

ELEKTRONISCHE MESS- UND  
DIAGNOSETECHNIK

( expleo )



# Multi Sensor Ramp Detection and Localization for Autonomous Valet Parking

Master thesis

Felix Saalfrank  
368199

February 21, 2022

Supervisor:  
Lars Schürmann  
Person 2  
Gutachter:  
Prof. Dr. Gühmann  
Prof. Dr. Orgelmeister

Technische Universität Berlin  
Faculty IV - Electrical Engineering and Computer Science  
Department of Energy and Automation Technology  
Chair of Electronic Measurement and Diagnostic Technology

Berlin, 3. Januar 2022

Masterarbeit für Herrn Felix Saalfrank, Matrikel-Nr. 368199

### **Multi-Sensor Ramp Detection and Localization for Autonomous Valet Parking**

**Problem:** Autonomous Valet Parking (AVP) will make parking easier in the future, by allowing the driver to exit the car in a drop off zone in front of a parking garage, and the car will find a parking spot on its own. When the driver calls the car again, it will also autonomously find its way to the driver. For this to work, a map of the parking garage and precise localization of the car is necessary. A challenging part is the necessary change of levels during the procedure because the ramps in parking garages are usually very narrow and require precise localization and control of the car. Therefore, information about whether or not the car is driving onto a ramp is necessary. This allows the controller of the car to adjust for the changing road conditions, e.g. increasing or decreasing the motor output power when driving up or down respectively. Also, because the maps used for the localization of the car are usually stored separately for each parking level, the loading of the new map should be initiated while the car is on a ramp.

**Task:** The goal of this thesis is to implement an algorithm for a car, which can detect ramps. Besides the detection, ramp properties such as the inclination angle or length should be measured. To implement this, various sensor setups will be used and compared. An Inertial Measurement Unit (IMU) will be the main sensor and will be responsible for the exact measurement of the ramp's properties, in conjunction with a wheel odometer. Additionally, a Light Detection and Ranging (LiDAR) sensor will be used to allow for the detection of the ramp, before entering it. The data of the LiDAR could also be fused with the IMU data to prevent false detections by the IMU. A camera will be tested as well for the early detection of a ramp and compared to the LiDAR. Test drives in one specific parking garage and test car will be performed. A camera will be used to validate if the detection was at the right time and the estimated ramp properties will be compared to manual measurements.

#### **Research steps:**

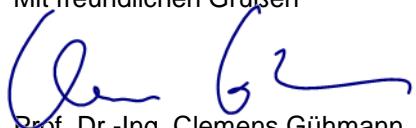
- Research of current methods to determine road grade angle using IMU, LiDAR or camera
- Comparison and selection of the most appropriate method for each sensor
- Implementation of a ramp detection algorithm
  - using an IMU
  - using a LiDAR sensor
  - using a camera
- Testing and optimizing of the methods
- Comparison and evaluation of the different methods used

- Documentation and presentation of the results and thesis

#### **Organization**

This thesis will be written in collaboration between TU-Berlin and Expleo Germany GmbH. Felix Saalfrank's supervisor is Lars Schürmann. All necessary documents and resources will be provided by Expleo Germany GmbH. The presentation of the researched literature, the analytical work and the experiments will be carried out according to the rules of best scientific practice. The results will be presented publicly in the seminar of the Electronic Measurement and Diagnostic Technology chair.

Mit freundlichen Grüßen



Prof. Dr.-Ing. Clemens Gühmann

# Kurzfassung

100-200 Wörter Kurzfassung (deutsch)

# Abstract

100-200 word abstract (english)

# **Declaration**

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

# Contents

<b>Acronyms</b>	<b>x</b>
<b>List of Symbols</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Outline . . . . .	1
<b>2 State of the art</b>	<b>2</b>
2.1 IMU . . . . .	2
2.2 LiDAR . . . . .	3
2.3 Camera . . . . .	3
2.4 Bla . . . . .	3
<b>3 Background</b>	<b>4</b>
3.1 Mathematical . . . . .	4
3.1.1 Mathematical Representations of 3D Orientations . . . . .	4
3.1.2 Vector projection . . . . .	5
3.2 Sensors . . . . .	5
3.2.1 IMU . . . . .	5
3.2.2 LiDAR . . . . .	8
3.2.3 Wheel speed sensor . . . . .	10
3.2.4 Camera? . . . . .	10
3.3 ROS . . . . .	10
3.4 Signal processing . . . . .	10
<b>4 Experimental Setup</b>	<b>12</b>
4.1 Sensors . . . . .	12
4.1.1 IMU . . . . .	12
4.1.2 LiDAR . . . . .	13
4.2 Sensor placement . . . . .	13
4.3 Car . . . . .	15
4.4 Garage . . . . .	16
<b>5 Methods</b>	<b>17</b>
5.1 IMU only . . . . .	18
5.1.1 Calibration . . . . .	18
5.1.2 Linear acceleration only . . . . .	19
5.1.3 Angular velocity only . . . . .	20
5.1.4 Complementary filter . . . . .	20
5.2 LiDAR only . . . . .	21
5.2.1 Calibration . . . . .	21

5.2.2	Algorithm . . . . .	22
5.3	Camera only . . . . .	24
5.4	Sensor fusion . . . . .	24
5.4.1	IMU and Odometer . . . . .	24
5.4.2	IMU and odometer and LiDAR . . . . .	25
<b>6</b>	<b>Results</b>	<b>26</b>
6.1	Evaluation concept . . . . .	26
6.2	Ramp metering? (IMU) . . . . .	26
6.3	Ramp detection (LiDAR and camera) . . . . .	27
<b>7</b>	<b>Conclusion</b>	<b>29</b>
<b>8</b>	<b>Appendix</b>	<b>33</b>

# List of Figures

3.1	A stable platform IMU [1] . . . . .	6
3.2	Micro structure of MEMS TODO: Better images . . . . .	7
3.3	Setup of a mechanical spinning LiDAR [2] . . . . .	9
4.1	The two used IMUs . . . . .	12
4.2	The two used Light Detection And Ranging (LiDAR)s . . . . .	14
4.3	Mounting of the LiDAR . . . . .	14
5.1	Car driving on a ramp. Due to forward acceleration the car tilts back. . . . .	17
5.2	Example of a typical sensor setup. All sensor must be aligned to the car frame. . . . .	18
5.3	Frame transformation . . . . .	19
5.4	Gravity measured by IMU (in car frame). . . . .	20
5.5	Algo for ramp detection . . . . .	23
5.6	Existing accelerations, when the car is accelerating . . . . .	25
6.1	Lidar points projected into the camera image. The green points were identified as part of a ramp. . . . .	28
6.2	caption . . . . .	28

# List of Tables

4.1	Comparison of the two used Intertial Measurement Unit (IMU)s [3, 4] . . . . .	13
4.2	Comparison of the two used LiDARs [5, 6] . . . . .	13
4.3	Some params . . . . .	14
4.4	Some params . . . . .	16
5.1	Used parameters for lidar algo . . . . .	24
6.1	Performance evaluation . . . . .	27
8.1	Performance evaluation . . . . .	33
8.2	Used parameters for lidar algo . . . . .	34

# Acronyms

**ABS** Anti-Lock Braking System

**AHRS** Attitude Heading Reference System

**AVP** Automated Valet Parking

**ESP** Electronic Stabilization Program

**FIR** Finite Impulse Response

**FOV** Field Of View

**GPS** Global Positioning System

**IIR** Infinite Impulse Response

**IMU** Intertial Measurement Unit

**LiDAR** Light Detection And Ranging

**MEMS** Microelectromechanical Systems

**RADAR** Radio Detection And Ranging

**RANSAC** Random Sample Consensus

**ROS** Robotic Operation System

**SLAM** Simultaneous Localization and Mapping

**SONAR** Sound Navigation And Ranging

# List of Symbols

${}^A_B \mathbf{q}$  Quaternion to transform from a to b

${}^A_B \mathbf{M}$  Rotation matrix to transform from a to b

$\mathbf{v}$  Vector

${}_{\mathcal{A}} \mathbf{v}$  Vector in coordinate frame  $\mathcal{A}$

$\|v\|$  Norm of vector

$\hat{\mathbf{v}}$  Unit vector

# Todo list

2do . . . . .	iii
2do . . . . .	iv
This is TODO note by my self . . . . .	1
This is an improvement note . . . . .	1
This is a question I have . . . . .	1
This is a suggestion from others . . . . .	1
Just copied this from the expose, needs improvement maybe some more sections about the specific topics or about goals . . . . .	1
2do . . . . .	1
Not really Intertial Measurement Unit (IMU), maybe in prev section instead . . . . .	2
what sensors does this paper use? . . . . .	2
I did not use most methods mentioned in the papers (e.g. Kalman filter). How to justify this? . . . . .	3
Performance	
And maybe implement something simple (from a package) . . . . .	3
2do . . . . .	3
How detailed should the explanation of other papers be? . . . . .	3
Short, about one sentence per paper . . . . .	3
2do . . . . .	3
Add ausblick was genommen wird . . . . .	3
Add image . . . . .	10
Probably not necessary, as everyone knows what a camera is . . . . .	10
Better explanation . . . . .	10
Not even sure if this needs an own section . . . . .	10
filtering also often introduces a delay . . . . .	11
How much about dsp? E.g. also noise, aa etc or only type of filters? . . . . .	11
what kind of filter? . . . . .	12
How to cite/link to a github package? . . . . .	12
i is natural number in range 0 to n, how to write? . . . . .	14
Instead of "long" derivation, just use the final equation? . . . . .	15
Improve text and table param description . . . . .	15
Text is bad e.g. "hacked" and table needs more rows or should be omitted . . . . .	15
Figure: Picture of eGolf . . . . .	15
Figure: Picture of ramps and/or figure of ramps showing angles . . . . .	16
Think of a good way to measure the true angle of the ramps . . . . .	16
Is this true? Was true when I tested it myself, but would've thought quat is faster . . . . .	19
Get correct equation and distinguish between discrete and continuous . . . . .	20
Whole text here is just copied from my notebook and will be rewritten . . . . .	21
Maybe brief overview of section/algorithm . . . . .	21
Add pcl citation somewhere [7] . . . . .	21
no idea if wheel chair ramps are common in parking garages? . . . . .	23

What happens with stairs? . . . . .	23
Add down ramp detection . . . . .	24
Add all algo parameters to table . . . . .	24
Or is this not important? . . . . .	24
In Anhang . . . . .	24
Nicht alle parameter einzeln erklaeren sonder nur die wichtigsten . . . . .	24
Brief explaination what sensor fusion is and why useful . . . . .	24
Explain model . . . . .	24
I can't explain it well . . . . .	25
Text is bad . . . . .	25
Algo description . . . . .	25
2do . . . . .	26
2do . . . . .	26
2do . . . . .	26
Raw accelerometer data or gyroscope peak is ramp start . . . . .	26
Also show consistency recordings (avg error and stuff) . . . . .	27
2do . . . . .	27
Make new recordings of level -2 ramps, approaching at different (e.g. 0, 15, 30, 45) angles . . . . .	27
2do . . . . .	27
Anzahl Fahrten als Spalte . . . . .	27
Subcategories rampentyp . . . . .	27

# Chapter 1

## Introduction

Note explanation

This is TODO note by my self

This is an improvement note

This is a question I have

This is a suggestion from others

### 1.1 Motivation

Parking is one of the most challenging driving tasks and the cause of almost half of the car accidents [8]. Current cars are already able to fully automated park on their own in parallel or perpendicular parking spaces. But due to the very limited space in cities, parking garages are often used in central areas [9]. Automated Valet Parking (AVP) allows for a fully automated parking experience. The car is left in a drop-off zone and finds a parking spot on its own. Afterwards the driver can give a command and the car leaves the parking spot again and picks up the driver. AVP saves time, the hassle of remembering the parking level and spot and furthermore allows to use the available space more efficiently and also minimizes the risk of collisions. For this to work an exact mapping of the environment and localization of the car in the garage is necessary.

