Automated Self-Driving Parking System (ASPS)

Vehicular Networks (2017/2018)
Final Report

Afonso Maria Viegas Bota Carreira de Caetano
82539
afonso.caetano@tecnico.ulisboa.pt
Group 4

Table of Contents

1 Introduction

Parking is still considered one of the most challenging tasks nowadays, where drivers have to circle around a parking garage and look for an available parking space. After this, the vehicle must be moved backward into a narrow space with high probability of collisions. As a consequence, 40% of accidents with physical loss or damage occur when parking or maneuvering **Error! Reference source not found.**.The complexity lies in the generation of a suitable parking maneuver, in the control of the vehicle itself (forward/backward and left/right) and of course the avoidance of any obstacles that might exist in the environment [2].

A lot of today's modern cars offer some kind of parking assistant systems, that can start from simple ultrasonic sensors on the bumpers and a reverse camera to aid the driver when backing up, to fully automated parallel and perpendicular parking functions using a conjunction of sensors present in the vehicle. This solutions still require the user to find a suitable parking space and are quite limited in the maneuver they can perform since the parking space must fulfil a set of requirement for the system to be able to attempt the parking maneuver.

Also the problem of driving to find a suitable parking spot results on more CO2 emission. According to a 2007 study by Donald Shoup at the University of California, Los Angeles, drivers in a 15-block district of LA notched up a staggering 1.5 million kilometers a year looking for parking spaces. This is equivalent of 38 trips around the Earth and 178,000 liters of wasted gasoline resulting in 662 tons of carbon dioxide [3]. Not to mention that driving thru a parking lot looking for an available space presents a hassle to the driver and also a waste of time.

1.1 Motivation

Even with the parking aids specified above, there are still challenges that need to be addressed and here is where the Automated Self-Driving Parking System can be helpful. ASPS tackled this issues by introducing a new parking concept that allows drivers to leave their cars on a drop-off zone, e.g. parking garage entrance, while the car accomplished the parking process by its own. This in conjunction with the parking aid systems already present in modern cars, as specified above, removes the hassle of vehicle parking to the driver and passengers.

This project aims to efficiently and effectively manage parking spaces and the overall parking operation in parking garages by having the driver request a parking space to the system, e.g. thru an mobile application, and leaving the system to compute a route to a vacant space. This already removes the problem of waste of fuel, time and the unnecessary emission of carbon dioxide while looking for a parking spot.

Then using an array of communication and localization technologies, in this case V2X communication and A-GPS, the task of driving the vehicle to the allocated parking space reduces significantly the probability of a collision. This function can also be aided by sensing technologies that might be present in the vehicle, e.g. 3D surround cameras, ultrasonic sensors, radar, etc. Also the garage itself may use the same technologies (plus LIDAR technology) to better map the environment.

When the vehicle approaches the respective parking spot, the automated maneuver takes place and ensures the vehicle is perfectly parked inside the lines. This guarantees the best use of space possible, since there is no need to leave a space gap between vehicles for the driver and passenger to be able to enter and exit the vehicle since this entire operation does not have human intervention. This results in more parking spaces and more profit per square meter.

1.2 Report Structure

The Section below (section 2) provides the system architecture and contains the use cases alongside the requirements, network and nodes architecture finalizing with the communication protocols. Then on section 3 the implementation is presented, compromised of the design options and system and node implementations. The report is finalized with the tests section followed by a brief conclusion.

2 Architecture

2.1 Use cases

In our system only one use cases applies, for there could be two use cases, the parking operation where a user enters the garage, drops off his vehicle and the retrieval of the vehicle by the user. But since the second one is the reverse of the first both use cases can be simplified to only one as can be seen below.

2.1.1 In/Out of Parking Garage

In a scenario where a person (e.g. Matt) plans to go to a Shopping Mall in the center of a busy city. The way the mall deals with parking of multiple vehicles is by having an underground garage. The driver approaches the entrance of the garage and follows the signs to the drop-off area. Here he exits the car (OBU) and using the application activates the ASPS feature.

From here the driver can just go and mind his own business while the vehicle request the garage (Center-ITS) for an available parking space, by feeding the garage with information about the vehicle model in question (vehicle length and width), this can ensure that best suitable spot is chosen. When the route is calculated, step-by-step instructions are send to the vehicle, and periodically the vehicle sends it position in order for the garage to monitor it’s position along the way. By the time the vehicle approaches the parking spot, it starts the parking maneuver ensuring a correct parking.

