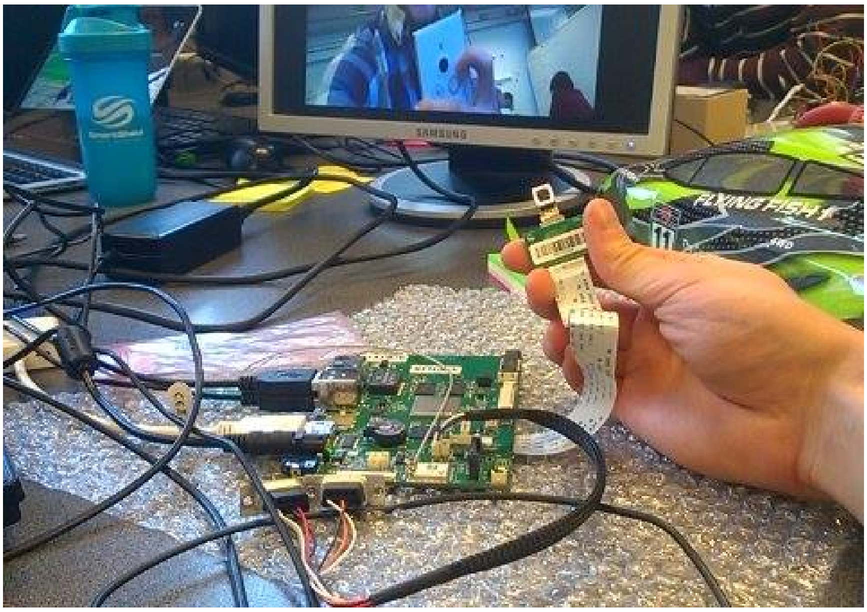

Figure 4: The camera was connected and tested.

We also started to investigate the possibility of using Simulink in order to create a virtual world for testing, see figure 5.

During this sprint requirements for the end product begun to be formulated. See appendix D for the resulting requirements.

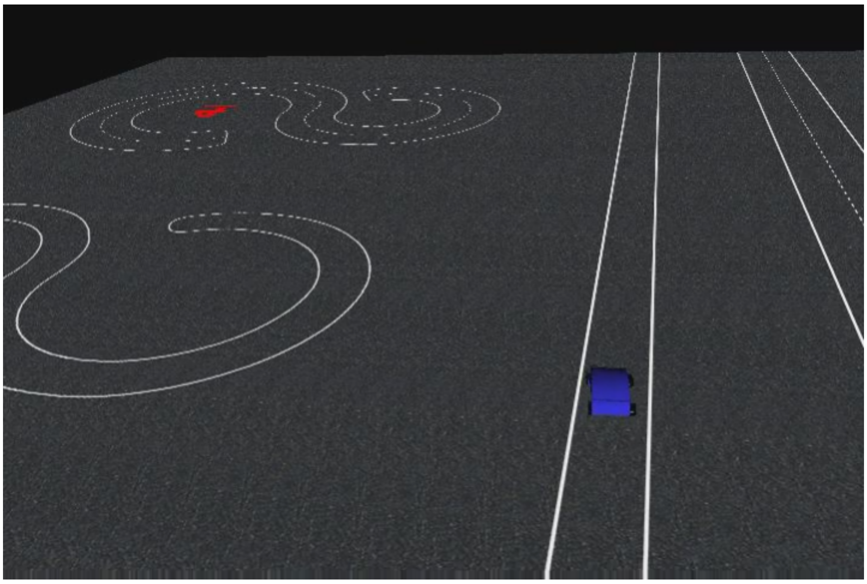

Figure 5: Test track simulink

2.3 Sprint three

In the last sprint different projects in autonomous driving was investigated in order to generate a state of the art about autonomous driving. In this sprint this report was also created. In-depth researches on requirements, diversity and inclusion in organization, project managment, and life cycle assessment were done as well during this sprint.

3 State of the art

This section will present five different self-driving vehicles and cover which methods they used to achieve the fundamental technologies presented in the scope. Several of these vehicles has competed in the Carolo Cup or the DARPA competition.

Carolo Cup has been organized by the Technische Universitat Braunschweig in Germany each year since 2008. In the competition several vehicles in scale 1 : 10 compete in autonomous driving, with tasks such as lane following, parking and obstacle avoidance. The aim is to learn about, test and research autonomous driving. [4]

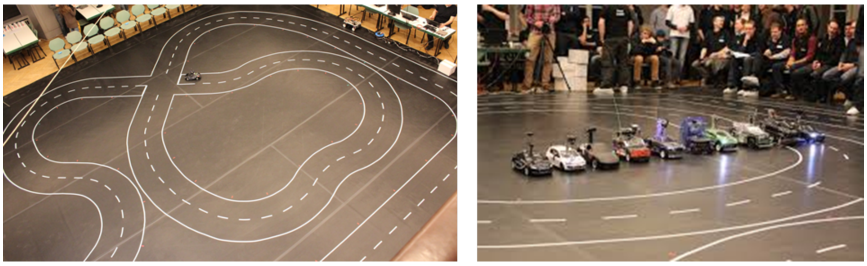

Figure 6: Track and start field for Carolo Cup

DARPA is a competition for full sized vehicles and was first held in 2004 with the task for the vehicles to autonomously navigate in the Mojave Desert. Year 2007 the competition had been extended to also include a mock city environment. [6].

Figure 7: The winning vehicle of DARPA year 2007.

In this chapter some of the project cars that have competed the last couple of years in Carolo or DARPA, or projects similar to these, are briefly investigated.

3.1 Berlin United

The team “Berlin United" from Freie Universitat Berlin participated in Carolo Cup in 2012 and their vehicle platform was used to do a master thesis by Lukas Maischak about lane localization for autonomous vehicles in 2014. [7] The vehicle used can be seen in Figure 8.

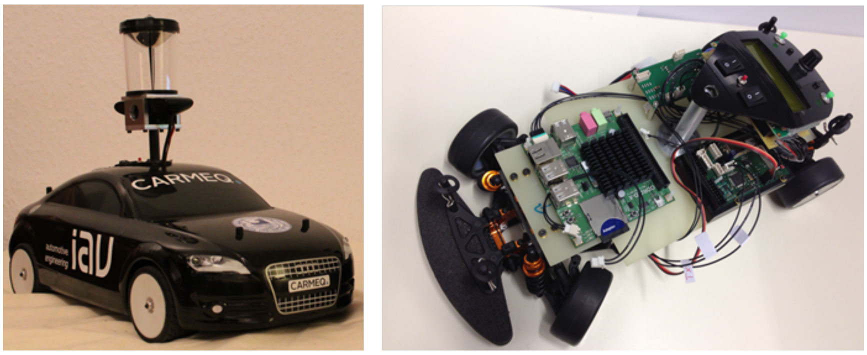

Figure 8: Team Berlin United’s competition vehicle

3.1.1 Observation

In order to observe the environment the vehicle was equipped with an omnidirectional mirror that a camera was looking through. This made it possible to have a 360 degree image view of the environment. The weakness of using omnidirectional vision is that the resolution of the image drops rapidly when the distance increases. To solve this problem a front facing camera is planned to be mounted. The vehicle was also equiped with a 9 degrees of freedom (DOF) inertial measurement unit (IMU). The IMU was used as odometer which helped when calculating the trajectory of the vehicle. No form of distance sensors was used, thus it solely relied on the camera to detect objects and to fulfill the Carolo Cup requirements.

3.1.2 Orientation

When the vehicle competed in Carolo Cup in 2012 the track was unknown before the competition so no pre-defined map was programmed in the vehicle. The purpose of the competition is not to navigate so that was not a problem. In the master thesis the focus lied on local localization while global localization was ignored. Local localization means that the initial position of the vehicle is known beforehand and the problem is thus only to track the vehicles movement. In global localization however, the initial position is unknown which creates more uncertainty about the pose and results in more computations. The data from the IMU was collected to give a first estimate of the vehicle's position which then was corrected by comparing the camera image with the pre-defined map. Unlike the competition a pre-defined map was used for the master thesis. It was used to compare the data from the sensors with the map in order to localize the vehicle. In theory the data from the image and the map should match but in reality they were slightly misaligned. Two solutions were tested to solve this problem: force field pose correction and particle filter. Both proved successful, each with their respective pros and cons. The lanes were detected by using edge detection on the image from the camera. Since the image from the omnidirectional mirror has a screwed perspective of the environment, image transformation was needed to represent the lanes as they are represented in the map.

3.1.3 Path planning and decision-making

If a pre-defined map is used and the vehicle is capable of accurately localizing itself in the map the vehicle can be able to drive to a certain point in the map. By calculating the path to the final destination, a list of points along the path is created. The vehicle will approach each point in the right order and when the vehicle is close enough to the point it can be dropped and the car will proceed to the next one. To solve junctions they can be mapped as point of interest where the vehicle will behave accordingly to the rules set for the vehicle. Since the focus is on lane localization nothing about decision making during driving is mentioned in the master thesis. There is also no report from when Team Berlin United competed in 2012 so how the vehicle makes decisions is therefore unknown.