This can be done either by Simultaneous Localization and Mapping (SLAM) or the area can be mapped beforehand (e.g. using Light Detection And Ranging (LiDAR) sensors) in which case only a localization of the car is necessary. The mapping can be done in 2D or 3D. 2D maps only show information of the current level. Hence if the car is driven up or down a ramp, the new map of the corresponding floor has to be loaded. Because the localization usually only works in a 2D-plane, a change of levels would not be detected. To solve this problem, a ramp detection has to be implemented.

Just copied this from the expose, needs improvement  
maybe some more sections about the specific topics or about goals

### 1.2 Outline

Brief overview over structure of thesis

# Chapter 2

## State of the art

### 2.1 IMU

In [10] different methods to estimate the road grade angle are discussed.

Not really Intertial Measurement Unit (IMU), maybe in prev section instead

There exist methods without Inertial Sensors relying on a model describing the longitudinal movement of the vehicle and the topology of the road. Both models are fused using a Kalman filter to improve the accuracy of the estimation [11]. A Kalman filter is also used in [12], where vehicle sensor data and Global Positioning System (GPS) data are fused. Besides the road grade, the vehicle mass is often also unknown and estimated as well, using common sensors of heavy duty vehicles [12] or

what sensors does this paper use?

[13]. More methods such as recursive least squares, extended Kalman filtering and a dynamic grade observer are discussed in [14]. Another method using GPS data and IMUs to calculate the vertical and horizontal velocity change respectively and thereby the road grade is proposed in [15]. [16] omits the IMU and relies on a GPS sensor and a barometer. GPS satellites broadcast information about their position and exact time to a GPS receiver, which than can calculate its position using triangulation [17]. While an accuracy of up to 1 m can be achieved when outside, the performance significantly drops when used indoors. The radio waves sent from the satellites are scattered, attenuated or blocked completely by walls and other obstacles, resulting in a very weak or even a complete loss of the signal [18].

Most methods mentioned above do not seem fit for the task, due to the reliance on GPS. Furthermore many internal measurements such as the engine torque, brake system usage, selected gear etc. can not easily be accessed and thus might not be available.

A method which does not use GPS, but only accelerometers and wheel odometers instead is described in [19] and [20]. The vehicle acceleration, calculated by deriving the wheel speed measurements in respect to time, is subtracted from the accelerometer signal in longitudinal direction. The remaining part is then the gravitational acceleration, which is zero if driving on flat ground but not anymore if driving on an elevated road, and can be used to calculate the road grade angle. A similar approach is used in [21]. [22] adds a gyroscope to the accelerometer and fuses their estimations using a quaternion unscented Kalman filter. The gyroscope measurements get integrated over time to receive the pitch angle. The angle from the angular velocity is accurate in short-term, but is suspect to drifting over time. The drift can be corrected by using the accelerometer signal, which is accurate in the long-term, but unlike the gyroscope not accurate in the short term. [23] uses all components of an IMU (meaning also the magnetometer) and fuses them using

a complementary filter. The estimated quaternions using the accelerometer and angular velocity measurements respectively are fused and the magnetometer data is used to improve the quaternion estimation from the accelerometer, but only if there are no magnetic disturbances. ANOTHER COMPL FILTER PAPER also uses an complementary filter, but fuses the estimated angle. from the accelerometer and gyroscope instead of the quaternions.

I did not use most methods mentioned in the papers (e.g. Kalman filter). How to justify this?

Performance

And maybe implement something simple (from a package)

## 2.2 LiDAR

- Light Detection And Ranging (LiDAR) methods for plane detection
- If found, methods for elevation estimation

How detailed should the explanation of other papers be?

Short, about one sentence per paper

## 2.3 Camera

- Something simple
- Stereo vs mono

[24] used an RGB-D sensor (camera image + depth sensor) to detect ramps for wheelchairs. Ramp properties such as angle, width, length and the orientation of the ramp were determined as well.

## 2.4 Bla

Due to the available sensor stack, not all mentioned methods can be tested. The gravity method and complementary filter will be tested for the IMU.

Add ausblick was genommen wird

# Chapter 3

## Background

### 3.1 Mathematical

#### 3.1.1 Mathematical Representations of 3D Orientations

Rotations can be expressed in several ways, I will briefly describe the ones used in this thesis.

##### 3.1.1.1 Rotation matrix

A rotation matrix  ${}^B_A \mathbf{M} \in \mathbb{R}^{3 \times 3}$  transforms an arbitrary vector from the coordinate system  $\mathcal{B}$  to the coordinate system  $\mathcal{A}$ . The rotation axis is the eigenvector for  $\lambda = 1$ , but the rotation angle can not easily be seen from the matrix. The rows of the rotation matrix are the axes of the new coordinate system. Rotation matrices can be concatenated, but this must be done in reverse order, e.g.

$${}^C_A \mathbf{p} = {}^C_B {}^B_A \mathbf{M} \mathbf{p} \quad (3.1)$$

to transform the point  ${}_A \mathbf{p}$  over the intermediate frame  $\mathcal{B}$  to the coordinate system  $\mathcal{C}$ .

##### 3.1.1.2 Euler angles

Euler angles are the closest to intuition but mathematically the worst way to represent orientations. Every orientation can be produced by a concatenation of three rotations around each of the coordinate axes. Because the resulting orientation depends on the order of which the rotations were performed, there are different conventions. Furthermore the conventions can be divided into intrinsic and extrinsic. Intrinsic rotation means that the coordinate system moves with the moving object, whereas with extrinsic rotations the original coordinate system remains static. The most common convention is the zyx (intrinsic I think?) (first rotation around z-axis and last around x-axis), where the angles are called yaw, pitch and roll.

The disadvantages are the many conventions and the possibility of singularity.

##### 3.1.1.3 Quaternions

A quaternion is defined as

$$q = q_w + q_x \mathbf{i} + q_y \mathbf{j} + q_z \mathbf{k} \quad (3.2)$$

with a real part  $q_w$  and three imaginary parts  $\mathbf{i}, \mathbf{j}$  and  $\mathbf{k}$  satisfying

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{i}\mathbf{j}\mathbf{k} = -1. \quad (3.3)$$

The quaternion can also be written in vector form by dividing the quaternion in a real or scalar and a imaginary or vector part

$$q = \begin{bmatrix} \mathbf{q} \\ q \end{bmatrix} = [q_x \quad q_y \quad q_z \quad q_w]^\top. \quad (3.4)$$

Note that in some literature the order of the real and imaginary part are swapped. If the quaternion is a unit vector, which means it satisfies

$$|q| = \sqrt{q^\top q} = \sqrt{|\mathbf{q}^2 + q_w^2|} = 1 \quad (3.5)$$

then it can be used to represent rotations. The rotation is then described with

$$q = \begin{bmatrix} \hat{\mathbf{j}} \cdot \sin\left(\frac{\alpha}{2}\right) \\ \cos\left(\frac{\alpha}{2}\right) \end{bmatrix} \quad (3.6)$$

with  $\alpha$  being the rotation angle and  $\hat{\mathbf{j}}$  being the normalized rotation axis. Each rotation is uniquely described by one rotation, but each orientation can be described by two unit quaternions with opposite signs. More information about quaternions can be found in [25].

A quaternion of the form  $\mathbf{q} = [w \quad x \quad y \quad z]^\top$  can be converted to a rotation matrix with

$$\mathbf{M} = \begin{bmatrix} 1 - 2y^2 - 2z^2 & 2(xy - zw) & 2(xz + yw) \\ 2(xy + zw) & 1 - 2x^2 - 2z^2 & 2(yz - xw) \\ 2(xz - yw) & 2(yz + xw) & 1 - 2x^2 - 2y^2 \end{bmatrix} \quad (3.7)$$

### 3.1.2 Vector projection

In general, the rotation to align a vector  $\mathbf{v}_1$  with a vector  $\mathbf{v}_2$  can be expressed using a quaternion. At first both vectors must be normalized, resulting in the two unit vectors  $\hat{\mathbf{v}}_1$  and  $\hat{\mathbf{v}}_2$ . The rotation axis is perpendicular to both vectors and can thus be calculated using the cross product, which is then divided by the norm, to get an unit vector

$$\mathbf{j} = \frac{\hat{\mathbf{v}}_1 \times \hat{\mathbf{v}}_2}{\|\hat{\mathbf{v}}_1 \times \hat{\mathbf{v}}_2\|}. \quad (3.8)$$

The angle between two vectors can be calculated with

$$\cos(\alpha) = \frac{\hat{\mathbf{v}}_1 \cdot \hat{\mathbf{v}}_2}{\|\hat{\mathbf{v}}_1\| \cdot \|\hat{\mathbf{v}}_2\|} = \hat{\mathbf{v}}_1 \cdot \hat{\mathbf{v}}_2 \implies \alpha = \arccos(\hat{\mathbf{v}}_1 \cdot \hat{\mathbf{v}}_2). \quad (3.9)$$

The denominator is equal to 1 (because the norm of an unit vector is 1) and thus omitted. The found rotation can then be applied using e.g. a quaternion or a rotation matrix.

## 3.2 Sensors

### 3.2.1 IMU

An Intertial Measurement Unit (IMU) is used to track the orientation and position of an object. Common uses are in the aerospace or automotive industry, often in combination with other sensors, to give information about the pose and position of a vehicle. More recently with the invention of Microelectromechanical Systems (MEMS) and specifically MEMS-IMUs which allow for a very small form factor at a low cost, IMUs are also used in consumer electronics such as smartphones or fitness tracker. An IMU usually consists of the three following sensors. The acceleration is measured using an accelerometer and can

be used to determine the velocity and the covered distance by integrating the acceleration with respect to time once respectively twice. The gyroscope gives information about the change of orientation. The third part is the magnetometer, which is able to measure the earth's magnetic field and is used to correct the measurements of the gyroscope. It allows for the determination of the absolute heading, whereas the gyroscope can only measure relative change. But because it is very sensitive to other magnetic objects, it is often omitted. IMUs can be typically divided into the two following categories.

In the first type, the stable platform systems, the inertial sensors are mounted in such

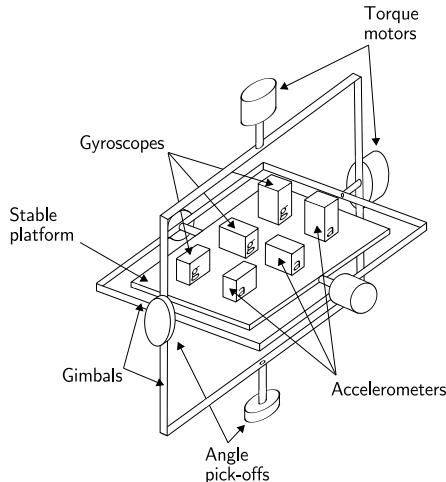


Figure 3.1: A stable platform IMU [1]

way, that they are always aligned with the reference frame. This is achieved using gimbals, which allow movement along all three axes. The gyroscopes on the platform measure the rotation and send them to torque motors, which rotate the gimbals to keep the platform in alignment with the reference frame. A typical setup of a stable platform system can be seen in fig. 3.1. The advantage of stable platform systems is that the calculation of orientation and position is straight forward. The angles of the gimbals can be measured to receive the orientation and position, the accelerometer measurements have to be corrected for gravity and be integrated two times. No coordinate transformation is necessary. The disadvantages are that the mechanical structure of the setup is complex, needs regular maintenance, requires a lot of space and has high costs.