When Matt returns from shopping, using the application, he requests his vehicle and waits at the pick-up area. The process of vehicle pick-up is just the reverse of the process specified above, being the only difference that now the vehicle instead of requesting a parking spot to the garage, it just simply receives a notification that the pick-up process is going to begin.

2.2 Requirements

The following is a list of previously defined requirements for the system:

- **Central server**, where all the requests, parking spaces mapping and allocation along with vehicles navigation is going to be implemented;
- **OBU** (On Board Unit – ITS), vehicle able to send and process messages along with a set of sensors with autonomous driving capabilities;
- **RSU** (Road Side Unit – IST), although optional, it aids the Central server with environment sensing and can also aid the vehicle directly. This can be a sensor like a LIDAR or a 3D camera.
- **Smartphone**, with several communication technologies like 3G/4G, Wi-Fi, Bluetooth, etc.

2.3 Network architecture

The system architecture is represented in figure 1.

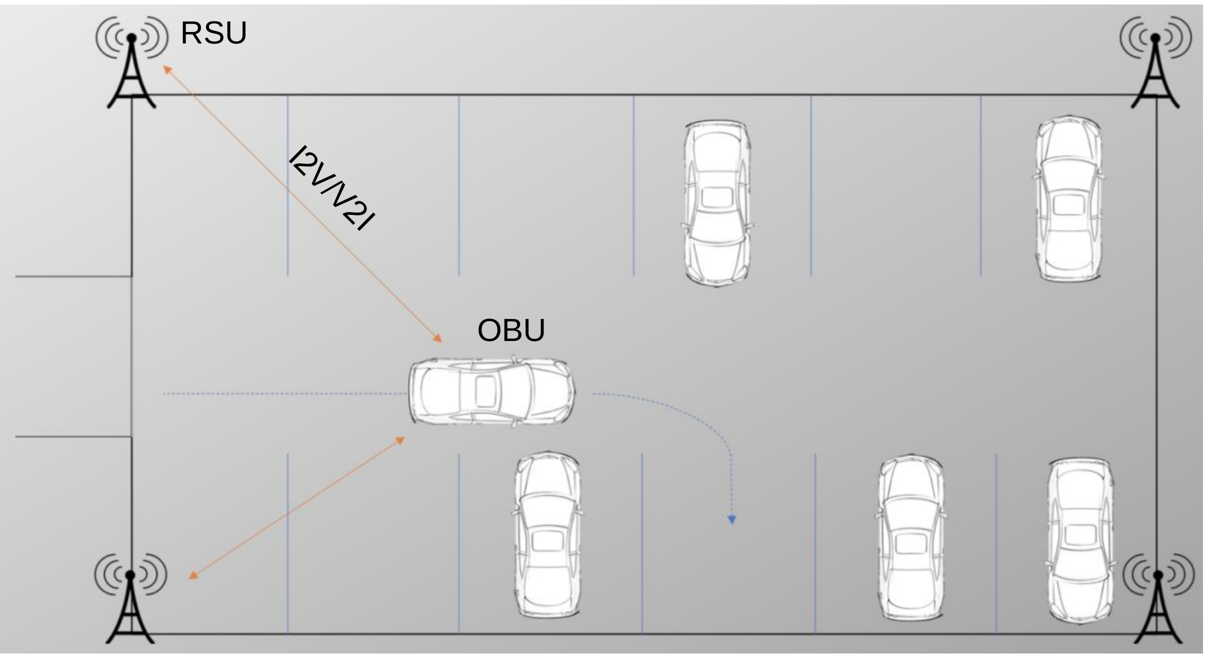

Figure 1 - ASPS network architecture

The system consists in three main components:

2.3.1 Central Server

It is the backbone of our system, it resides in the garage infrastructure and is responsible for the management of all parking spaces along with all parking operations. It communicates with all the RSU units by C2I (Center-to-infrastructure) communication.

2.3.2 Road Side Units

This ones are scattered along the garage and act as intermediary between the garage and the vehicles. Each unit communicates with the central server by I2C (Infrastructure-to-Center) communication and with the vehicles by I2V (Infrastructure-to-Vehicle) communication.

2.3.3 On Board Units

All vehicles are OBU's, and they communicate with the nearest RSU in order to perform the parking procedure. The communication between this two is done by V2I (Vehicle-to-Infrastructure) communication.