3.2 Meili

In 2013 a team from Chalmers University of Gothenburg participated in Carolo Cup with the miniature vehicle Meili.[8] Meili was equipped with multiple sensors of different types, a microcontroller and a single-board computer amongst other hardware. The team took first place in the Carolo Cup Junior Edition 2013.[9]

3.2.1 Observation

The Meili used multiple ultrasonic- and infrared-based distance sensors and a monocular camera. The development team came to the conclusion that depth camera sensors or laser scanners were too expensive compared to the chosen sensors. The chosen distance sensors have a simple technical interface so it was easy for them to integrate and maintain them. The distance sensors were used to detect objects around the vehicle while the camera was used to detect the lanes.

The distance sensors were connected to a STM32F4 Discovery microcontroller which was connected to a PandaBoard ES which is a low-cost single-board computer. The camera was connected to the computer. All the software is located on the computer which handles image capturing, feature detection, lane following etc.

3.2.2 Orientation

The vehicle is equipped with an IMU to determine the vehicle's heading. The rear wheels were equipped with wheel encoders to measure the distance driven by the vehicle. Since the exact track for the competition is unknown for the contestants in Carolo Cup, no predefined map was available for the vehicle to use. No form of SLAM (Simultaneous Localization And Mapping) or GPS was used in the vehicle. The vehicle only knew where it was heading and what was around its current position.

3.2.3 Path planning and decision-making

The decision making model used for the Meili is a state machine which covers all the requirements for the competition. This is used to make sure that they know exactly what the vehicle will do in the scenarios that are possible when participating in Carolo Cup. To test their algorithms they used a virtual test environment which proved to be very helpful in speeding up the development process. By using a test environment it was possible to inject faulty or noisy data to systematically test the system behaviour.

3.3 Gulliver

Gulliver was a master thesis project written by Benjamin Vedder at Chalmers University of Gothenburg in November 2012.[10] The goal of the project was to make 1:8 scale miniature vehicles drive along a route and be able to change lanes, and to create a graphical interface to communicate with the vehicles. The vehicle can be seen in Figure 9. All goals were achieved at the end of the project.

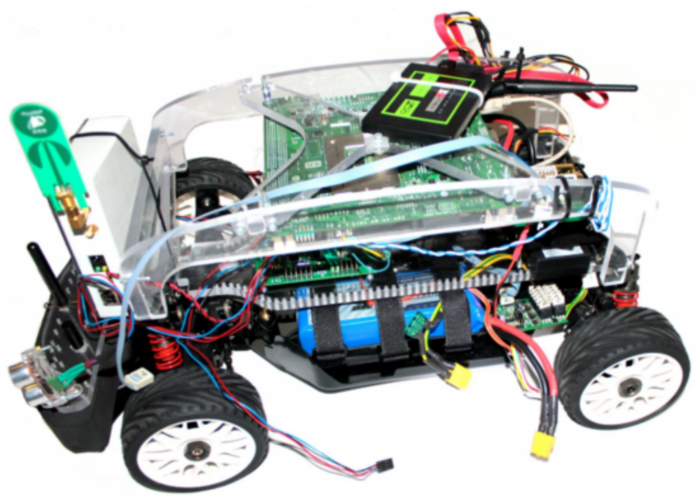

Figure 9: The Gulliver vehicle

3.3.1 Observation

Ultrasonic sensors and infrared distance sensors were used to observe the surroundings. They were chosen because Vedder believed they complemented each other and would give a similar function to what a 360 degrees laser scanner would give. The sensors task were to detect obstacles close to the vehicle.

3.3.2 Orientation

In order for the vehicles to orient themselves, a localization system was used. The localization system was developed in parallel to the Gulliver project. The vehicles were equipped with a broad band radio module, a ranging and communication module (RCM). The RCM is able to measure the distance to another RCM by measuring the time of flight (the time it takes for the signal to reach the other module). Three fixed anchor points with one RCM each were used to be able to triangulate the signal to the vehicle in order to determine the position of the vehicle. The magnetic field sensor was used as a compass to know the direction of the vehicle. The accelerometer and the gyroscope was used to compensate for rotations and sudden accelerations. After each RCM measurement the data was sent through a Kalman-filter to update the position.

3.3.3 Path planning and decision-making

The main approach of navigation for the vehicle is to use the pre-defined map with well placed points on each lane. To navigate the pre-defined route, the current position of the vehicle must be known. The steering angle was calculated by making sure that the vehicle would follow an arc through the next point on the route. The lane changing was done by choosing to follow the closest point in the other lane. The vehicle also featured an adaptive cruise control which made sure that the vehicle adjusted its speed to match the vehicle in front of it. It used the distance sensors in front of the vehicle to achieve this.

No form of advanced path planning or decision making algorithms were used, since the vehicle solely relies on the predefined route and on the cruise control in order not to crash in to objects.

3.4 MIG

MIG, short for "MadeInGermany", was created at the Freie Universität of Berlin. MadeInGermany is the first car licensed for autonomous driving on the streets and highways of the German state of Berlin. This project has, with the use of real-time algorithm, enabled an autonomous car to comfortably follow other cars at various speeds while keeping a safe distance. This work was published November 2013 and is a fitting example of an important aspect of self driving cars, namely sensor fusion.[11]

Figure 10: Autonomous Car, MadeInGermany.

3.4.1 Observation

The experimental car has in total six ibeo Lux LIDAR scanners with an horizontal field of view of 110 degrees. Only the three forward pointing LIDARs are used for the car following algorithm. A TRW radar sensor with a horizontal field of view of 12 degrees also points straight forward. Both types of sensors have a similar range of 200 m (depending on the material and inclination angle of the reflecting surface).

Autonomous cars are naturally complex systems, meaning having many sensors is a necessities for gathering enough data from the environment in order to fulfill the critical functions normally associated with self driving. Keeping the number of sensors to a minimum have both positive as well as negative effects. The gains for having fewer sensors is lowering the complexity of the system but reducing the redundancy will lower the safety, what happens if one sensor fails? In the case of MIG there is both two LIDARs and a radar that can see straight forward. By having both types of sensors one can take advantage of the different properties of the sensors. The radar gives a much better measurement of velocities of objects than that of the LIDAR but both readings are used. However, the system trusts the radar more and will weigh measurements coming from the radar as more accurate.

A black and white video camera behind the rear view mirror is used to detect the white lane strips and center the car on its lane. Two color cameras are used to identify traffic lights and their state. The car also uses an IMU to sense orientation and position estimation.

3.4.2 Orientation

To process all the data from the six LIDARs and the radar in the front, MIG uses a centralized fusion architecture which is separated into sensor layer and fusion layer, see figure 11.

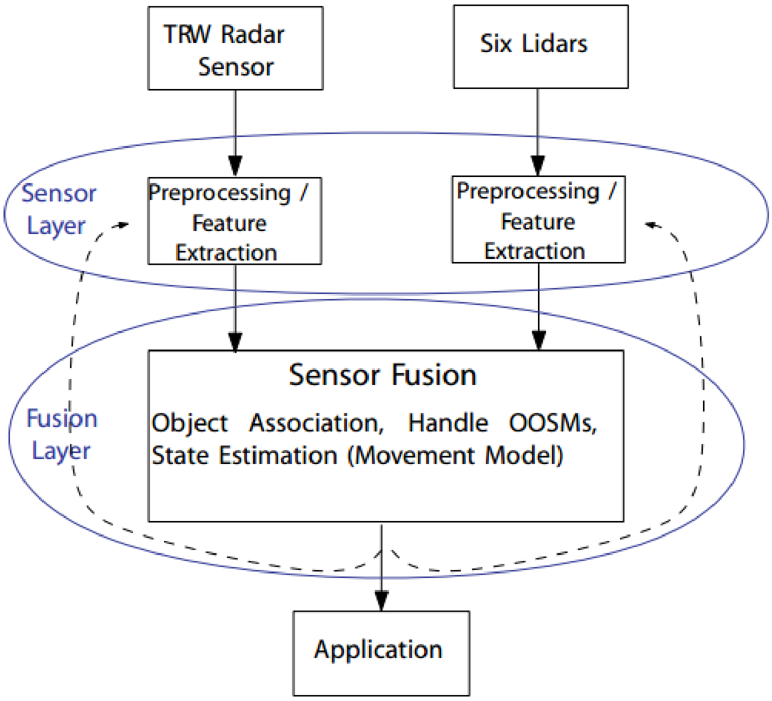

Figure 11: The Fusion Architecture is separated into Sensor Layer and Fusion Layer.