The second type are strapdown systems, which are mostly used today. As the name suggests all the parts are fixed onto the device and are thus not anymore always aligned with the reference frame. Advantages are that due to the lack of gimbals and motors a significantly smaller build is possible while also being cheaper to mass produce. A disadvantage is that the calculation of the orientation and position is more complex, the rate gyroscopes have to be integrated to get the orientation and can then be used to transform the accelerometer signals into the reference frame. But with the decrease of computational cost this disadvantages continues to diminish.

There are many different types of gyroscopes and accelerometers such as mechanical, optical or solid state, but only the functionality of MEMS will be described, because those will also be used in the experiment. Information about the working principle of other systems and also much more information about IMUs in general can be found in [1].

MEMS consist of electrical and/or mechanical components in the size of 100 nm to 1 mm, allowing for a very small form factor. Other characteristics of MEMS are that they can easily be mass produced allowing for low cost and usually also need less power than traditional systems, because everything is integrated on the chip [26]. Almost all consumer grade electronics uses MEMS-IMUs nowadays, but they also find more and more use in

many industry segments, as their accuracy continues to improve [27].

### 3.2.1.1 MEMS Accelerometer

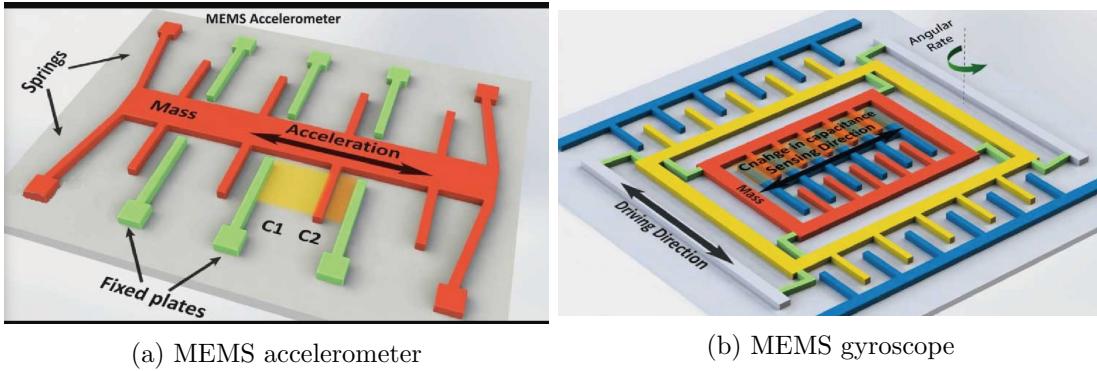


Figure 3.2: Micro structure of MEMS TODO: Better images

The accelerometer is used to measure the acceleration. Additional to the dynamic acceleration there is the static and constant gravity acceleration on earth, which is measured by the IMU in upward direction. This allows for the determination of one axis of the IMU, even if it is not moving. Often times only the dynamic accelerations are of interest and to get them the acceleration data during stand still must be measured and subtracted.

The micro structure of a MEMS accelerometer is shown in figure 3.2a. A mass is suspended by springs along one axis and if an acceleration along this axis occurs, the mass moves in the opposite direction due to Newton's second law. The mass has little fingers perpendicular to the axis along which the movement occurs, which affect the capacity between the fixed plates. The change of capacity and thus voltage can be measured, from which the acceleration can be calculated. To be able to measure the acceleration along all three axis the same setup is used three times, perpendicular to each other.

$$\hat{\mathbf{a}} = \mathbf{a} + \mathbf{b}_a + \mathbf{n}_a. \quad (3.10)$$

The measurement differs from the true acceleration  $\mathbf{a}$  due to random noise  $\mathbf{n}_a$  and a bias  $\mathbf{b}_a$ . More about the specific error terms can be read in 3.2.1.4.

### 3.2.1.2 MEMS Gyroscopes

A gyroscope measures the angular velocity. The setup of a MEMS gyroscope is similar to that of a MEMS accelerometer. A proof mass is suspended on a frame and responds to an input force. MEMS gyroscopes make use of the Coriolis effect, which states that an rotating object with the angular velocity  $w$  of mass  $m$  and velocity  $v$  experiences a force

$$F_C = -2m(w \times v). \quad (3.11)$$

To measure the effect, a mass is vibrating along one axis, which in turn is also suspended. If the mass is oscillating along one axis and a rotation is applied, a second oscillation on the axis perpendicular to the rotation axis can be observed. E.g. if the mass oscillates along the x-axis and a rotation around the z-axis is applied, a vibration along the y-axis can be observed. By measuring the amplitude and phase of the secondary oscillation the absolute value and direction of the angular velocity can be calculated. While MEMS gyroscopes do not achieve the same accuracy as optical gyroscopes they offer many advantages such as smaller physical properties (weight and size), lower power consumption and startup time as well as a significantly lower cost. MEMS gyroscopes have replaced

other gyroscope types in most areas, but in areas where the highest precision possible is necessary, typically in military industry, optical gyroscopes are still used today [27]. Similar to the accelerometer, the measurements  $\omega$  of the MEMS gyroscope are influenced by errors, which can be modeled as

$$\hat{\omega} = \omega + \mathbf{b}_g + \mathbf{n}_g \quad (3.12)$$

where  $\mathbf{b}_g$  is the bias vector and  $\mathbf{n}_g$  the random noise vector.

### 3.2.1.3 (MEMS) Magnetometer

A magnetometer measures the local magnetic field. Most sensors work using the Hall effect. A current is set to flow through a conductive plate. Without the presence of a magnetic field the electrons flow in a straight line, but if a magnetic field is introduced the electrons do get deflected to one side. The voltage between the two sides can then be measured, from which the strength and direction of the magnetic field can be determined. Without any magnetic disturbances, the magnetometer measures a constant local magnetic field vector. The vector points to magnetic North and can thus be used to determine the heading. Because the magnitude of the earth's magnetic field is very low ( $25\text{ }\mu\text{T}$  to  $65\text{ }\mu\text{T}$ ), the magnetometer readings can easily be influenced by other objects [28]. The distortions can be divided into two categories: hard or soft iron. Hard iron distortions are created by objects which actively produce a magnetic field, causing a permanent bias. Soft iron disturbances are due to deflections or alterations of an existing magnetic field. Both types of disturbances can be removed with a proper calibration if their position and orientation, relative to the sensor, stays the same [29]. Every time the sensor is placed in a (magnetically) new environment, a recalibration is necessary.

### 3.2.1.4 Typical MEMS errors

The errors can be divided into two categories: systematic and stochastic errors [30]. Systematic errors or also known as calibration errors are constant over time and can be eliminated by calibration. Typical examples are bias (offset), scaling or axis misalignments. Integrating a constant bias once or twice leads to a drift (error grows linearly with respect to time) or a second-order drift (error grows quadratically) respectively. Hence the elimination of the bias is necessary to get reliable estimations of the orientation, velocity or position, which are calculated by integrating the angular velocity or the accelerometer measurements.

Stochastic errors change at every measurement and can be modeled using a statistical approach. The turn-on bias is different every time the IMU is powered up, but can be eliminated after a rest period. Errors due to temperature fluctuations influencing the measurements are also common, but because most IMUs are equipped with a temperature sensor, the introduced error can be eliminated. Harder to correct is the introduced error due to thermo-mechanical noise, which is measured as white noise. The integration of white noise leads than to a random walk.

Angle errors introduced by random walk are usually the hardest to correct and are the reason, why the measurements of the gyroscope can not be trusted over a long period of time [1].

## 3.2.2 LiDAR

Light Detection And Ranging (LiDAR) is a method to measure distance to objects. Similar to other systems such as Sound Navigation And Ranging (SONAR) or Radio Detection And Ranging (RADAR), LiDAR uses the time-of-flight principle. A short laser pulse with

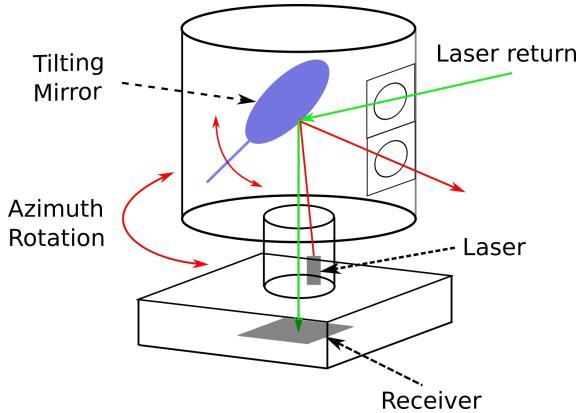


Figure 3.3: Setup of a mechanical spinning LiDAR [2]

the velocity  $v$  is sent into the environment and the reflected light is analyzed. The duration  $\Delta t$  it took from sending to receiving can then be used to calculate the distance  $s$  between the LiDAR and the object that the light hit with

$$s = c \frac{\Delta t}{2} \quad (3.13)$$

with  $c$  being the speed of light. The change of intensity and wavelength of the returning light are measured as well and can provide information about the reflectivity of the object (intensity) or the chemical composition of the air (wavelength). Common uses of LiDAR are the analysis of earth's atmosphere, 3D mapping of environments or in the field of autonomous driving for object detection, tracking and Simultaneous Localization and Mapping (SLAM). Basically all applications which use RADAR can also be used with a LiDAR instead, allowing for a greater accuracy.

There are different LiDAR types but their working principles are similar. A transmitter generates a signal and sends it into the environment using a scanning system and a transmission optic. As transmitter a laser with a wavelength of 850 nm to 950 nm (near-infrared) is typically used. The scanning system allows the laser to explore a large area instead of only a single point by steering the light at different azimuths and vertical angles and can be divided in mechanical spinning or solid state systems. Mechanical spinning systems is the oldest technology and is still mainly used today. A mirror which can be rotated around an axis is used, allowing for a greater vertical Field Of View (FOV). Moreover the whole LiDAR base on which the laser is mounted can be rotated independently from the mirror, allowing for a 360° horizontal FOV. To get a sufficient resolution the LiDAR has to spin at a high speed, but some LiDARs also use additionally a vertical array of lasers instead of only one to further increase the density of the generated point cloud. The working principle of a LiDAR using the mechanical spinning method is shown in figure 3.3. While mechanical spinning systems are very precise, they are bulky, need a lot of power and are expensive [31].

Solid state systems and especially MEMS LiDARs try to overcome those problems. MEMS-LiDAR are quasi-static, the only part that moves is the on the chip embedded mirror, but because of the small size (1 mm to 7 mm diameter), very little power has to be used to move it. The mirror can be rotated on up to two axes, but because the base cannot be rotated as with mechanical systems, a horizontal view of 360° is not possible. Though by using multiple lasers with different incident angles the FOV can be increased. Advantages of MEMS-LiDARs compared to mechanical systems are the smaller form factor and lower cost [32].

After transmitting the laser signal the reflected light passes through the receiving optic

and is received by photodetectors. A processing unit then generates a 3D point cloud from all the received measurements.

### 3.2.3 Wheel speed sensor

The wheel speed sensors measure the speed of each wheel and allow for the calculation of the car velocity. The measurements are also used by many driver assistance systems such as Anti-Lock Braking System (ABS) to be able to detect wheel slip. Different techniques exist to measure the speed with the most common ones being magnetoelectric and Hall type wheel speed sensors.