The interaction between the different nodes can be seen in Figure 2.

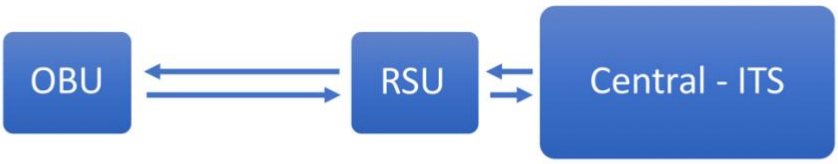

Figure 2 - Nodes interaction

2.4 Nodes architecture

On the next two figures (Figure 3 and 4) the architecture of the central server and the OBU can be seen respectively.

2.4.1 Center - ITS

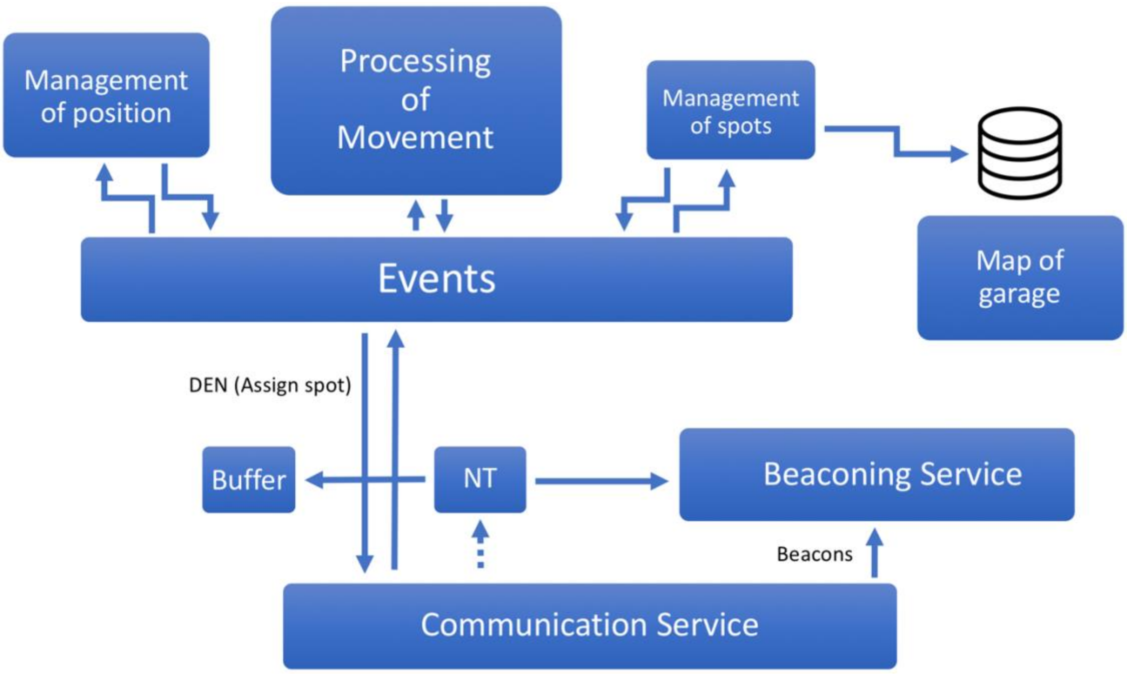

Figure 3 - Central server node architecture

The central unit is where all parking requests are processed, along with all the vehicles route calculation and monitoring. On a database a virtual map of the garage is stored alongside the occupied parking spots. Then we have a module responsible for all the vehicles movement processing as well as a module to process current vehicle positioning and route monitoring. This information passes the beaconing service implemented in the central server.