A big problem when it comes to having multiple sensors is the synchronization. The sensor information from all sensors around the car does not necessarily have to be sampled at the same time. Partly because the use of many different kinds of sensors, but also because processing time for the raw data is sensor dependent. In MIG there is no hardware synchronization, which causes a problem when applying orientation algorithms. The fusion layer must therefore cope with so called OOSM (Out-Of-Sequence Measurement). To solve this problem MIG uses buffering and measurement reprocessing.[12]

The information from all the sensors are processed individually for feature extraction. With many of the sensors seeing the same objects merging these are not easy. All this happens in the sensor layer. The fusion layer is what keeps track of and merges objects which are found in different sensors in the sensor layer. These objects have many different attributes. To give two examples, they can be either OFF-ROAD, NEAR-ROAD or ON-ROAD and they can also be static or dynamic depending on whether they are moving objects or not. By doing this association all trajectories being evaluated only need to be collision checked with obstacles near the road or on it. This saves computation time.

3.4.3 Path planning and decision-making

By this step the system should have all the information to choose a suitable trajectory. A behavioral module is in charge of generating a set of trajectories along the road network given a specific mission destination. These are then individually scored and the planner selects the trajectory with the best score. If obstacles are still on the chosen trajectory these are updated to be ON-TRAJECTORY instead. Moreover, the nearest obstacle with respect to the car is specifically marked as FIRST-ONTRAJECTORY. Now if for example a static obstacle is on the trajectory, the car can calculate how and when it should start braking and act accordingly. However if a better trajectory opens up the car will change trajectory using the same scoring system as when evaluating possible trajectories.

3.5 Junior

Junior is an autonomous vehicle made by the Stanford University. [13] It participated in DARPA in the year 2007 and placed second. The vehicle can be seen in Figure 12.

Figure 12: The Junior vehicle

3.5.1 Observation

To be able to detect obstacles and other vehicles a Velodyne LIDAR was mounted on the roof along with two LIDARs at the rear of the vehicle and two at the front bumper. The Velodyne LIDAR has a range of up to 60 m and can scan 360 degrees horizontally and 30 degrees vertically. Additionally the vehicle was provided with 5 long range radars at the front grill to get better information about the surrounding vehicles. However no camera was used. To navigate, the Junior has an Applanix system which includes a 6 DOF IMU and wheel encoders via a distance measurement unit (DMI). Junior also had two Intel quad core servers with a Linux operative system that communicated over ethernet.

3.5.2 Orientation

To simplify the task the vehicles were provided with a digital pre-defined map of the environment in GPS coordinates. But since GPS isn't accurate enough to estimate the vehicle position, Junior uses all of its sensors to get more reliable measurements. The GPS and odometry gives an initial position which is then updated by an unscented kalman filter that uses the different sensor measurements and compare them to the given map.

Since the car can't see everything at one given time it's important to integrate multiple measurements to be able to map the static environment.

Junior therefore stores all the sensor measurements into a local map and updates it according to a standard Bayesian rule in order to get rid of the false detections that only show up in a couple of measurements.

The detection of moving objects was made by scanning the environment in 2D. The scans are performed recursively and each scan is compared to the last one in order to detect if and where a change has been made. When a moving object is found a particle filter algorithm is used to estimate the size, location and velocity of the object. A set of particles is initialized. Each particle is given a different weight and implements a different hypothesis surrounding the location, velocity and size. Information about the moving object is received and the weights of the particles are updated accordingly. Next time a new set of samples is produced that will have a higher concentration at the place where the previous particles had the highest weight. The filter can then reliably track the object after 3 sightings.

3.5.3 Path planning and decision-making

The path planner uses the information from the perception module that tells where Junior is located and what's around it. The road to the goal is divided into several checkpoints. To be able to navigate through the environment Junior starts with planning the path to the next checkpoint. It plans the path by computing the minimal cumulative cost from every location in the graph to the desired checkpoint. That's useful if the vehicle should somehow get out of position. The cost function takes both the time and risks into account and is recursively updated as the vehicle moves forward.

The behavior of the vehicle is a finite state machine (FSM) that consists of 13 states where the top level states are normal driving modes such as lane keeping, intersection handling and parking. Based on the information it gets from the path planner the behavior transitions into different states.

4 Comparison

In this section the projects presented in the State of the art section are compared with each other.

4.1 Observation

Both the Meili and the Berlin United vehicle uses a camera to detect lanes. Both of them are using a monocular camera, however the mirror used by the Berlin projects gives it the effect of an omnidirectional camera. The omnidirectional camera has a bigger view than the monocular one but a worse resolution at longer distances. To be able to detect static objects the Meili and Gulliver vehicles use ultrasonic and infrared sensors instead of using the camera as in the Berlin project. The full-sized vehicles on the other hand uses LIDARs and radars for object detection.

Something that sets Junior aside from the other projects is that Junior doesn't use a camera. For the smaller vehicles the camera is the most important sensor. Junior instead relies on the information of the LIDARs and radars and uses its pre-defined map to be able to accurately know where the roads are. Since it's an urban environment there are a lot of landmarks to perceive which gives Junior the location of the road. The road would thus be harder on a track with fewer features to perceive.

4.2 Orientation

All vehicles used some sort of IMU to determine where the vehicle was going. The Meili, Junior and Gulliver additionally used wheel encoders to determine how far it had gone. Both Meili and Berlin United make all the computations on board while the Gulliver transfer some computations to a host computer. Junior also makes the computations on board but it stores a big computer in the car.

Most of the projects have in common that they have some sort of filter to get a more precise location of the vehicle. The Berlin United vehicle have an IMU that gives a good first estimate of the position, which is then updated by a particle filter based on the information from the omnidirectional camera. The Gulliver vehicle only has a Kalman filter that updates the position while Junior is using an unscented Kalman filter, which unlike Gullivers Kalman filter works for nonlinear systems. The Meili project on the other hand didn't use any filter which means that they will not get a precise location. It was however enough to fulfill the requirements of the Carolo cup. A difference between the Meili and Berlin projects is that Berlin tested the vehicle directly on a smaller Carolo cup like track while Meili used a virtual testing environment. For the projects where localization of the vehicle was important some pre-defined map was made.

4.3 Path planning and decision-making

All of the researched projects uses some kind of state machine to determine how the vehicle should behave. The full-sized vehicles have more advanced decision-making and path planning systems than the miniature ones. Since the safety issue needs to be considered, the decision making system of a full-sized vehicle will come up with multiple solutions to a problem that the vehicle need to solve.

5 Discussion

The State of the art shows that there are several ways to make a vehicle self-driving. Which way to choose depends (among other things) on the size of the vehicle, the budget of the project and the specific environment the vehicle will be used in. As an example, the safety issue is more important in a full-sized vehicle than in a miniature one. Because of this they will demand more and better sensors. Full-sized vehicles also drive in a richer environment with more objects to perceive compared to the miniature ones. It is thus more justified for them to use an expensive LIDAR than it is for the miniature vehicles. The LIDAR is expensive but more accurate and less sensitive to changes in the environment. This is reflected in the choices made by the projects researched in this report. For orientation, the full-sized vehicles use several LIDARs while the miniature-vehicles use techniques which are cheaper and more suited to controlled environments.

Something that differs between the three miniature vehicles is where they do their computations. The advantage with having the computations done on the vehicle is that it makes the delays shorter so quicker decisions can be made. The downside is that it reduces the complexity of the algorithms that can be performed. The full-sized vehicles do not have this dilemma since they have the capacity of storing more advanced computers onboard.

There are also different ways of testing the vehicle. A successful approach seems to be to use a virtual test environment. This can speed up the testing process compared to only doing tests with the actual hardware. A possible downside is that developing the virtual test environment could be very time consuming. Thus, while it could enable more thorough testing, it would not necessarily save any time for the project.

When it comes to orientation, some projects use pre-defined maps while others simply orient themselves as they go along, without knowing their exact location. A third way, not used by any of the researched projects, is to use SLAM (Simultaneous Localization And Mapping). The idea of SLAM is to navigate through an unknown environment while simultaneously mapping it for future reference (see Appendix E for more details). This technique can be useful in real life situations where a pre-programmed map might not always be available or might quickly be outdated. To make a pre-defined map can also be cumbersome and time consuming.

5.1 Future for the ELAD project

For our project we will use the hardware which we already possess but if concluded that a certain component lacks the capabilities necessary for us to meet the requirements, we will change that component.

In order for us to easily and quickly change code we have also decided to move the heavy computation to a host computer. Not knowing if the motherboard's CPU power would be enough also played a role in the decision. All sensor information will be sent to the host computer via WIFI communication before being analyzed. The host computer will then send back a set of instructions for the motor and steering. A delay between the acquired sensor data and response to the drive system will occur, we however reasoned that the reaction time is dependent on the driving speed and therefore decided to focus on driving accurately rather than quickly.