The magnetoelectric sensor is composed of a sensor head and a ring gear. The head is mounted stationary on the car frame while the ring gear is mounted on the wheel hub or axle and rotates with the wheel. The sensor head is composed of a permanent magnet core and a coil. When the wheel and thus the ring gear turns the teeth and gaps of the wheel pass by the sensor head and change the magnetic field which induces an alternating voltage in the coil. The amplitude and frequency of the induced voltage increases with increasing wheel speed. Advantages of the technique are the low cost, robustness and good performance even in the presence of mud etc. A disadvantage is the frequency dependency, very low speeds can not be measured due to the induced voltage being too small, while at very high speeds the changes can not be picked anymore up by the head [33].

Nowadays, almost exclusively the hall wheel speed sensor is used. The functionality is similar to that of the magnetoelectric sensor, but instead of ring gear a ring, on which alternating north and south magnets are placed, is used. The hall element in the sensor head measures the alternating magnetic field. A signal amplifier and processing unit is integrated in the sensor head and thus allows for a greater detection rate and range [34].

Add image

### 3.2.4 Camera?

Probably not necessary, as everyone knows what a camera is

## 3.3 ROS

Robotic Operation System (ROS) is a framework that allows the communication between sensors and actuators of a robot. It is a meta-operating system and provides services such as hardware abstraction, low-level device control..... Different languages such as C++, Python or Lisp are supported. The fundamental concepts are nodes, messages, topics and services [35]. A node is a process that performs computation and should be responsible for only one task. A package can contain multiple nodes. The communication between nodes is done using messages. There are different type of messages, but they all consist of standard types such as integer, float or bool but can also contain other messages. The messages are published on a specific topic. The topic can then be subscribed by other nodes, to retrieve the messages. A topic can be published and subscribed by multiple different nodes. Services allow for a synchronized communication, instead of the asynchronous topics.

Better explanation

## 3.4 Signal processing

Not even sure if this needs an own section

... is necessary. ... can be divided into filtering and smoothing. Filtering can be used in live applications and produces an estimate of the current value by taking the past values into account, whereas smoothing uses past and future samples and thus introduces a delay if used on live data. Because the detection should be live, only the filtering methods will be examined. Digital filters can be generally divided into two different categories. Finite Impulse Response (FIR) filter rely on a fixed number of recent input values. An example would be the moving average filter, which takes the past  $n$  values into account. Infinite Impulse Response (IIR) filter rely on previous output and most recent input by summing all points with a certain weight (e.g. exponential filter). This also explains the naming of the two types, the FIR filter "forgets" past values, whereas the IIR filter uses the previous estimate and thus theoretically takes all past values into account.

Savitzky-Golay filter

$$y = \frac{1}{h} \sum_{i=\frac{1-m}{2}}^{\frac{m-1}{2}} C \quad (3.14)$$

filtering also often  
introduces a delay

How much about dsp? E.g. also noise, aa etc or only type of filters?

# Chapter 4

## Experimental Setup

### 4.1 Sensors

#### 4.1.1 IMU

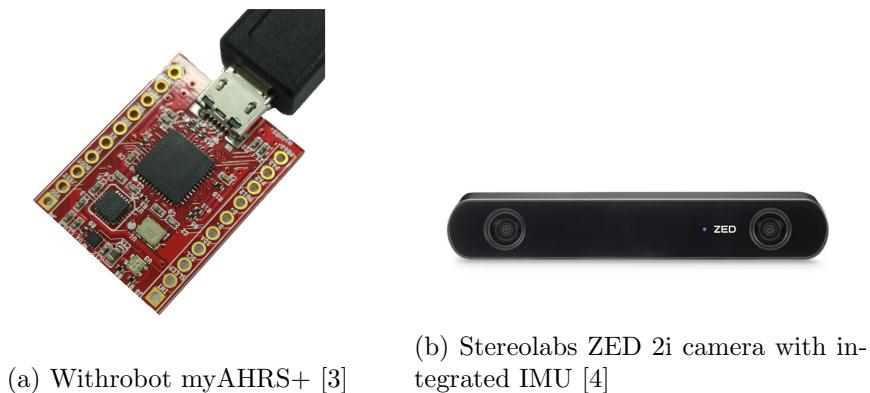


Figure 4.1: The two used IMUs

Two different Intertial Measurement Unit (IMU)s will be used for the experiment. The first one being the Withrobot myAHRS+, a low cost high performance Attitude Heading Reference System (AHRS). An AHRS contains an IMU and outputs the raw measurements, but also has an on-board processing system which estimates attitude and heading. The myAHRS+ uses an extended Kalman filter for the estimation and outputs the estimation in quaternion form and also in euler angles. The second IMU used during the experiment is integrated in the Stereolabs ZED 2i stereo camera. The ZED 2i camera has two horizontally displaced lenses, allowing for stereo vision and thus also depth estimation. An barometer, temperature sensor and an IMU are integrated as well. The ZED 2i is also an AHRS and estimates the orientation using . Both sensors are connected to the computer via USB. The specifications of each IMU can be read in table 4.1. Because no information about the noise density or random walk of the myAHRS+ IMU could be found, a manual estimation of the error was done using the package `allan_variance_ros`<sup>1</sup>. It analyzes the allan variance from IMU measurement data recorded over a two hour period, during which the IMU has not been moved.

How to cite/link to a github package?

<sup>1</sup>[https://github.com/gaowenliang imu\\_utils](https://github.com/gaowenliang imu_utils)

Table 4.1: Comparison of the two used IMUs [3, 4]

	myAHRS+	ZED 2i IMU
Accelerometer range	$\pm 16 \text{ g}$	$\pm 8 \text{ g}$
Gyroscope range	$\pm 2000 \frac{\circ}{\text{s}}$	$\pm 1000 \frac{\circ}{\text{s}}$
Magnetometer range	$\pm 1200 \mu\text{T}$	$\pm 2500 \mu\text{T}$
Rate	100 Hz	400 Hz
Accelerometer noise density	$4.502 \times 10^{-3} \frac{\text{m}}{\text{s}^2 \sqrt{\text{Hz}}}$	$1.148 \times 10^{-3} \frac{\text{m}}{\text{s}^2 \sqrt{\text{Hz}}}$
Accelerometer random walk	$7.337 \times 10^{-5} \frac{\text{m}}{\text{s}^3 \sqrt{\text{Hz}}}$	$6.458 \times 10^{-5} \frac{\text{m}}{\text{s}^3 \sqrt{\text{Hz}}}$
Gyroscope noise density	$1.674 \times 10^{-4} \frac{\text{rad}}{\text{s} \sqrt{\text{Hz}}}$	$8.254 \times 10^{-5} \frac{\text{rad}}{\text{s} \sqrt{\text{Hz}}}$
Gyroscope random walk	$5.042 \times 10^{-6} \frac{\text{rad}}{\text{s}^2 \sqrt{\text{Hz}}}$	$1.632 \times 10^{-7} \frac{\text{rad}}{\text{s}^2 \sqrt{\text{Hz}}}$

### 4.1.2 LiDAR

Two different Light Detection And Ranging (LiDAR)s will be used during the experiment. The RS-Bpearl and the Velodyne UltraPuck. The most relevant specifications of the two LiDARs can be seen in table 4.2. Both are mechanical LiDARs and have the same number of laser channels, but the Velodyne has a significant better vertical resolution, due to the smaller vertical Field Of View (FOV). Both LiDARs need an external power supply and transfer the data over Ethernet.

Table 4.2: Comparison of the two used LiDARs [5, 6]

	RS-Bpearl	Velodyne Ultra Puck
Channels	32	32
Range	100 m	200 m
Range accuracy	$\pm 3 \text{ cm}$	$\pm 3 \text{ cm}$
Horizontal FOV	360°	360°
Vertical FOV	90°	40° (-25° to 15°)
Horizontal resolution	0.2° to 0.4°	0.1° to 0.4°
Vertical resolution	2.81°	0.33°
Frame rate	10 Hz to 20 Hz	5 Hz to 20 Hz
Laser wavelength	905 nm	903 nm
Points per second	576,000	600,000

## 4.2 Sensor placement

The IMU must be placed on a rigid point of the car, such that the IMU's position stays always the same relative to the car. Other then that it should also be placed in the transversal center? to guarantee that the centripetal acceleration is not skewed towards one side. The ZED 2i camera with the integrated IMU was placed on the roof of the car, because the camera images were used as well. The myAHRS+ IMU was mounted on the floor of the car trunk.

The LiDAR will be placed on top of the car, to get a greater FOV. The pitch angle  $\beta$  at which the LiDAR will be mounted should be chosen such that the number of points in the area at the beginning of the ramp are maximized. This allows for the most accurate distinction between planes of different inclination angles and therefore also a good distance estimation. Because the distance to the ramp is not constant due to the movement of the car, the optimization can only be done for a specific distance. The coordinates at which



(a) Robosense RS-Bpearl [5] (b) Velodyne Ultra puck [6]

Figure 4.2: The two used LiDARs

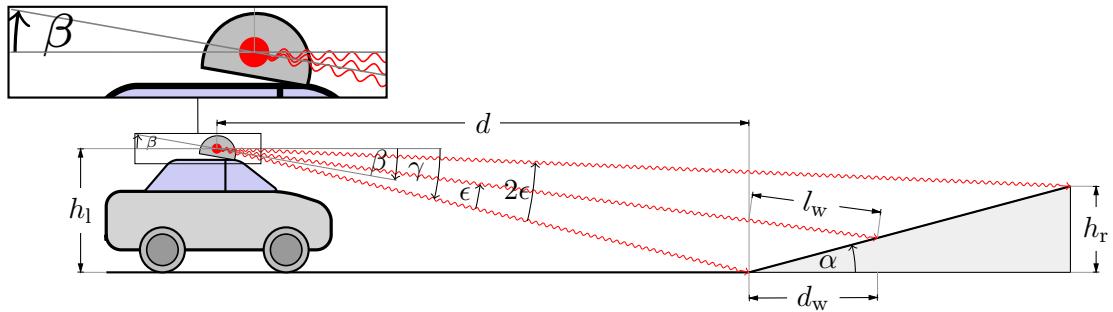


Figure 4.3: Mounting of the LiDAR

the lasers hit the ground and ramp depend on the height of the LiDAR  $h_1$ , the distance to the ramp  $d$ , the angle of the ramp  $\alpha$ , the angle  $\beta$  at which the LiDAR has been mounted on the car and finally on the vertical resolution  $\epsilon$  and FOV of the LiDAR. The coordinates can be calculated in the following way.

The angle  $\gamma$  between the NOT ground and each laser wave is defined as

$$\gamma = \beta - i\epsilon \quad (4.1)$$

with  $i \in \mathbb{N}, 0 \dots n$  being the "laserID" starting from the lowest opening angle and going to the highest.

i is natural number in range 0 to n, how to write?