2.4.2 On Board Unit (OBU)

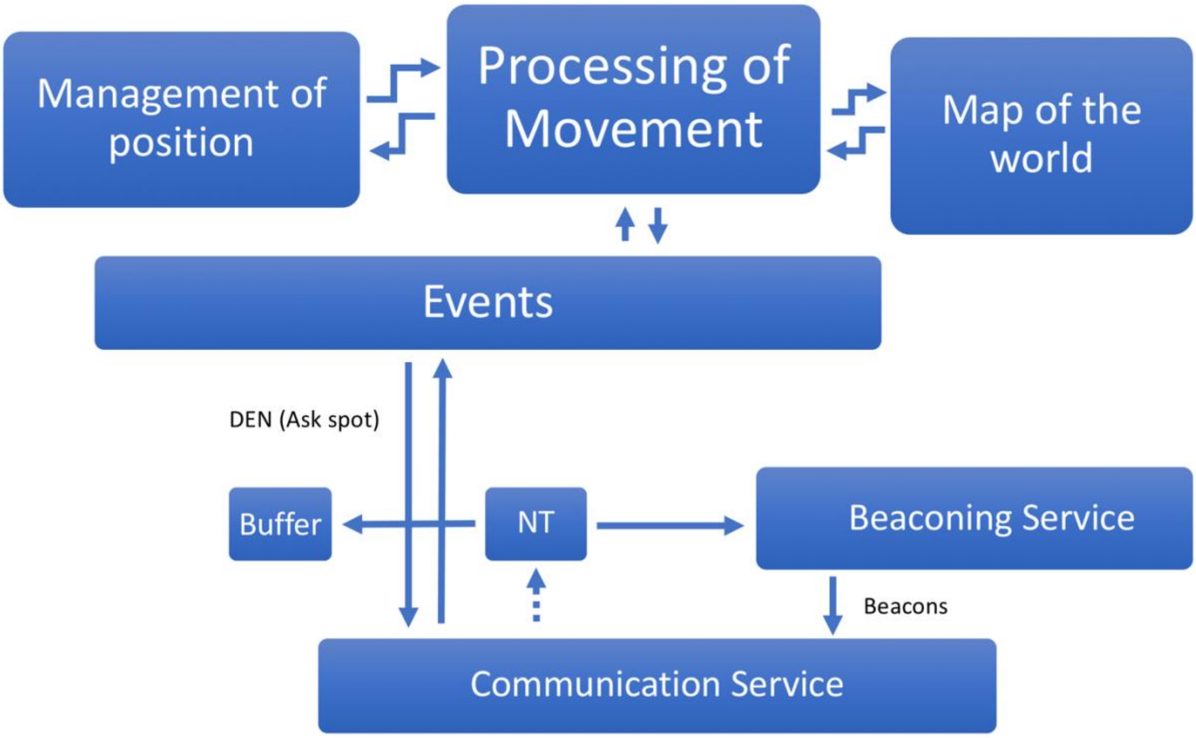

Figure 4 - OBU node architecture

On the vehicle all of the movement process is calculated with the aid of the virtual map of the garage and the management of the current position of the vehicle. The OBU receives step-by-step directions from the RSU, and performs this movement while sending periodic beacons with its current position.

2.4.3 Communication Module

In both components above the communication module is very similar (seen from the “Events” box to bottom) and consists of and Event handler which is the “bridge” between the communication module and the application logic. It responds to certain messages received from the Communication Service and also generates events in response to operations on the application level.

A Beaconing Service is present and is responsible for sending and receiving periodically messages from other components with information regarding position. Then a neighbor table (NT) is completed with the information regarding the nearest nodes. And the Buffer just hold the messages to the neighbors that are no longer near, to be send later.

2.5 Communication protocols

The system communication between all nodes uses a UDP and IPv6 because of the high number of vehicles circulating in major cities that need to be addressed and the current limitation of IPv4 addresses. Also the problem of IP addressing present in IPv4 where a central entity must distribute the addressing between nodes (DHCP) or the addresses must be configured manually can be eliminated by using Stateless Address Autoconfiguration (SLAAC) with IPv6.

The messages with information regarding parking requests made by the vehicles and the step-by-step movements are sent using Decentralized Environmental Notification Basic Service DEN messages. Because of this, it is necessary to specify how the communication is going to be performed between the entities. Below on Figure 5, DEN messages structure can be seen.