We will use a color camera primarily for lane detection and if time allows it add object recognition. The camera view will first be transformed to bird-eye-view (top view) and the lane detection will be done in two parallel algorithms. The first will be a more deterministic, well known and well researched algorithm, exactly which is yet to be decided. The second will be a neural network with the ability to give a confidence level of its output, meaning it will tell us how sure it is about the lane recognition. The final lane observation will be weighed based from the confidence level of the neural network and is very similar to fuzzy logic. If the confidence is high the neural network answer will be weighed heavier and if low enough ignored.

Finding objects will be done from the LIDAR sensor data. Conventional algorithms will be used such as particle or Kalman filters. We will also try to implement SLAM in order to locally localize the vehicle.

For state recognition we will use the same method as the camera namely by having one deterministic and one neural network based algorithm. This time the choice of state is discrete and since we cannot use both of the outputs from the algorithms we will only use the neural network solution if the confidence level is high enough.

When the state is determined, like for example following a lane, a suitable trajectory needs to be computed. We have yet to decide how this will be done but in the case of lane following selecting a few points, forming a line in the middle of the current lane, will be intuitive. How to handle intersections is not yet decided upon.

During the development it helps to visualize and display what the vehicle is currently seeing, what it thinks it's seeing and what it will do next (state and trajectory). Having all this information on the host computer during the testing will also help further implementations later made by ÅF.

For quick and easy testing during the development, having a virtual world will help and also be a good tool for ÅF for further development of the vehicle. With us planning to use AI algorithms, teaching these will be challenging if it needs to be done in the real world. Since the virtual world already knows exactly what its world looks like teaching the AI will be easier.

Lastly we will focus a great deal on documenting and structuring our final product and the development of it. Having well documented code and informative text will help to speed up the learning for people new in the field. During the project we will learn of new and maybe better ways of doing tasks and therefore the current approach will most likely change.

References

1. Bellis, M. The History of the Automobile. Retrieved 2016-05-15 from http://inventors.about.com/library/weekly/aaautomobilenameorigin.htm

2. Weber, M. (2014). Where to? A History of Autonomous Vehicles. Retrieved 2016-04-14 from http://www.computerhistory.org/atchm/where-to-a-history-of-autonomous-vehicles/

3. Cheng, H. (2011). Autonomous Intelligent Vehicles, Theory, Algorithms and Implementation. Springer, London

4. Carolo cup. Retrieved 2016-05-16 from https://wiki.ifr.ing.tu-bs.de/carolocup/

5. Torc Robotics. DARPA Urban Challenge. Retrieved 2016-04-25 from http://www.torcrobotics.com/darpa-urban-challenge

6. Hooper, J. (2004). Leonardo's car brought to life. Retrieved 2016-04-11 from http://www.theguardian.com/world/2004/apr/24/italy.arts

7. Maischak, L. (2014). Lane Localization for Autonomous Model Cars. [online] Berlin: Freie Universität. Available at: https://www.mi.fu-berlin.de/inf/groups/ag-ki/publications/B-14-03/B-14-03.pdf [Accessed 19 May 2016].

8. Berger, C. (2014). From a Competition for Self-Driving Miniature Cars to a Standardized Experimental Platform: Concept, Models, Architecture, and Evaluation. Journal of Software Engineering for Robotics, [online] 5(1), pp.63-79. Available at: http://arxiv.org/abs/1406.7768 [Accessed 12 May 2016].

9. Chalmers.se. (2016). Seger i tävling för modellbilar. [online] Available at: http://www.chalmers.se/sv/nyheter/Sidor/Seger-i-tavling-for-modellbilar.aspx [Accessed 12 May 2016].

10. Vedder, B. (2012). Gulliver: Design and Implementation of a Miniature Vehicular System. [online] Gothenburg: Chalmers University of Technology. Available at: http://publications.lib.chalmers.se/records/fulltext/173682/173682.p [Accessed 16 May 2016].

11. Daniel Gohring, Miao Wang, Michael Schnurmacher, Tinosch Ganjineh. (2011). Radar/Lidar Sensor Fusion for Car-Following on Highways.

12. N. Kaempchen and K. Dietmayer. (2003). Data synchronization strategies for multi-sensor fusion.

13. Montemerlo, M et. al (2007). Junior: The Stanford Entry in the Urban Challenge, Stanford University. Available at: http://robots.stanford.edu/papers/junior08.pdf [Accessed 17 May 2016].

Appendix A

LIDAR

How LIDAR works

A LIDAR, Light Detection and Ranging, uses laser to measure the distance to the target it illuminates. To calculate the distance to the target a simple formula is used. Since laser is light and the speed of light is known the equation for the distance can be seen in Equation 1 [1].

$$Distance = \frac{(Speed\ of\ Light * Time\ of\ Flight)}{2} \tag{1}$$

By rotating the LIDAR at a certain speed it is possible together with the calculated distance and a computer to visualize the environment of the LIDAR. The chosen wavelength of the LIDAR often ranges outside the visual spectrum of the human eye, 400-700 nanometer [4], to make sure the eye won't be able to focus on the laser. If however a wavelength within the spectrum is used the power is limited to make sure the laser won't hurt the eye. The advantage with a lower wavelength is improved accuracy so the choice of wavelength highly depends on the usage area of the LIDAR.

LIDAR and autonomous vehicles

Many different types of sensors are used and combined to achieve some form of autonomous vehicle. These sensors are for example RADAR, LIDAR, cameras and ultra-sound sensors. The LIDAR is used differently depending on the company developing the vehicle. Google's self-driving car is using a LIDAR with 64-beam laser from Velodyne which allows the vehicle to generate a detailed 3D map of its surroundings. The LIDAR is mounted on the top of the vehicle and rotates. Ford's autonomous research vehicles are also using LIDAR technology to map the vehicles environment.One iteration of their research vehicles used 4 LIDAR's on the roof in different angles [2]. Other car manufacturers, for example Volvo [3], are using LIDARs as well but not to map the whole surroundings but only in the front of the car.

Velodyne LiDAR

Velodyne is the leading developer and manufacturer of LIDARs. Companies like Google, Ford [5] and Volvo [6] are using LIDARs from Velodyne for their research in autonomous vehicles.

The LIDAR Google, among other, are using is the HDL-64E LIDAR from

Figure 13: Velodyne HDL-64E

Velodyne and can be seen in figure 13. The HDL-64E has a price tag at around $75,000 (2016-04-14) which makes it very expensive to equip on a car. This LIDAR can be seen as state of the art due to its advanced technology and is used by some of the leading research teams for autonomous vehicles who uses LIDAR. Some specification for the HDL-64E are the following:

- Distance range 120 m
- 5-20 Hz (2.2 million points per second)
- Angular resolution = 0.08°
- 64 laser beams
- Laser wavelength: 905 nanometer

RoboPeak RPLIDAR

The LIDAR given to this project by ÅF can be seen in Figure 14. The price tag of the RPLIDAR is around $400 (20116-04-14) and can be seen as a budget choice when is comes to LIDARs. The RPLIDAR has the following specifications:

- Distance range 0.2-6 m
- 5.5 Hz (2000 sample/sec)
- Distance resolution $<$ 0.5 mm
- Angular resolution =$<$ 1°

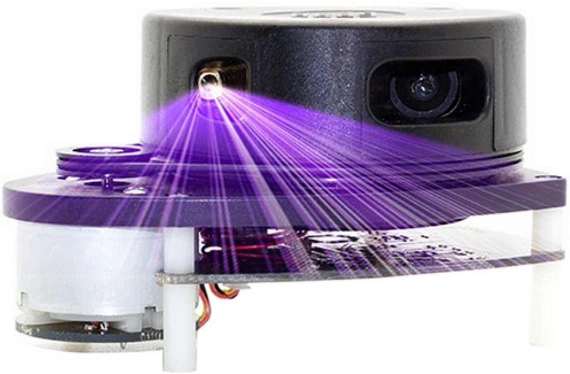

Figure 14: RoboPeak RPLIDAR

- Single laser beam
- Laser wavelength: 785 nanometer
- Laser power: 3 milliwatt

Due to the single laser beam of the LIDAR only a 2D view of the environment can be achieved.

References

[1] *http://www.lidar-uk.com/how-lidar-works/* 2016.

[2] *http://www.pcworld.com/article/3020407/ces/how-fords-autonomous-test-vehicles-make-3d-lidar-maps-of-the-world-around-them.html* 2016.

[3] *http://www.volvocars.com/intl/about/our-innovation-brands/intellisafe/intellisafe-autopilot/drive-me/real-life* 2016.

[4] *George L. Heritage, Andrew R. G. Large, Laser Scanning for the Environmental Sciences* 2009.

[5] *https://media.ford.com/content/fordmedia/fna/us/en/news/2016/01/05/ford-tripling-autonomous-vehicle-development-fleet–accelerating.html* 2016.

[6] *http://www.volvocars.com/intl/about/our-innovation-brands/intellisafe/intellisafe-autopilot/news/seeking-the-truth-with-autonomous-cars* 2016.