Table 4.3: Some params

Variable	Description	Unit
$h_1$	LiDAR height above ground	m
$d$	Distance to ramp	m
$h_r$	Height of ramp	m
$l_w$	Light travel distance from ramp start to contact point	m
$d_w$	Distance from ramp start to contact point	m
$\alpha$	Ramp angle	°
$\beta$	LiDAR mount angle	°
$\gamma$	True angle?!	°
$\epsilon$	LiDAR vertical resolution	°
$n$	Number of laser channels	

On flat ground the distance at which the laser waves hit the ground can be calculated with

$$x = \tan(90^\circ - \gamma)h_l. \quad (4.2)$$

With a ramp, the assumption from 4.2 does not hold anymore. The light does not travel as far. The height above ground, when the light is at the beginning of the ramp can be calculated with

$$h_{w,start} = h_l - d \tan(\gamma) \quad (4.3)$$

The distance  $l_w$  which the light travels from the beginning of the ramp to the contact point with the ramp can be calculated using the law of sines

$$l_w = \frac{h_{w,start}}{\sin(\alpha + \gamma)} \sin(90^\circ - \alpha) \quad (4.4)$$

$$= -\frac{h_{w,start}}{\sin(\alpha + \gamma)} \cos(\alpha) \quad (4.5)$$

The travelled distance along the x-axis from the start of the ramp to the contact point is then

$$d_w = l_w \sin(90^\circ - 2\alpha - \gamma) \quad (4.6)$$

which results in the total x distance from the LiDAR

$$d_{hit} = d + d_w. \quad (4.7)$$

Instead of "long" derivation, just use the final equation?

$$d_{hit} = d + \frac{h_l - d \tan(\gamma)}{\sin(\alpha + \gamma)} \cos(-\alpha) \sin(90^\circ - 2\alpha - \gamma) \quad (4.8)$$

Using the formulas the optimal mounting pitch angle  $\beta$  for the two LiDARs has been found with  $\beta_{velodyne} = 0^\circ$  and  $\beta_{robos} = 20^\circ$ , optimized for a distance of 10m to the ramp. The angle between the two LiDARs differs due to the different starting opening angle of  $-25^\circ$  and  $0^\circ$  for the velodyne and robosense respectively.

Improve text and table param description

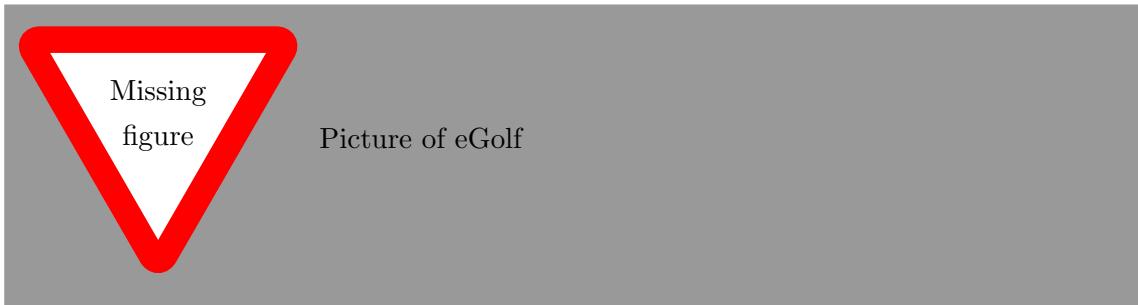
### 4.3 Car

The car used in the experiment is an eGolf 2017. Being an electric car it provides better vibration properties than a car with an internal combustion engine, which means that the IMU measurements are less influenced by external noise. The car has been "hacked" which allows for the reading of the wheel ticks by tapping the signals from the CAN bus. Unfortunately the wheel speed readings are only available in this "hacked" state, in which the output power of the motor is limited. In this mode, the maximum speed is capped at  $5 \frac{\text{km}}{\text{h}}$  and the power is not enough to allow for the traversing of ramps. The car can only make it about half way up. Because of that, the normal mode was used to drive between different levels. Before driving down, the mode was switched again to also provide the wheel speed measurements. The car has a pc in the booth, at which all the sensors were connected to.

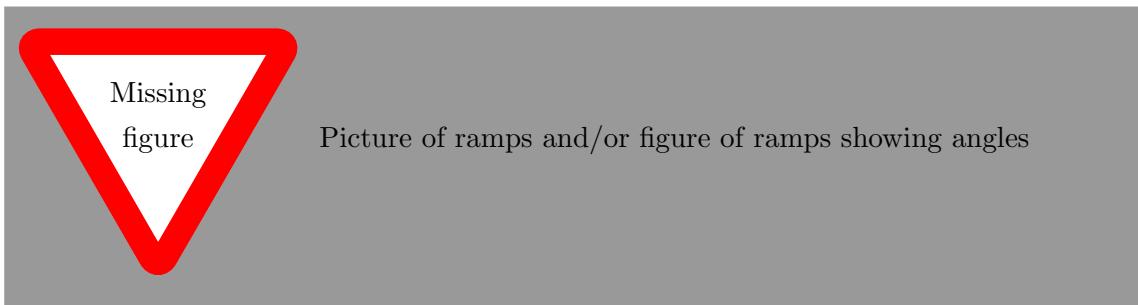
Text is bad e.g. "hacked" and table needs more rows or should be omitted

Table 4.4: Some params

Variable	Value	Unit
Wheelbase	2.63	m
Track width	1.512	m
Max linear velocity x	1.389	$\frac{m}{s}$
Max linear acceleration x	0.5	$\frac{m}{s^2}$



## 4.4 Garage



Think of a good way to measure the true angle of the ramps

# Chapter 5

## Methods

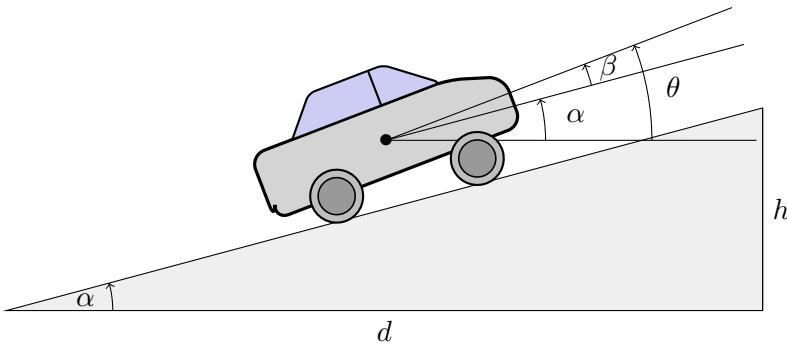


Figure 5.1: Car driving on a ramp. Due to forward acceleration the car tilts back.

The road grade  $\alpha$  is the angle between the road plane and the ground plane. The ground plane is perpendicular to the gravity vector. The road grade can be represented in degrees

$$\alpha = \arctan\left(\frac{h}{d}\right) \quad (5.1)$$

or in percentage

$$r = 100 \cdot \tan(\alpha). \quad (5.2)$$

The pitch angle  $\theta$  of the car is defined as the angle between the ground plane and the longitudinal axis of the car. An acceleration or deceleration on the ramp compresses the suspension in the back or respectively front, which makes the pitch angle not in alignment with the road grade anymore. The difference between the two angles is defined as  $\beta = \theta - \alpha$  and may also occur due to rotational movement or vibrations. The mentioned variables are visualized in fig. 5.1.

In this chapter different sensors and methods will be tested to estimate the car's pitch angle. A common problem is that the coordinate frame of the sensor is usually not aligned with car's frame, see. fig. 5.2. To determine the true car pitch angle the sensor frame must be transformed onto the car frame. Semi-automatic calibration methods for the Intertial Measurement Unit (IMU) and Light Detection And Ranging (LiDAR) sensor will be represented, which determine the necessary rotation to align both frames. But the translation difference can not be easily estimated and must be measured by hand.

At first the accelerometer and gyroscope of the IMU will be used separately determine the pitch angle. Then it will be tested how the estimate improves when both sensors will be used together in a complementary filter. The gravity method, which involves the accelerometer and wheel sensor measurements will be tested as well. A method using a

LiDAR sensor will be represented in the end, followed by a final method which incorporates all available sensors.

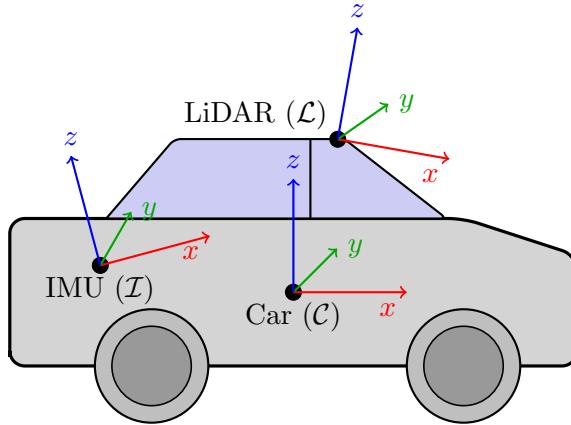


Figure 5.2: Example of a typical sensor setup. All sensor must be aligned to the car frame.

## 5.1 IMU only

### 5.1.1 Calibration

The IMU is usually not placed in such a way, that the coordinate frame of the device  $\mathcal{I}$  aligns with that of the car  $\mathcal{C}$ , see fig. 5.3a. Because of that, a transformation between the two frames must be found. This can be achieved using a rotation matrix  $\mathcal{I}^{\mathcal{C}}\mathbf{M} \in \mathbb{R}^{3 \times 3}$  which transforms the measurements of the linear acceleration  $\mathcal{I}\mathbf{a}_n \in \mathbb{R}^{1 \times 3}$  and angular velocity  $\mathcal{I}\mathbf{v}_n \in \mathbb{R}^{1 \times 3}$  into the car frame. Note that the upper index to the left of the matrix symbol denotes the source frame, whereas the destination frame is written below it.  $n \in \mathbb{N}$  is the time step.

During standstill, the only measurable acceleration besides noise and bias is the acceleration due to gravity. Assuming the car stands on flat ground, the gravity acceleration in the car frame is measured only in upwards z-direction. Using this, a transformation from IMU frame  $\mathcal{I}$  to the intermediate frame  $\mathcal{B}$  can be found. In the new  $\mathcal{B}$  frame both z-axes are aligned  $\mathcal{B}\mathbf{z} = \mathcal{C}\mathbf{z}$  and thus the pitch and roll angle between the two frames become zero. Note that this is not necessarily true for the other axes,  $\mathcal{B}\mathbf{x} \neq \mathcal{C}\mathbf{x}$  and  $\mathcal{B}\mathbf{y} \neq \mathcal{C}\mathbf{y}$ , see fig. 5.3b.

According to Euler's rotation theorem, which says that any arbitrary rotation of a rigid body while holding one point (origin) fixed can be achieved by a rotation around a single fixed axis passing through the origin, there exists one rotation axis  $\mathbf{j}$  and rotation angle  $\alpha$  to achieve this.

As described in section 3.1.2, the rotation axis needed for the transformation can be calculated with

$$\mathbf{j} = \frac{\mathcal{I}\hat{\mathbf{a}} \times \mathcal{C}\hat{\mathbf{z}}}{\|\mathcal{I}\hat{\mathbf{a}} \times \mathcal{C}\hat{\mathbf{z}}\|} \quad (5.3)$$

and the rotation angle with

$$\alpha = \arccos(\mathcal{I}\hat{\mathbf{a}} \cdot \mathcal{C}\hat{\mathbf{z}}). \quad (5.4)$$

with  $\mathcal{I}\hat{\mathbf{a}} \in \mathbb{R}^{1 \times 3}$  being the normalized measured linear acceleration in the IMU frame and  $\mathcal{C}\hat{\mathbf{z}}$  the (normalized) z-axis of the car.

The quaternion

$$\mathcal{B}\mathbf{q} = \begin{bmatrix} \mathbf{j} \cdot \sin\left(\frac{\alpha}{2}\right) \\ \cos\left(\frac{\alpha}{2}\right) \end{bmatrix} \quad (5.5)$$

then describes the rotation between the two frames.