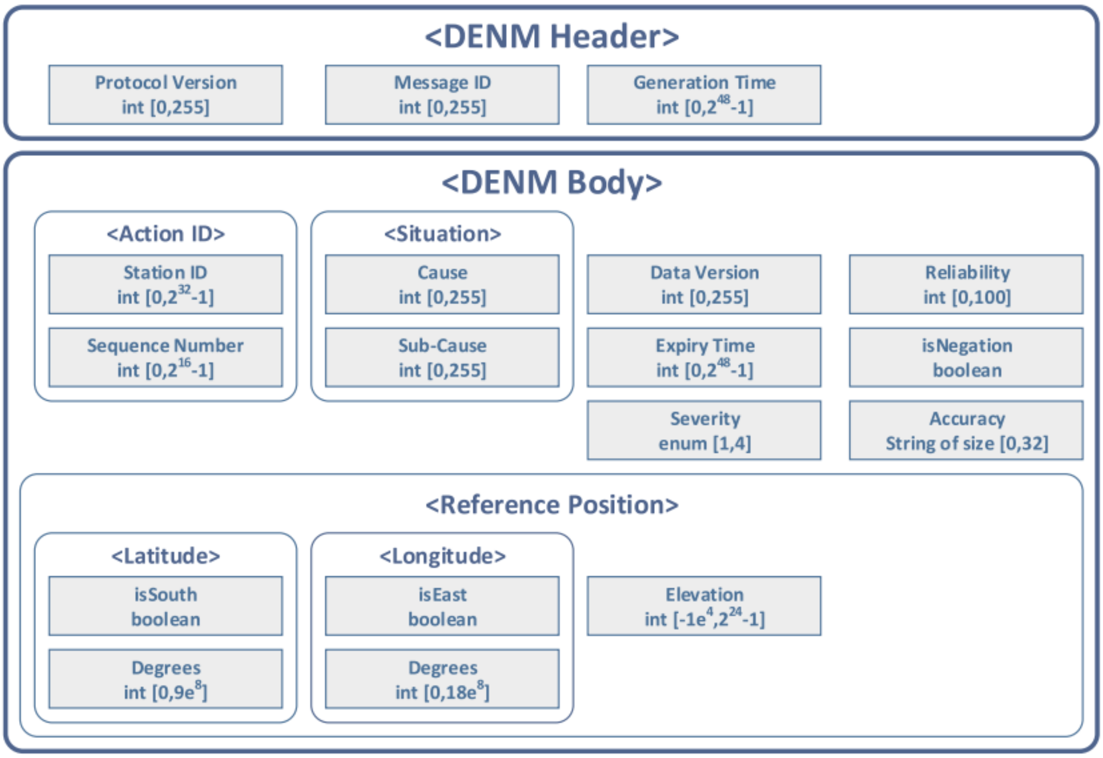

Figure 5 - DENM message structure

3 Implementation

On this section a solution of the Automated Self-Driving Parking System will be presented.

3.1 Design options

The options of implementation of the prototype can be separated in two main components, the hardware and the software.

3.1.1 Hardware

For the **Central Server** a MacBook Pro running Mac OS High Sierra (version 10.13.4) was chosen. The **OBU** consists of a toy vehicle provided by the teacher alongside a Raspberry Pi 0 used for sensing. On the following table (Table 1) the capabilities of the vehicle can be seen.

Vehicle				Computer Board			
Type	**Sensor**	**Actuator**	**Power supply**	**Model**	**Network interfaces**	**Serial interfaces**	**Power supply**
Generic vehicle	None	Traction motor Direction motor	AA / AA-Li	Raspberry Pi 0	Wi-Fi BT	GPIO	uUSB AA-Li

Table 1 - Vehicle capabilities

The **RSU** was supposed to be implement with a Raspberry Pi 0 connected with a ultrasonic sensor.

3.1.2 Software

The programming language chosen for the implementation was *Python* (version 3.6) and the reasons behind are below:

- **Readable and maintainable code** – Since the syntax allow to express concepts without writing additional code, while at the same time emphasizes on code readability;
- **Compatible with major platforms and systems** – At present, *Python* supports many operating systems and allows to run the same code on multiple platforms without recompilation;
- **Robust Standard Library** – A large and robust standard library makes this language score over others.

3.2 System Implementation

The OBU's communicate with the central server (Center – ITS) through a Wi-Fi hotspot provided by the OBU (Raspberry Pi 0) for testing purposes. And the RSU module would communicate with the central server thru an Ethernet connection, and would relay the messages to the OBU thru the Wi-Fi hotspot provided by the vehicle. The reverse procedure would apply so that the messages coming from the OBU would flow thru the Wi-Fi connection to the RSU, and then thru the Ethernet connection to the central server.

3.3 Node Implementation

This section will only describe the positioning tracking system implemented in the OBU component.

3.3.1 Assisted Global Position System (A-GPS)

Since there is a need to know where every vehicle is located inside the parking garage and because the majority of the parking lot are underground or have some kind of roof, making it almost impossible to have GPS signal inside. This is where A-GPS can be helpful, by measuring the distance the vehicle travelled is possible to estimate the relative position of the vehicle regarding the garage.