Appendix B

Neural networks and artificial neural networks

The fastest computer in the world is the human brain. If you walk down a crowded street you can recognize a familiar face in approximately 100–200 ms. For a computer tasks of much lesser complexity still takes a great deal longer. So how does a human brain do it? Our brains are highly complex, nonlinear and use parallel computation. They have the capability to organize their structural constituents, known as neurons, so as to perform certain computations (e.g., pattern recognition, perception, and motor control). [1] It is estimated that there are approximately 10 billion neurons in the human cortex with 60 trillion synapses (connections) between them. [2]

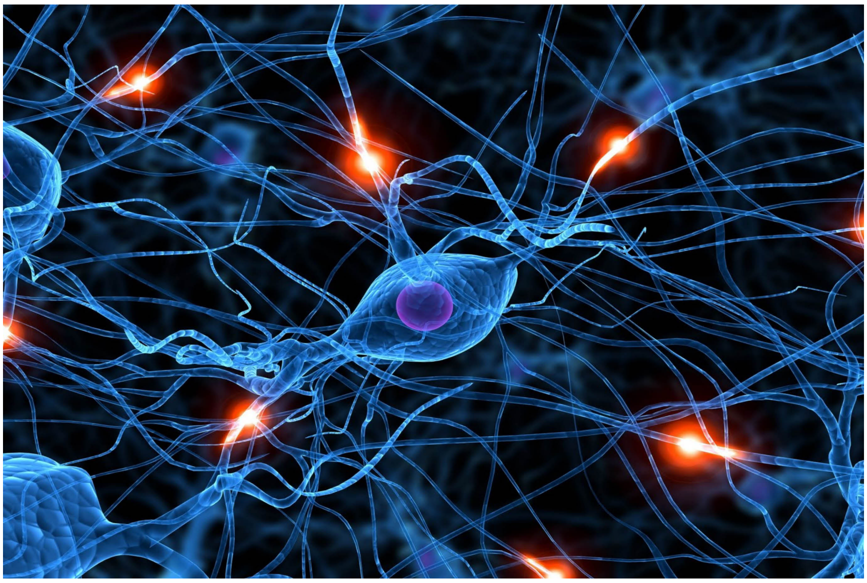

Figure 15: The brain consists of billions of neurons connected with each other. The figure shows a model of a network of neurons connected with each other.

A neuron gets input from other neurons via dendrites, processes the inputs and then either fire or not fire an output signal via the axon that connects to other neurons, figure 16. In this way a big network of neurons is made and depending on how the neurons connect to each other we learn different things.

Frank Rosenblatt was the first one to make an artificial neural network in 1958. The intention was to model how the human brain learned to recognize objects. Soon researchers realized that these models could be useful tools in their own right. By the late 1980s, many institutes were using Artificial neural networks for a variety of purposes.[3]

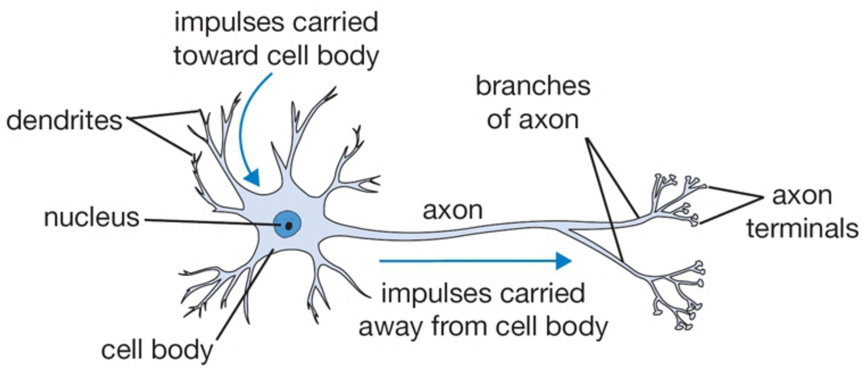

Figure 16: Model of neuron in human brain. A neuron gets input via dendrites and output via an axon. The dendrites and the axon connects to other neurons.

An artificial neuron is based on a real neuron and can be modeled as seen in figure 17. The neuron gets input, x_i, from a set of connecting links. Each link is characterized by a weight, w_{kj}, which the input is multiplied with. The k in the subscript of the weight w_{kj} stands for what neuron the link connects to and the j stands for which input signal x_i that is transferred. The input signals are then summed and a bias value b_k is added. Thereafter the result from this is inputted to an activation function and at last the output y_k is recieved.[1]

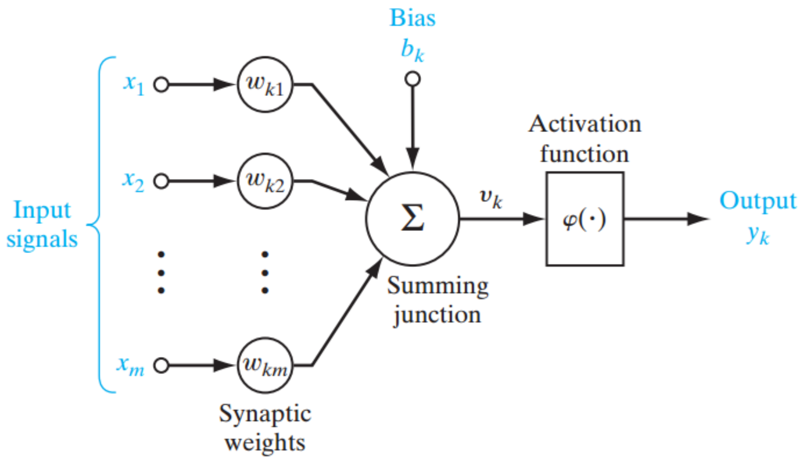

Figure 17: Model of artificial neuron. Input, x_i, are multiplied with weight values, w_{kj}, specific for each connecting link. The input signals are summed and a bias value b_k is added. Thereafter the result from this is inputted to an activation function and at last the output y_k is received.

Mathematically a neuron can be described as:

$$v_k = \sum x_i \cdot w_{kj} + b_k \tag{2}$$

$$y_k = \varphi(v_k) \tag{3}$$

An artificial neural network, ANN, consists of many neurons connected to each other in different layers, figure 18. The first layer is called input layer and the last layer is called output layer. The layers in between are called hidden layers. Number of neurons and number of hidden layers are hyper-parameters that needs to be tuned by testing. The idea is that the strength of the output can be controlled with the weights, w_{kj}, and that the values of the weights can be trained so that the system act in a desired way.

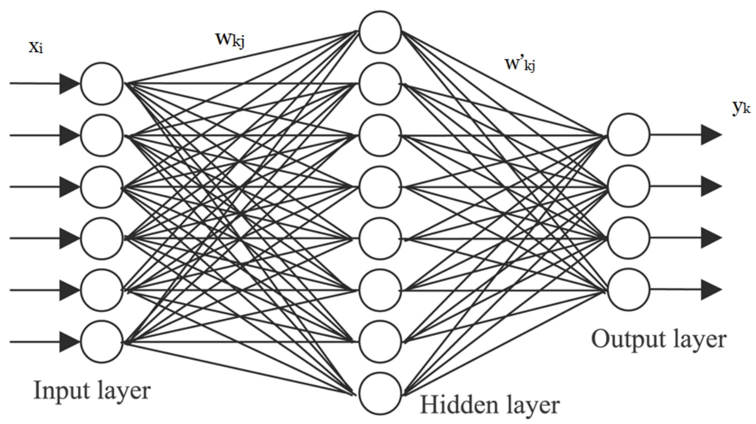

Figure 18: A network of several neurons, where each neuron is modeled as in figure 17. It might seem as if a neuron in the hidden layer outputs several different outputs, that is not the case. Each neuron yields one output and this output is then sent to several neurons, therefore all the lines.

So how is the values of the weights learned? There are several approaches for learning applicable in different situations. Learning strategies can be divided into "Learning with teacher" and "Learning without a teacher". The former one is also called supervised learning. In this type of learning you train the network on a big set of labeled data by comparing the output from the ANN with the labeled desired output from the data. Training can be done with back-propagation, gradient descent, least mean square error etc. The latter category can be sub-categorized into unsupervised learning and recurrent learning.[1]

Based on learning method plus activation function you get different kinds of neural networks. Example MLP - multi layer perceptron, RBFN - radial basis function network, recurrent network etc etc.

References

1. Haykin, S. (2009). Neural Networks and Learning Machines, 3d edition. New Jersey, Pearson Education.

2. Shepherd, GM. The Synaptic Organization of the Brain, 5th edition.

3. Kay, A. (2001). Artificial Neural Networks. Retrieved 2016-4-28 from http://www.computerworld.com/article/2591759/app-development/artificial-neural-networks.html

Appendix C

Driverless cars - fun facts from da Vinci to DARPA

Already in 1478 Leonardo da Vinci designed a cart that would have been somewhat autonomous. It was self propelled and featured programmable steering and a brake that could be released from a distance by an operator with a rope. Figure 19 shows a sketch of the cart.