Now that the z-axes of the  $\mathcal{I}$  and  $\mathcal{C}$  frame are aligned, the x- and y-axis can be aligned by a rotation  $\beta$  around the z-axis. This yaw correction could usually be achieved using the magnetometer measurements, but because those are heavily obscured indoors and especially in the parking garage [36], an other solution must be found. A possible solution to this problem is accelerating the car straight forward and then using the accelerometer to measure along which axis the acceleration occurred. Assuming the car tilt (pitch) during the acceleration is minimal, the acceleration is only being measured along the x- and y-axis. The resulting vector is being aligned with the forward axis of the car, such that  ${}_B\hat{\mathbf{a}} = {}_C\mathbf{x}$ , in the same way as before. Resulting in the rotation angle

$$\beta = \arccos({}_B\hat{\mathbf{a}} \cdot {}_C\hat{\mathbf{x}}) \quad (5.6)$$

and the quaternion

$${}_{\mathcal{C}}\mathbf{q} = \begin{bmatrix} {}_C\hat{\mathbf{z}} \cdot \sin\left(\frac{\beta}{2}\right) \\ \cos\left(\frac{\beta}{2}\right) \end{bmatrix}. \quad (5.7)$$

The two quaternions can then be concatenated (in reverse order) to get the final quaternion

$${}_{\mathcal{C}}^{\mathcal{I}}\mathbf{q} = {}_{\mathcal{C}}^{\mathcal{B}}\mathbf{q} \otimes {}_{\mathcal{B}}^{\mathcal{I}}\mathbf{q} \quad (5.8)$$

which transforms the measurements of the IMU to the car frame.

The quaternion is then converted into a rotation matrix, because it is faster.

Is this true? Was true when I tested it myself, but would've thought quat is faster

Finally the measurements  $\mathbf{A}$  can be transformed using

$${}_{\mathcal{C}}\mathbf{A} = {}_{\mathcal{C}}^{\mathcal{I}}\mathbf{M} \cdot {}_{\mathcal{I}}\mathbf{A} \quad (5.9)$$

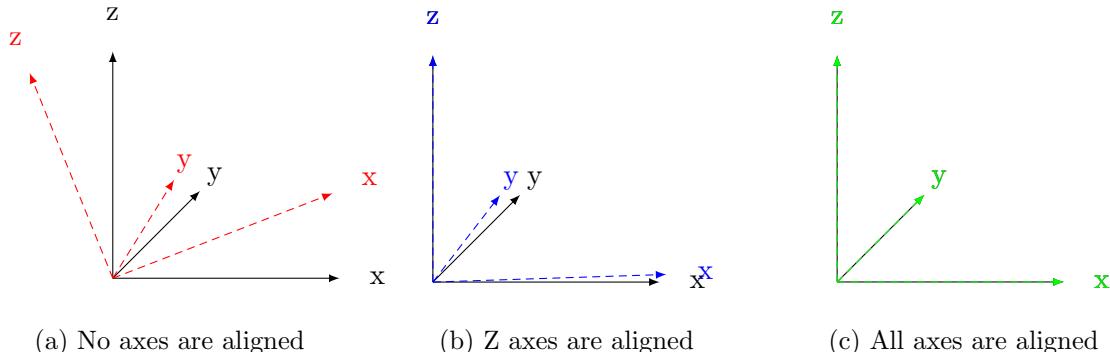


Figure 5.3: Frame transformation from the IMU frame  $\mathcal{I}$ , over the intermediate frame  $\mathcal{B}$  to the car frame  $\mathcal{C}$

### 5.1.2 Linear acceleration only

If the car stands still, the only measurable acceleration besides measurement errors is the acceleration due to gravity. When standing only flat ground, only the z-axis measures an acceleration. But when the car is tilted, e.g. on a ramp, the gravity is measured also by the x-axis, see fig. 5.4. The proportion of acceleration measured along the x-axis of the overall gravity can then be used to determine the pitch angle in the following way

$$\theta = \arcsin\left(\frac{a_x}{\|\mathbf{g}\|}\right) \quad (5.10)$$

where  $\mathbf{a}_x$  is the linear acceleration measured along the x-axis and  $\|\mathbf{g}\|$  the magnitude of the overall measured acceleration.

According to the definition, the angle is then zero if the car is parallel to the ground and  $90^\circ$  if the car would be pointing straight up. The angle is positive when driving up a ramp and negative if driving down. Disadvantages of this method are that the acceleration measurements are quite noisy and that no other accelerations are taken into account, e.g. the acceleration caused by the motor. A better approach, which incorporates the accelerations caused by the car, is described in 5.4.1.2.

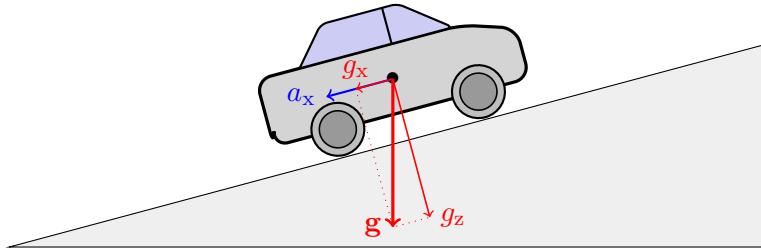


Figure 5.4: Gravity measured by IMU (in car frame).

### 5.1.3 Angular velocity only

To receive the rotation the angular velocity must be integrated with respect to time. To get the pitch angle the measurements along the y-axis are of interest.

The angular velocity is defined as

$$\omega = \frac{\Delta\theta}{\Delta t} \quad (5.11)$$

which means that the angular change between two times can be calculated with

$$\Delta\theta = \omega\Delta t. \quad (5.12)$$

The current angle can then be estimated with

$$\theta = \theta_0 + \int_0^t \omega_y dt \quad (5.13)$$

or with

$$\theta_k = \theta_0 + \sum_{i=1}^k \theta_i \quad (5.14)$$

in discrete time. Disadvantages of using only the angular velocity are that the estimations are not reliable over a long period of time. The random walk introduced by integrating white noise or a constant bias causes the estimation to drift away from the true value.

### 5.1.4 Complementary filter

The complementary uses both the linear acceleration and angular velocity measurements and tries to combine them, such that the good properties of each sensor are used to reduce the poor properties of the other. The linear acceleration is reliable in the long term, but is quite noisy. The angular velocity measurements on the other hand provide good short term accuracy, but should not be used for longer estimations due to drift. Therefore the complementary filter uses mostly the angular velocity, but corrects them with the linear acceleration data, to prevent drifting.

Get correct equation and distinguish between discrete and continuous

The angle estimation can be calculated with

$$\theta[k] = (1 - \alpha)(\theta[k - 1] + \theta_\omega) + \alpha(\theta_a). \quad (5.15)$$

with  $\alpha \in \mathbb{R} [0, 1]$  the mea

$$\theta_\omega = \quad (5.16)$$

The complementary filter uses the good short term accuracy of the gyroscope and combines it with the accelerometer data, which is good in the long run but not in the short term, to reduce the drifting. Hence the name complementary filter. The formula is as follows

$$\theta = (1 - \alpha)(\theta + gyrData \cdot dt) + \alpha(accData). \quad (5.17)$$

$$\theta[k] = (1 - \alpha)(\theta[k - 1] + gyrData \cdot dt) + \alpha(accData). \quad (5.18)$$

Whole text here is just copied from my notebook and will be rewritten

Where  $\theta$  denotes the estimated angle from fusing the measurements

- $\theta$ : Estimated angle from fusing measurements
- $\alpha$ : Time constant response time in the range of [0-1]. With a value of 0 only the gyroscope is being used and with  $\alpha = 1$  only the accelerometer.
- $gyrData$ : Angular velocity
- $accData$ : Angle from the low-pass filtered accelerometer data using  $accData = \arctan 2(a_x, a_z)$
- (in our case  $\arctan 2(a_x - a_{car}, a_z)$  to be precise)

A high value of  $\alpha$  means that the measurements of the gyroscope are trusted less than the from the accelerometer, and vice versa a small  $\alpha$  value means that the are gyroscope measurements are weighted more. A decrease of  $\alpha$  leads to a smoother signal, but at the cost of adding some additional time delay.

## 5.2 LiDAR only

Maybe brief overview of section/algorithm

Add pcl citation somewhere [7]

### 5.2.1 Calibration

Same as for the IMU, a transformation from the LiDAR frame to the car frame is necessary. The calibration is very similar to that of the IMU. At first both z-axes will be aligned. This is achieved by detecting the ground plane in the point cloud and finding the transform, such that the normal vector of the plane aligns with the z-axis of the car (the plane gets projected onto the xy-plane of the car frame). This results in the correct pitch and roll angle. The yaw angle can not be determined and is assumed to be to be zero, but can also be measured by hand and given as parameter.

For the ground plane detection Random Sample Consensus (RANSAC) [37] is used. RANSAC is a non-deterministic algorithm to remove outliers and is often used in computer vision. RANSAC can also be used for plane segmentation in 3D point clouds. Consider a point cloud with  $n$  points, where each point  $i$  has the coordinates  $x_i, y_i, z_i$ . In a first

step, three random points from the point cloud are selected. Three, because this is the minimum number of points needed for a plane. Now the parameters  $a, b, c, d$  of the plane equation

$$ax + by + cz + d = 0 \quad (5.19)$$

can be calculated. Then for every other point the deviation from the proposed plane can be calculated with

$$r = \frac{ax_i + by_i + cz_i + d}{\sqrt{a^2 + b^2 + c^2}} \quad (5.20)$$

and is then summed up. If the distance is within a certain threshold, the point counts as an inlier. After iterating through the whole point cloud, the number of inlier points and their coordinates are stored. This process is then repeated again until the maximal number of iterations are reached. The plane with the greatest number of inliers is then selected.

Then the normal vector of the plane, which can be conducted from the plane equation as follows

$$\mathbf{n} = (a \ b \ c)^\top \quad (5.21)$$

is projected onto the z-axis of the car. The necessary rotation is then applied to the detected plane. Now that a plane has been found it must be ensured, that it really is the ground plane. Typically either the ceiling, ground or a side wall gets detected with RANSAC. The greater the plane is (or the more points lie inside a plane), the more likely is the detection of the plane. Due to the mounting and Field Of View (FOV) of the LiDAR, the ceiling usually does have the most points and is thus detected in the first iteration.

An accidental ceiling detection can be prevented by looking at the average z-values of the detected plane. Because the lidar is mounted on the roof of the car, the z-values of the detected plane must be negative. If they are positive, the ceiling has been detected. Furthermore it is known, that the WHICH angle to rotate the lidar to level ground should not exceed the mount angle. If that is the case, most likely a side wall has been detected. If either condition has not been fulfilled, the detected plane does get removed from the point cloud and using RANSAC a new ground plane estimation is made and validated. This process gets repeated until both conditions are fulfilled. The yaw angle and the x- and y-translation from the LiDAR to the centered front of the car must be entered manually, but the pitch and roll angle and the distance from the LiDAR to the ground are used from the calibration.

### 5.2.2 Algorithm

Because the raw LiDAR data is too big to allow for real time processing, preprocessing is necessary. It consists of a passthrough filter to remove unwanted points (e.g. behind the car) and a voxel grid filter to downsample the point cloud. Before the passthrough filter can be applied, the point cloud must be transformed to the car frame. The in the previous section described calibration algorithm is performed once at the start and its returned rotation is then applied to every new measurement.