For this, a virtual mapping of the garage was created with a basic matrix, and both the central server and the OBU have access to this. The OBU uses this information and the information of its current location to estimate the current position. Then after receiving a DEN UPDATE from the central server with a movement to be performed, it estimates de relative position and updates periodically its current position. Of course this recently updated information is in all the Beacon messages sent making sure the most recent position is sent to the central server. Then with this information the server can monitor that the vehicle is travelling thru the correct route.

This can also be used to form a sort of a protection matrix around the vehicle, opening a door for some kind of obstacle avoidance system. By having each vehicle occupying not only its physical space on the matrix but also a border all around the OBU, e.g. a frame of 1 unit square all around the vehicle, it is possible to detect when two OBU's are very close together and might be in a collision course. When an intersection of both matrix frames of the vehicles is detected, the central server can send a DEN UPDATE message issuing the two vehicles to stop immediately in order to prevent a possible collision.

4 Tests

To make sure the prototype works as expected, two tests were performed being these the parking of an arriving vehicle and also the retrieval of a parked vehicle.

4.1 Test Scenario

Considering the scenario where the toy vehicle is positioned in the drop-off area, on the matrix represented by the coordinates (0,0). Then the vehicle requests the garage for an available parking space. Considering that the garage was configured with three parking spots, more precisely with one occupied space and two vacant spaces. After the vehicle request is processed by the server, a route is calculated and a step-by-step guidance is sent to the vehicle (DEN UPDATE).

The first movement received by the vehicle is to drive forward a certain distance until it approaches the allocated parking spot. After the server confirms the position of the vehicle by the beacons messages sent periodically by the OBU, it send the second DEN UPDATE message informing it to turn right 90 degrees into the parking spot.

After some time (aprox. 15sec), the retrieval process is initiated, and the garage issues a DEN UPDATE to the OBU containing the reverse movement while turning 90 degrees now to the left in order to be facing towards the pick-up area. Then a second message in inbound to the vehicle, after the vehicle position is confirmed by the central server, commanding the vehicle to drive forward the required distance in order to get to its final destination, the drop-off area.

4.2 Test Results

The results obtained from the test scenario above were partially achieved, due to the inconsistent behavior of the toy car. On two runs of the scenario on the same toy car the performance displayed by the electric motor alongside the steering actuator were very erratic.

Although the first part of the scenario, the parking operation, did present better result by having the vehicle always parked on the correct spot and between the lines, the retrieval of the vehicle presented some major challenges, beginning with the very bad performance the motor demonstrated while reversing associated with the poor turn angle made the exit of the parking space impossible. This resulted in the impossibility of demonstrating a correct exit of the parking spot, although it was possible to demonstrate the monitoring of the vehicle position by the garage. While reversing because the vehicle could not achieve the desired position, the server did not send the next move to be performed, resulting in the vehicle being stationary and not completing the maneuver.

The scenario above was performed on two different toy vehicles (Porsche Macan and Mini Countryman S) and both presented inconsistent behavior. This can be traced to the bad precision of the toy vehicles in lows speed operations that are absolutely necessary to our prototype.

5 Conclusions

The objective of this project was to develop a fully automated self-driving parking solution and we think we achieved a prototype demonstrating the possibility of a full scale implementation.

For future work without a doubt the implementation of more sensing technology like LIDAR (Laser Radar), 3D cameras, radar, ultrasonic sensors and many others in other to enhance the mapping capabilities of the system and consequently add feature like obstacle avoidance and multiple vehicles operations at the same time.

6 References

[1] *A sudden bang when parking. Allianz SE*, 2015, [online] Available: https://www.allianz.com/en/press/.

[2] Holger Banzhaf, Luigi Palmieri, Dennis Nienhüser, Thomas Schamm, Steffen Knoop, J. Marius Zöllner, "Hybrid curvature steer: A novel extend function for sampling-based nonholonomic motion planning in tight environments", *Intelligent Transportation Systems (ITSC) 2017 IEEE 20th International Conference on*, pp. 1-8, 2017, ISSN 2153-0017.

[3] Graham-Rowe, D. (2012). Parking sensors to take pain out of finding a space. Retrieved from https://www.newscientist.com/article/mg21328506.100-parking-sensors-to-take-pain-out-of-finding-a-space/#.Uw9h8vR5MdJ

[4]