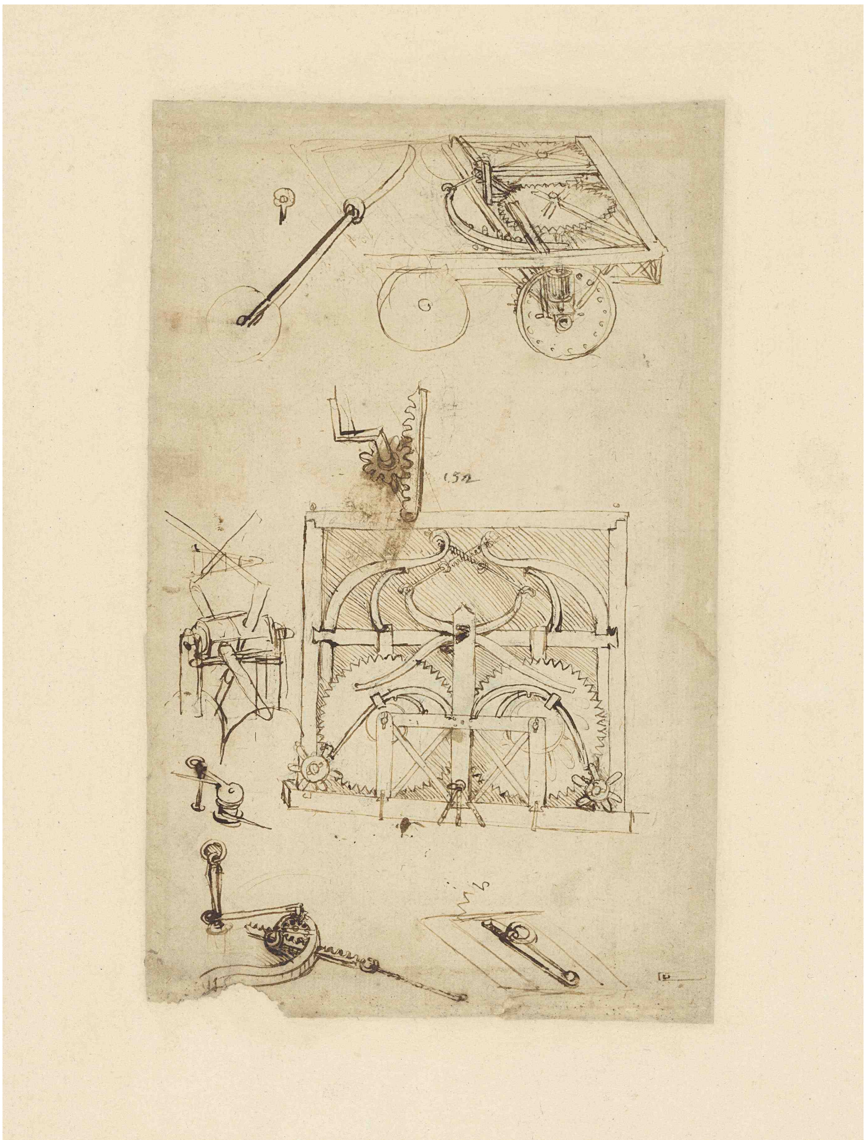

Figure 19: The worlds first robot?

Da Vinci never built the cart but in 2004 Carlo Pedretti, director of the

Armand Hammer Center for Leonardo Studies in Los Angeles, managed to build it from the original sketch. And it worked! [1].

A video of the working model can be found here: https://www.youtube.com/watch?v=a2qeZrejZp0

In 1925 people in New York could see the first driver less car cruise down the streets. The car was controlled by radio from a second car driving close behind, figure 20. Unfortunately the trip ended with a small accident [4].

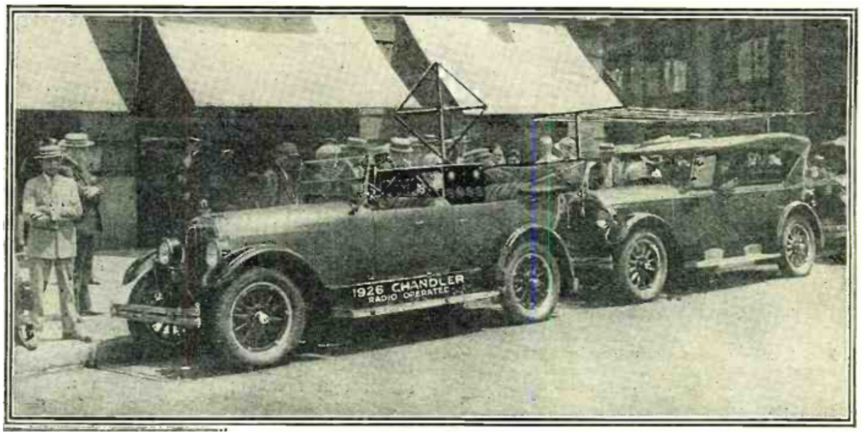

Figure 20: New York 1925. Driver less car to the left in the figure with the controlling car behind it. The car was controlled by radio.

Autonomous cars have figured in science fiction since the 1900s. 1935 David H. Keller wrote in the American science fiction Magazine Wonder stories:

> “Old people began to cross the continent in their own cars. Young people found the driverless car admirable for petting. The blind for the first time were safe. Parents found they could more safely send their children to school in the new car than in the old cars with a chauffeur". [2].

Since cars operate in a very complex environment with fragile human around them autonomy has been implemented on other types of vehicles earlier than on cars, for example on airplanes, boats and mars rovers. Early self-driving plans for cars were more focused on special freeways for guiding cars safely along them than autonomous robot cars. The American designer and futurist Norman Bel Geddes showed this vision with smart autobahns on the futuruma exhibit at 1939 World’s fair. His idea was to incorporate electronic speed and collision control systems common to rail roads on highways. The driver was supposed to drive to the Autobahn and when arriving engage an automatic system that would drive instead of the driver until the exit of the freeway. Figure 21.

Figure 21: Futurama model of a smart highway, the cars where supposed to be driving in separated lanes and distance between cars and speed was to be controlled with radio.

In the 1950s General Motors (GM) together with the Radio Cooperation of America (RCA) started to test smart highways in line with what Norman had suggested at the Futurama exhibit. They developed automated highway prototypes with radio control for speed and steering. Magnets in the car tracked a steel cable embedded in the road and control towers managed overall traffic flow. See figure 22.

Figure 22: GM and RCA developed and tested smart highways in the 1950s. Magnets in the car tracked a steel cable in the road and control towers managed the traffic regarding speed and steering using radio control.

At the 1950s America's Electric Light and Power Companies had an advertisement about the "Driverless Car of the Future" highlighting some positive effects of self driving cars, figure 23. However the massive consensus needed to build public infrastructure like this never happened.

In the 1960s Stanford had an Artificial Intelligence Laboratory cart that managed to navigate in unknown environments using new AI-techniques. And in the 70s semi-autonomous space probes where in use. [2] But it wasn't until 1977 that the first autonomous car was built. The Japanese Tsukuba Mechanical Engineering lab managed to build a car that could follow a road at speed up to 30 km/h. The car found it's way by tracking white street markers with machine vision. The car was equipped with two cameras that used analog computer technology for signal processing.

Next big thing in the history of autonomous cars was the German aerospace engineer Ernst Dickmanns' (the pioneer of autonomous cars) series of projects in the 1980s. One of the big projects was the Mercedes van called VaMoRs. It had two cameras, eight 16-bit Intel microprocessors and a host of other sensors and software. VaMoRs drove in more than 90 km/h for abut 20 kilometers. Some years later, in 1993, Dickmanns' Mercedes sedan, called VaMP,