The passthrough filter then removes all the points which lie outside the specified x, y and z limits. Because the car drives forward, only points in front of the car are of interest. Furthermore the points further away than a certain threshold are neglected, because the resolution and accuracy of the measurements of the LiDAR decreases with increasing distance. The ceiling points are removed by limiting the points in z-direction. The exact values used for the passthrough filter can be seen along the other parameters in table 5.1. The next step in reducing the point cloud size is the voxel grid filter [38]. The point cloud is converted into a 3D grid consisting of small cubes called voxels. Each cube can contain multiple points or none, the size of the voxels (also known as leaf size) determines the

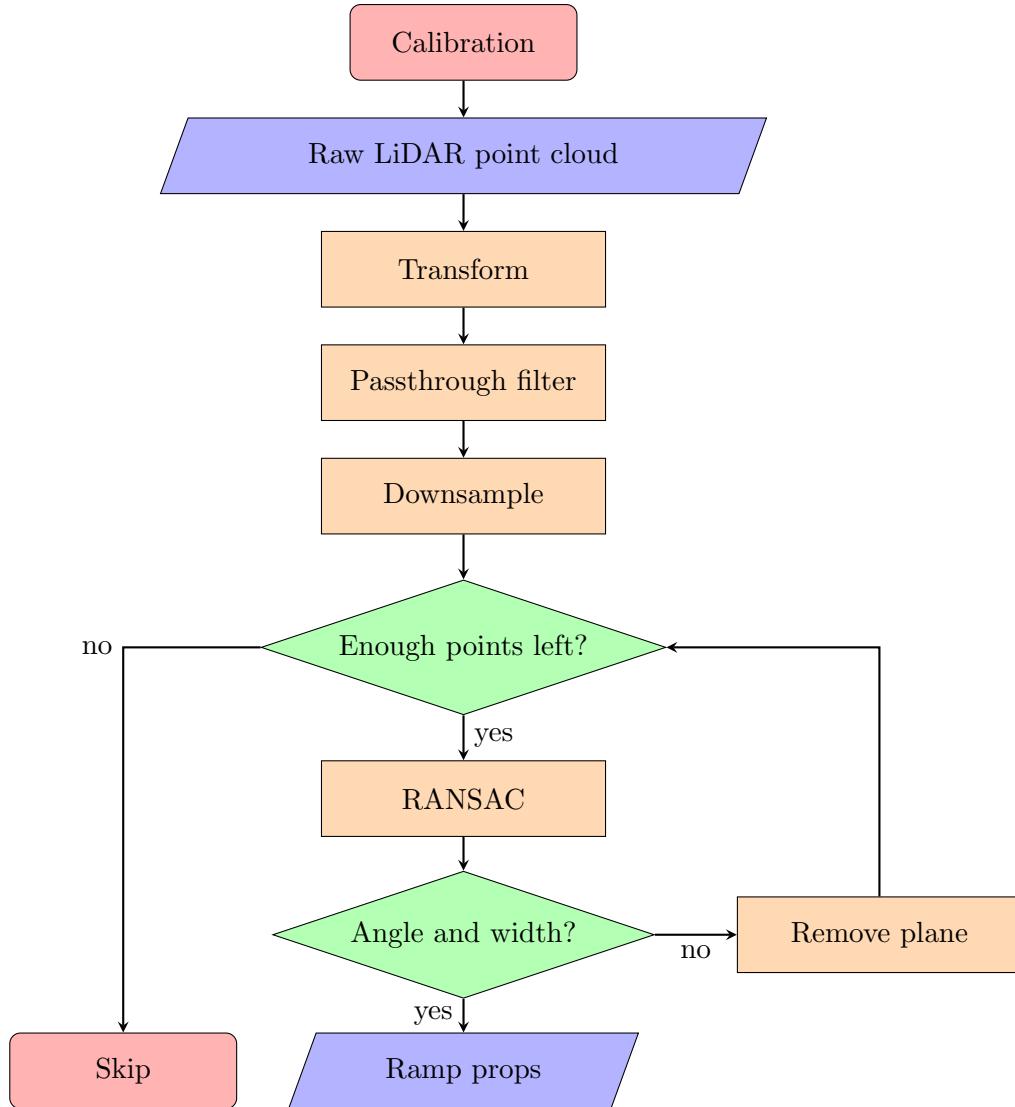


Figure 5.5: Algo for ramp detection

resolution. All the points inside a cube are then reduced to their most centroid point. If the cube does not contain any points, it is neglected.

Now that the point cloud size is reduced greatly the actual ramp detection can be performed with sufficient performance. The RANSAC algorithm usually detects the following types of planes: ceiling, ground, side wall or the desired ramp. RANSAC is applied iteratively until a plane of type ramp has been found. If a plane of different type has been found, it gets removed and the RANSAC algorithm is applied again. To prevent an infinite loop, the algorithm will exit after either a certain number of iterations has been performed, or if after the removal of a plane not enough points are left in the point cloud. The accidental detection of the ceiling was already prevented during the passthrough filter step, where the ceiling points have been removed from the point cloud. By limiting the maximal angle between the ground plane and the detected plane the side walls are ignored. Similarly, the detected plane gets classified as ground plane, if the angle between the ground plane and the detected plane is near zero (or below the specified minimum angle). Beside the angle, the width of the plane (y-range) is being calculated and compared to the parameters, to make sure that the plane is a drivable ramp for cars and not e.g. a small ramp for wheel chairs. The distance to the ramp is estimated by taking the  $n$  nearest points of the ramp plane and calculating their mean.

no idea if wheel chair ramps are common in parking garages?

What happens with stairs?

If a ramp has been detected, the estimated angle and distance from the car front to the beginning of the ramp are returned.

Add down ramp detection

An visual representation of the algorithm is depicted in fig. 5.5.

Add all algo parameters to table

Or is this not important?

In Anhang

Nicht alle parameter einzeln erklaeren sonder nur die wichtigsten

Table 5.1: Used parameters for lidar algo

parameter	value
<b>Passthrough filter</b>	
x	0 m to 30 m
y	-2 m to 2 m
z	-1 m to 2 m
<b>Voxel filter</b>	
leaf_size	0.1 m
<b>RANSAC</b>	
max_iter	100
distance_threshold	0.11 m
normal_distance_w	0.01
<b>rd</b>	
angle	3° to 9°
width	2 m to 6 m
o	4

### 5.3 Camera only

### 5.4 Sensor fusion

Brief explaination what sensor fusion is and why useful

#### 5.4.1 IMU and Odometer

##### 5.4.1.1 Car acceleration from odometer data

Because the odometer only delivers the speed of each wheel, the car velocity has to be calculated first. During turns the left and right wheels travel at different speeds, the wheel on the inner side of the turn travels slower, than the outer wheel. E.g. during a left turn, the left wheel moves slower than the right wheel. A simple yet sufficiently accurate model to calculate the car velocity from the wheel speeds is the linear single track model ("Einspurmodell") [39]. In this model both wheels on one axis are replaced with one wheel in the middle.

### Explain model

The linear assumption holds true for low lateral accelerations (up to  $4 \frac{\text{m}}{\text{s}}$ ), which will not be surpassed in the parking garage scenario. Using the assumptions from above, the car velocity  $v_{\text{car}}(t)$  can be calculated with

$$\alpha(t) = \frac{v_{\text{rl}}(t) - v_{\text{rr}}(t)}{d} \quad (5.22)$$

$$\gamma(t) = \frac{\alpha(t)}{f_{\text{odom}}} \quad (5.23)$$

$$v_{\text{car}}(t) = \frac{v_{\text{rl}}(t) + v_{\text{rr}}(t)}{2} \cdot \cos(\gamma) \quad (5.24)$$

with  $v_{\text{rl}}$  and  $v_{\text{rr}}$  being the wheel speeds of the rear right and rear left wheel respectively.

$\alpha$  is a helper variable ,  $d$  the track width! and  $\gamma$  is the yaw angle of the car.

The car's acceleration can be derived from the velocity using

$$a_{\text{car}}(t) = \frac{d}{dt} v_{\text{car}}. \quad (5.25)$$

But because all measurements are discrete, numerical differentiation e.g. forward difference must be used

$$a_{\text{car}}(h) = \frac{v_{\text{car}}(x+h) - v_{\text{car}}(x)}{h} \quad (5.26)$$

with  $h$  being the step size, which depends on the rate of the sensor.

#### 5.4.1.2 Gravity method

As described in ..., the linear acceleration measurements can be used to determine the pitch angle. But the estimation is only valid under the condition, that there are no accelerations other than the acceleration due to gravity. This condition is not necessarily true when the car is driving, during which the car can accelerate or break. To get the correct estimation the car's acceleration must be subtracted from the measurement. The car's acceleration can be calculated by deriving the wheel speed measurements with respect to time. It is important that both the IMU and odometer measurements are synchronized, otherwise bad. The prevailing accelerations can be seen in figure 5.6. When the car breaks, the direction of  $a_x$  inverts.

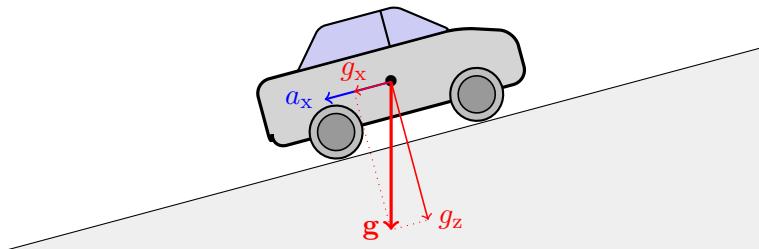


Figure 5.6: Existing accelerations, when the car is accelerating

$$\alpha = \arcsin \left( \frac{a_x - a_{at}}{g} \right) = \arcsin \left( \frac{a_x - \frac{dv}{dt}}{g} \right) \quad (5.27)$$

Text is bad

#### 5.4.2 IMU and odometer and LiDAR

Algo description

# Chapter 6

## Results

### 6.1 Evaluation concept

- The straight ramp on floor -2 will be used as main example
- All plots will be only made for that ramp
- Except maybe for special edge cases (e.g. influence of braking on imu estimation)
- All the other ramps will be only evaluated using scores, displayed in tables

### 6.2 Ramp metering? (IMU)

- How well do different Intertial Measurement Unit (IMU) methods work...
  - ... at ramp detection
  - ... at ramp distance measuring
  - ... at angle estimation
- Which methods should be compared?
  - Because gravity method only works with odometer, which is only available when driving down or half way up
  - Should new recording at different offset angles be made?
  - How to handle false positives, e.g. when braking on level ground?
  - What to display in the table, only normal drives or also ones with braking etc?
  - Or evaluate those edge cases separately? (Probably with plot of est angle)
  - How to handle offline vs live detection (for the offline detection the measurements can be butterworth filtered)?
  - How to determine the delay of the detection (using the camera image)?
  - What about the consistency test recordings, how or where to display the results?