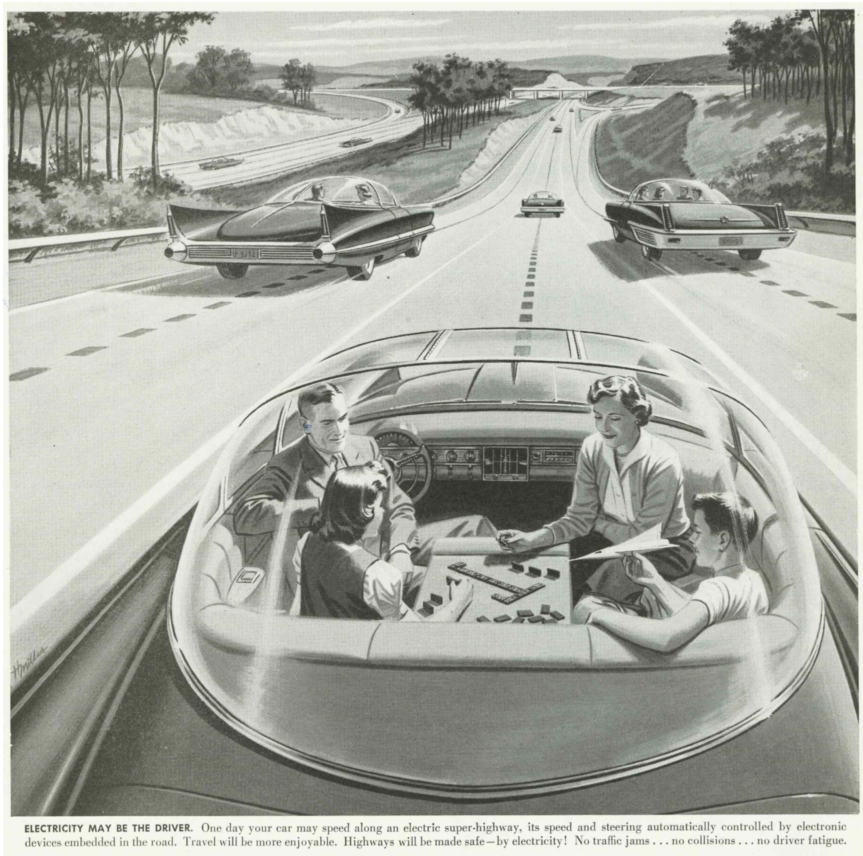

Figure 23: 1950's dreaming about the future, original figure text saying: " Electricity may be the driver. One day your car may speed along an electric super highway, it's speed and steering automatically controlled by electronic devices embedded in the road. Travel will be more enjoyable. Highway will be made safe - by electricity! No traffic jams...no collision...no driver fatigue."

could recognize road markings, know where it was positioned in the lane and detect other vehicles. In a test drive near Paris the car drove at up at 130 km/h in simulated traffic and could even judge whether it was safe to change lanes or not. The VaMP was part of the massive Eureka PROMETHEUS project (an EU financed project that aimed to develop driverless cars between the years 1987-1995). [2] [5]

The year after Dickmann's team piloted a Mercedes S-Class from Munich to Odense, Denmark, a trip of more than 1,600 kilometers at a maximum speed of 180 km/h with, as Dickmanns notes, "about 95 percent of the distance ... traveled fully automatically."

The success of these vehicles redirected research away from cars guided

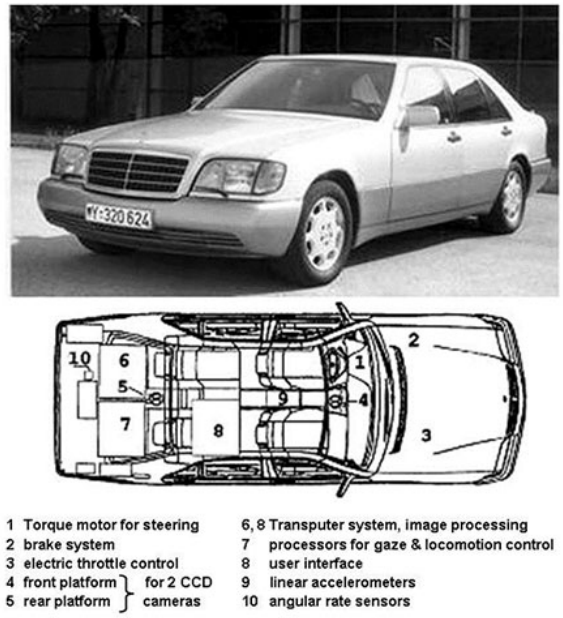

Figure 24: Dickmanns automated car VaMP managed to drive in simulated traffic using cameras to track lateral lines of the road and detect vehicles.

by inductive signals received from cables embedded in the roads and instead the research became more focused on vision-based systems for lateral guidance.[5]

In 2004, the U.S. Defense Advanced Research Projects Administration (DARPA) challenged dozens of teams to drive autonomously through the Mojave desert, the winner would get 1 million dollar. The hope with this contest was to speed up the development of autonomous cars in order to get one third of the military vehicles to drive themselves by 2015. The contest didn't go well and most cars crashed only after some km. The next year however went better and by 2007 the challenge had been extended to also include a mock city environment. Figure 26 and 25.

Figure 25: 11 vehicles took part in the DARPA challenge year 2007.[6]

Figure 26: The Darpa winner was Boss, a robotized 2007 Chevy Tahoe. It followed California driving laws as it navigated the course and operated in a safe and stable manner. In fact, all of the robots seemed to make good decisions, due to DARPA Director Tony Tether. That meant speed became the determining factor and Boss was the fastest of the competitors. Boss averaged about 23 kilometers per hour over approximately 89 kilometers.[7]

References

1. Hooper, J. (2004). Leonardo's car brought to life. Retrieved 2016-04-11 from http://www.theguardian.com/world/2004/apr/24/italy.arts

2. Weber, M. (2014). Where to? A History of Autonomous Vehicles. Retrieved 2016-04-14 from http://www.computerhistory.org/atchm/where-to-a-history-of-autonomous-vehicles/.

3. Keller, H, David. (May 1935). Wonder Stories.

4. Radio News for November, 1925. Retrieved 2016-04-14 from http://www.americanradiohistory.co
Radio-News/20s/Radio-News-1925-11-R.pdf. Page 592 and 656.

5. Vanderbilt, T. (2012). Retrieved 2016-04-14 from http://www.wired.com/2012/02/autonomous-vehicle-history/

6. Torc Robotics. DARPA Urban Challenge. Retrieved 2016-04-25 from http://www.torcrobotics.com/darpa-urban-challenge

7. Truck Trend. (2007). Auto News: Carnegie Mellon Tartan Racing Wins 2 Million dollar DARPA Urban Challenge. Retrieved 2016-04-25 from http://www.trucktrend.com/news/163-news071105-darpa-urban-challenge-winner/

Appendix D

Elad requirements

Functional Requirements

Prio 1:

1. Vehicle shall be capable of driving $5m$ while staying within two straight solid lines $30cm$ apart. See track number 1.

2. Vehicle shall be capable of driving $5m$ while staying within two S-formed solid lines $30cm$ apart. See track number 3.

3. Vehicle shall be able to detect a $20cm$ x $20cm$ x $20cm$ white stationary box at a distance of minimum $10cm$ and maximum $5m$ and at a maximum speed of $10km/h$.

4. Vehicle shall be able to stop with normal braking in front of a $20cm$ x $20cm$ x $20cm$ stationary box when driving on a straight road with solid lines.

    (a) After vehicle has stopped, the distance between vehicle and box shall be no farther than $20cm$ and no less than $10cm$.

        i. The measured distance between the box and vehicle shall be within a 5 percent error at the maximum speed of $10km/h$.
        ii. The vehicle shall start to brake at the required distance it would take to come to a complete stop with the current speed and with a deceleration of $0.7m/s^2$.

5. Vehicle shall use emergency braking if normal braking will not be enough to stop at a minimum distance of $10cm$ in front of a $20cm$ x $20cm$ x $20cm$ stationary box when driving on a straight road with solid lines.

    (a) Emergency braking shall use the maximum braking acceleration that the car is capable of.
    (b) When the distance to the object is enough for the normal braking to come to a complete stop within a minimum distance to obstacle of $10cm$ then switch back to normal braking mode.
    (c) The vehicle should indicate automatic emergency brake mode.

6. Vehicle shall at all times beware of its current speed.

Prio 2:

1. Vehicle shall be capable of driving 5 meters while staying within one straight solid lines to the right and one straight dashed line to the left 30 cm apart. Another solid line on the far left (meeting lane) should also be 30 cm apart from the dashed center line. See track number 2.

2. Vehicle shall be capable of driving 5 meters while staying within one solid S shaped line to the right and one dashed S shaped line to the left 30 cm apart. Another solid line on the far left (meeting lane) should also be 30 cm apart from the dashed center line. See track number 4.

3. When an stationary obstacle block the current driving lane, the vehicle shall slow down to $3km/h$ and if right lane exist and is clear it shall then proceed to drive by obstacle without crossing the white solid line. If this is not possible then vehicle shall come to a complete stop in front of obstacle.

4. When there is nothing within $3m$ in front of the vehicle, it should accelerate to the maximum speed of that current road type.

5. If vehicle lost track of lane, it shall be capable of finding the lane again.

   (a) The vehicle shall be capable of determining if it has lost track of the lane it was previously following if lane consists of either single lane road or dual lane road.
   (b) The vehicle should back up slowly to maximum of 1 meter while scanning the environment in order to find the lane again.
   (c) If the vehicle can not find the lane, it should stop at 1 m.
   (d) If the vehicle finds the lane it shall continue driving in the same direction as it did before.

6. When two or more vehicles are waiting at a intersection, the vehicle shall be capable of deciding of which vehicle goes first and wait for its turn in order to follow traffic rules.