Raw accelerometer data or gyroscope peak is ramp start

Also show consistency recordings (avg error and stuff)

### 6.3 Ramp detection (LiDAR and camera)

Table 6.1: Performance evaluation

Structure	Distance	Frames	TP	FP
<b>Straight (ascending)</b>				
uc	0 m to 5 m	62	100.00%	0.00%
uc	5 m to 10 m	62	100.00%	0.00%
uc	10 m to 15 m	59	100.00%	0.00%
uc	15 m to 20 m	61	97.92%	2.08%
uc	20 m to 25 m	61	97.83%	2.17%
uc	25 m to 30 m	59	42.75%	0.00%
us	0 m to 5 m	162	83.33%	6.67%
us	5 m to 10 m	149	83.33%	0.00%
us	10 m to 15 m	144	100.00%	0.00%
us	15 m to 20 m	150	87.41%	0.74%
us	20 m to 25 m	169	79.38%	0.69%
us	25 m to 30 m	78	43.42%	0.00%

- Confusion matrix or similar (false negatives could be hard (e.g. curve))
- Estimated angle and distance compare to ground truth
- Do downwards ramp work?
- Camera Light Detection And Ranging (LiDAR) projection for nice visualization
- Something about camera

Make new recordings of level -2 ramps, approaching at different (e.g. 0, 15, 30, 45) angles

- Should structure be only one ramp (and direction) or should it include multiple?
- How to choose distance interval?
- What about offset angle?
- What about estimated distance, angle, width and length?
- What is a true positive? (E.g. some detected points lie in region, or at least  $x\%$  lie in region, or less than  $y\%$  points lie outside...)
- Are number of frames important?
- I've done multiple recordings of e.g. straight ramp on -2 floor, but only one of the ones on floor -1, how to handle?

Anzahl Fahrten als Spalte

Subcategories rampentyp



Figure 6.1: Lidar points projected into the camera image. The green points were identified as part of a ramp.

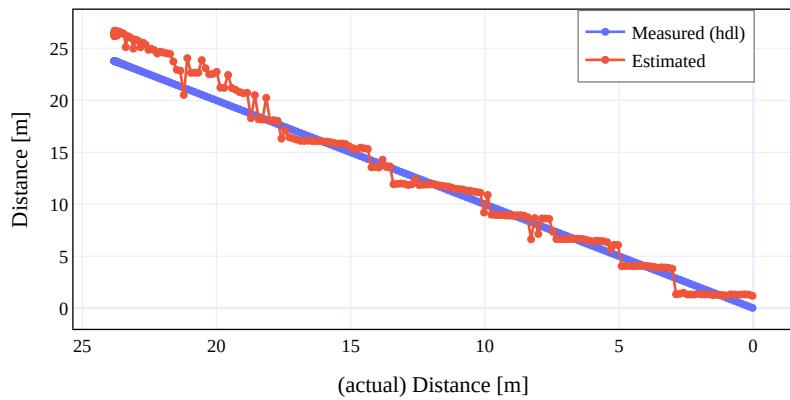


Figure 6.2: caption

# Chapter 7

## Conclusion

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

# Bibliography

- [1] O. J. Woodman, “An introduction to inertial navigation,” Research report 696, University of Cambridge, aug 2007.
- [2] Y. Li and J. Ibanez-Guzman, “Lidar for Autonomous Driving: The Principles, Challenges, and Trends for Automotive Lidar and Perception Systems,” *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 50–61, 2020.
- [3] Withrobot, “myAHRS+ User Manual,” 2017.
- [4] Stereolabs, “ZED2i User Manual,” 2019.
- [5] RoboSense, “RS-Bpearl User Manual,” 2020.
- [6] Velodyne, “VLP-32C User Manual,” 2018.
- [7] R. B. Rusu and S. Cousins, “3D is here: Point Cloud Library (PCL),” in *2011 IEEE International Conference on Robotics and Automation*, pp. 1–4, IEEE, may 2011.
- [8] A. SE, “A sudden bang when parking,” 2015.
- [9] H. Banzhaf, D. Nienhuser, S. Knoop, and J. Marius Zollner, “The future of parking: A survey on automated valet parking with an outlook on high density parking,” *IEEE Intelligent Vehicles Symposium, Proceedings*, no. Iv, pp. 1827–1834, 2017.
- [10] J. Jauch, J. Masino, T. Staiger, and F. Gauterin, “Road grade estimation with vehicle-based inertial measurement unit and orientation filter,” *IEEE Sensors Journal*, vol. 18, no. 2, pp. 781–789, 2018.
- [11] P. Sahlholm, H. Jansson, E. Kozica, and K. H. Johansson, “A SENSOR AND DATA FUSION ALGORITHM FOR ROAD GRADE ESTIMATION,” *IFAC Proceedings Volumes*, vol. 40, no. 10, pp. 55–62, 2007.
- [12] P. Sahlholm and K. Henrik Johansson, “Road grade estimation for look-ahead vehicle control using multiple measurement runs,” *Control Engineering Practice*, vol. 18, pp. 1328–1341, nov 2010.
- [13] K. Maleej, S. Kelouwani, Y. Dube, and K. Agbossou, “Event-Based Electric Vehicle Mass and Grade Estimation,” in *2014 IEEE Vehicle Power and Propulsion Conference (VPPC)*, pp. 1–6, IEEE, oct 2014.
- [14] N. Kidambi, R. L. Harne, Y. Fujii, G. M. Pietron, and K. W. Wang, “Methods in Vehicle Mass and Road Grade Estimation,” *SAE International Journal of Passenger Cars - Mechanical Systems*, vol. 7, pp. 981–991, apr 2014.
- [15] J. Ryu and J. C. Gerdes, “Integrating Inertial Sensors With Global Positioning System (GPS) for Vehicle Dynamics Control,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 126, pp. 243–254, jun 2004.

- [16] B. Yazdani Boroujeni and H. C. Frey, "Road grade quantification based on global positioning system data obtained from real-world vehicle fuel use and emissions measurements," *Atmospheric Environment*, vol. 85, pp. 179–186, mar 2014.
- [17] L. Mainetti, L. Patrono, and I. Sergi, "A survey on indoor positioning systems," in *2014 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pp. 111–120, IEEE, sep 2014.
- [18] B. Ozdenizci, V. Coskun, and K. Ok, "NFC Internal: An Indoor Navigation System," *Sensors*, vol. 15, pp. 7571–7595, mar 2015.
- [19] M. Nilsson and E. Öhlund, "Estimation of road inclination," 2012.
- [20] N. Palella, L. Colombo, F. Pisoni, G. Avellone, and J. Philippe, "Sensor fusion for land vehicle slope estimation," in *2016 DGON Intertial Sensors and Systems (ISS)*, pp. 1–20, IEEE, sep 2016.
- [21] C. Sentouh, S. Mammar, and S. Glaser, "Simultaneous vehicle state and road attributes estimation using unknown input proportional-integral observer," in *2008 IEEE Intelligent Vehicles Symposium*, pp. 690–696, IEEE, jun 2008.
- [22] W. He and J. Xi, "A Quaternion Unscented Kalman Filter for Road Grade Estimation," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1635–1640, 2020.
- [23] J. Wu, Z. Zhou, J. Chen, H. Fourati, and R. Li, "Fast Complementary Filter for Attitude Estimation Using Low-Cost MARG Sensors," *IEEE Sensors Journal*, vol. 16, pp. 6997–7007, sep 2016.
- [24] M. Nejati and B. D. Argall, "Automated incline detection for assistive powered wheelchairs," in *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 1007–1012, IEEE, aug 2016.
- [25] E. Lefferts, F. Markley, and M. Shuster, "Kalman Filtering for Spacecraft Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, vol. 5, pp. 417–429, sep 1982.
- [26] D. K. Shaeffer, "MEMS inertial sensors: A tutorial overview," *IEEE Communications Magazine*, vol. 51, no. 4, pp. 100–109, 2013.
- [27] M. Perlmutter and S. Breit, "The future of the MEMS inertial sensor performance, design and manufacturing," *2016 DGON Inertial Sensors and Systems, ISS 2016 - Proceedings*, pp. 1–12, 2016.
- [28] M. Kok and T. B. Schon, "Magnetometer Calibration Using Inertial Sensors," *IEEE Sensors Journal*, vol. 16, pp. 5679–5689, jul 2016.
- [29] Pengfei Guo, Haitao Qiu, Yunchun Yang, and Zhang Ren, "The soft iron and hard iron calibration method using extended kalman filter for attitude and heading reference system," in *2008 IEEE/ION Position, Location and Navigation Symposium*, pp. 1167–1174, IEEE, 2008.
- [30] P. Zhang, X. Zhan, X. Zhang, and L. Zheng, "Error characteristics analysis and calibration testing for MEMS IMU gyroscope," *Aerospace Systems*, vol. 2, pp. 97–104, dec 2019.
- [31] T. Fujii and T. Fukuchi, *Laser Remote Sensing*. CRC Press, 2005.

- [32] D. Wang, C. Watkins, and H. Xie, “MEMS Mirrors for LiDAR: A Review,” *Micro-machines*, vol. 11, p. 456, apr 2020.
- [33] K. Reif, *Automobilelektronik: eine Einführung für Ingenieure*. Springer-Verlag, 2014.
- [34] K. Reif, *Bosch Autoelektrik und Autoelektronik*. Springer-Verlag, 2011.
- [35] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Oote, J. Leibs, R. Wheeler, and A. Y. Ng, “ROS: an open-source Robot Operating System,” in *ICRA workshop on open source software*, vol. 3, p. 5, 2009.
- [36] B. Li, T. Gallagher, A. G. Dempster, and C. Rizos, “How feasible is the use of magnetic field alone for indoor positioning?,” in *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–9, IEEE, nov 2012.
- [37] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, pp. 381–395, jun 1981.
- [38] G. Vosselman, B. Gorte, G. Sithole, and T. Rabbani, “Recognising structure in laser scanner point clouds,” *International archives of photogrammetry, remote sensing and spatial information sciences*, vol. 46, no. 8, pp. 33–38, 2004.
- [39] M. Mitschke and H. Wallentowitz, *Dynamik der Kraftfahrzeuge*. Springer-Verlag, 2014.

# Chapter 8

## Appendix

Code extracts and extra plots etc

Table 8.1: Performance evaluation

Structure	Distance	Frames	TP	FP
uc	0 m to 5 m	62	100.00%	0.00%
uc	5 m to 10 m	62	100.00%	0.00%
uc	10 m to 15 m	59	100.00%	0.00%
uc	15 m to 20 m	61	97.92%	2.08%
uc	20 m to 25 m	61	97.83%	2.17%
uc	25 m to 30 m	59	42.75%	0.00%
us	0 m to 5 m	116	100.00%	0.00%
us	5 m to 10 m	117	100.00%	0.00%
us	10 m to 15 m	116	100.00%	0.00%
us	15 m to 20 m	123	100.00%	0.00%
us	20 m to 25 m	140	99.23%	0.00%
us	25 m to 30 m	50	59.78%	0.00%
us2	0 m to 5 m	21	100.00%	0.00%
us2	5 m to 10 m	23	100.00%	0.00%
us2	10 m to 15 m	28	100.00%	0.00%
us2	15 m to 20 m	27	37.04%	3.70%
us2	20 m to 25 m	29	0.00%	3.45%
us2	25 m to 30 m	28	10.71%	0.00%
us3	0 m to 5 m	25	0.00%	40.00%
us3	5 m to 10 m	9	0.00%	0.00%
us3	10 m to 15 m	0	nan%	nan%
us3	15 m to 20 m	0	nan%	nan%
us3	20 m to 25 m	0	nan%	nan%
us3	25 m to 30 m	0	nan%	nan%

Table 8.2: Used parameters for lidar algo

Parameter	Value	Unit
<b>Passthrough filter</b>		
x	0 to 30	m
y	-2 to 2	m
z	-1 to 2	m
<b>Voxel filter</b>		
leaf_size	0.1 m	m
<b>RANSAC</b>		
max_iter	100	m
distance_threshold	0.11 m	m
normal_distance_w	0.01	m
<b>rd</b>		
angle	3° to 9°	m
width	2 m to 6 m	m
<i>o</i>	4	m