   (a) The vehicle shall be capable of turning in a intersection both to the left and right.
   (b) The vehicle shall be capable of going straight forward in a intersection.
   (c) The vehicle shall be able to detect vehicles waiting at the intersection which has the same size as the vehicle in context. It shall know how many and where they are located.
   (d) The vehicle shall be able to determine while waiting in the intersection other vehicles states, such as arriving, waiting, driving left, driving right, driving straight.

Non-functional Requirements

Prio 1:

1. The vehicle shall have an error code signal capable of informing that an error has occurred and if so what type.

2. The vehicle shall have an warning code signal capable of informing that an warning has occurred and if so what type.

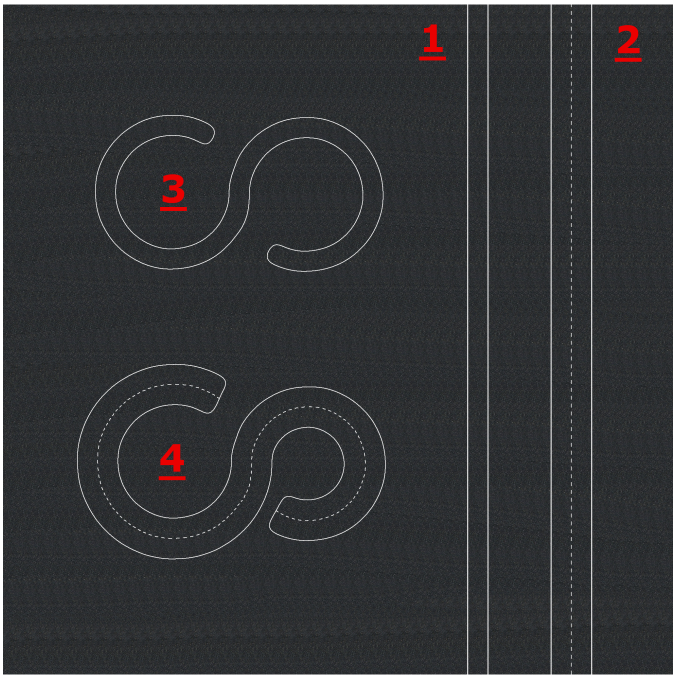

Figure 27: Tracks 1 to 4 that will be used

3. The system shall give a warning when the remaining running time of the vehicle is less than 5 minutes.

4. Vehicle shall be able to receive commands and send sensor data to a host computer after turning on the system.

5. If one of the Lidar, camera and WIFI components no longer can communicate with host computer, vehicle shall slow down and stop in 1 second and give a warning.

6. When the host computer for the vehicle sends an order to the vehicle, the response time shall be less than 0.2 second.

7. The vehicle shall be capable of working during 15 min normal use.

8. The vehicle should shut down if no further orders are received from the host computer after 5 min.

9. The platform should be documented so that a new user could pick up and continue development.

10. A model of the car in a virtual world shall give users the capability of trying out new algorithms without having to test on the actual vehicle directly.

    (a) The virtual world shall include tracks suitable for the performance requirements.

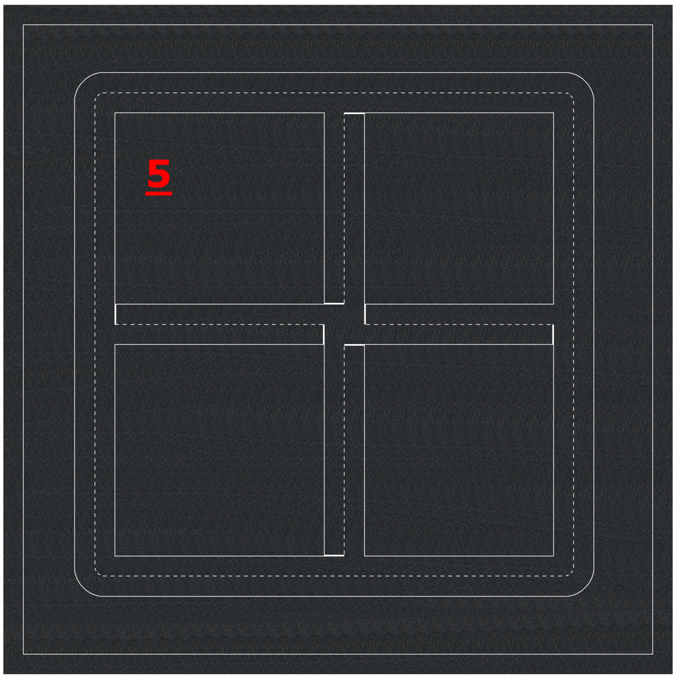

Figure 28: Track 5 that will be used

(b) The dynamics of the vehicle should behave similar to how they are in real life.

11. The vehicle should convey a ÅF feeling.

    (a) ÅF's logo should be on the vehicle.

12. The vehicle should looks professional.

    (a) No loose wires should be visible.
    (b) It should look robust.

13. The vehicle shall be built in such a way that adding simple sensors such as buttons should be easy.

14. A usable library of important functions shall be written which can be used as a foundation for further implementation.

    (a) Function such as sensor data acquirement and data processing shall be primarily focused on.

Prio 2:

1. Lidar data shall be saved continuously on the host computer. A SLAM algorithm shall then be used in order to play back what the robot saw.

2. The camera data shall be recorded and saved on the host computer in a automatic generated time stamped file and folder structure.

Appendix E

SLAM

The chicken and egg problem

In order to orientate yourself a map is needed but a fully drawn map is most often not available. To solve this problem we do exactly the same way as the first explorers did when they discovered the deep jungles, we draw a map as we go along. This is what SLAM (Simultaneous Localization And Mapping) is all about, it combines localization and mapping.

The SLAM problem

How can a body navigate in a previously unknown environment while constantly building and updating a map of its workspace using on board sensors only?

Method

Use of range measurement device

SLAM is more of a concept than an algorithm but it builds on using some kind of mapping sensor of the environment. Different sensor can be used for this for some commonly used ones are LIDAR (A laser time of flight sensor), sonar or cameras. There are problems and benefits with each one, LIDAR gives accurate readings, works with 2D and 3D but is very expensive and has problems with very specular surfaces. Sonar are very cheap but often give bad readings and the emitted scan scales up in size with 30 degrees compared to the laser light which is about 0.25 degrees. Finally we have the camera solution which has become popular quite recently with today's fast computers. Slow computing time has long been a bottleneck for this method. Using cameras has many advantages such as there is a lot of information to extract in a picture, the cameras are cheap, it is intuitively (humans work the same). The disadvantages are the complexity of the algorithms and won't work when completely dark. [1]

Landmarks

Landmarks are distinct features in the environment which are easily re-observed and hopefully stationary. The robot then uses these landmarks to localize itself. It compares where the landmarks were and where they went in order to finds it's position.

Many different algorithms exist to find these landmarks two examples are Spikes and RANSAC. Spikes looks in the input data for large gradients which often corresponds to sharp corners which are excellent landmarks.

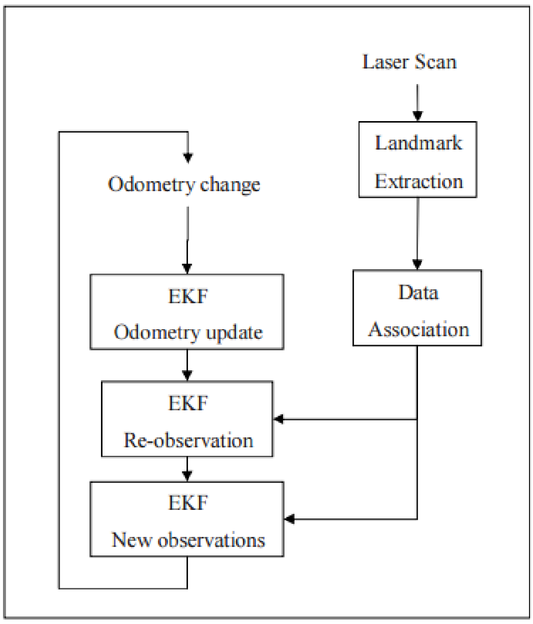

Figure 29: Overview of the SLAM process

RANSAC finds flat linear surfaces such as walls instead of corners. It does so by randomly selecting samples and fits a line with the least square method. Then it checks again how many data points actually are represented by this best fit line and if enough data points have small enough error then this data segment is used as a landmark.

Extended Kalman filter approach

See [1] for information on implementing a simple Extended Kalman filter. Also examine the process for using such a method in Figure 29.

State of the art of SLAM

Monocular SLAM

In order to commercialise self driving cars keeping a realistic price is important. Using LIDARs and other sensors can be expensive however using a few cameras is extremely cheap and which is of course desirable. Other autonomous products will surly in the future use visual SLAM or similar for positioning. How this works can be found in [2].

References

[1] *SLAM for dummies http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-412j-cognitive-robotics-spring-2005/projects/1aslam_ blas_ repo.pdf*

[2] *Monocular SLAM http://www.csc.kth.se/c̃elle/papers/jensfelt_ icra06.pdf*