Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institute for Dynamic Systems and Control
IDSC
Institut für Dynamische Systeme und Regelungstechnik

Patricia Sung

Localization and Mapping for Close Loop Autonomous Racing

Master's Thesis

Institute for Dynamic Systems and Control
Swiss Federal Institute of Technology (ETH) Zurich

Supervision

Dr. Andrea Carron
Prof. Dr. Melanie Zeilinger

January 2021

IDSC-MZ-AC-06_MT

Abstract

Autonomous driving is a topic that covers various fields, where intense research has been done in the past decade. Given the significant progress made so far, real-life testing is still labor intensive and financially challenging. Therefore, we present a race car scaled down by 1:10 and equipped with the ability to perform mapping and localization. The car can explore and then navigate itself through different unknown tracks using different controllers, due to the system's extensibility.

Keywords: Autonomous racing, Mapping, Localization, Track Generation, Control, Collaborative Robotic Systems.

Acknowledgment

Throughout the experimental implementation, presentations and the writing of this thesis, I have received a great deal of support.

I would first like to express my sincere gratitude to my supervisor, Dr. Andrea Carron, whose expertise was invaluable in guiding me through the problem formulation and methodology. You always encouraged me patiently and never hesitated to provide me with insightful feedback. It was a pleasure to work with you, as you not only pushed me to sharpen my engineering mind but also instilled in me personal growth.

I would like to acknowledge my colleagues at the CRS group for their selfless sharing of knowledge and advice. I would particularly like to single out Ben Tearle and Christian Küttel for providing me with critical suggestions that made the implementation possible in the end.

In addition, I would like to thank my parents for their wise counsel and sympathetic ear. I have always been giving you a hard time by making drastic changes to my career path, nevertheless, you always stand by my side.

A special thanks to my friends throughout college. When I switched to engineering in my sophomore year, you were the ones who made me fearless of all the challenges and unknowns lying ahead. I will never forget the voluntary help from all of you that brought me to the position I am today. Especially Chin-Yi, I'm grateful to have your support whenever I need stimulating discussions at many crucial steps of my life.

I would also like to thank my friends, Manuel, Xavi and Niclas, who made Zürich home to me. You supported me in various aspects and gave me the strength to pull through till the completion of my thesis.

Last but not least, I would like to thank Ermes, for his love, encouragement and for advising me in all my doubts. You've been my motivation throughout the past semester to overcome many obstacles. No words can express my gratitude and love for you.

Ringraziamenti

Durante l'intero periodo di prove sperimentali, presentazione e stesura della presente tesi, ho ricevuto un notevole supporto da parte di numerose persone che hanno contribuito a vario titolo al raggiungimento di questo importante traguardo.

Innanzitutto, vorrei esprimere la mia più sincera stima e gratitudine al mio supervisore, Dr. Andrea Carron, la cui straordinaria esperienza mi ha fornito le indicazioni necessarie per una corretta formulazione e metodologia di lavoro. Ho sempre ricevuto da Lei parole di incoraggiamento e preziosi consigli che mi hanno guidata per tutto il percorso di questa tesi. E' stato un sincero piacere aver avuto l'opportunità di poter lavorare con Lei. Per il suo importante contributo alla mia crescita professionale e personale le sarò sempre profondamente grata.

Vorrei ringraziare i miei colleghi del gruppo CRS per il loro aiuto e per i preziosi consigli forniti. Vorrei nello specifico ringraziare Ben Tearle e Christian Küttel per il loro fondamentale contributo all'implementazione finale del progetto.

Un profondo ringraziamento alla mia famiglia, per i saggi consigli e il costante ascolto ricevuto. Sono consapevole di avervi fatto passare momenti agitati a causa delle drastiche scelte spesso intraprese durante la mia vita. Il vostro supporto costante è stato per me di straordinaria importanza.

Uno speciale ringraziamento ai miei amici di Universtà. Quando scelsi la facoltà di ingegneria al mio secondo anno accademico, voi siete stati coloro che mi hanno fatto forza nonostante tutte le sfide che si prospettavano all'orizzonte. Non dimenticherò mai il vostro aiuto che mi ha permesso di essere oggi la persona che sono. Chin-Yi, sono così grata di aver avuto la tua comprensione e supporto ogni qualvolta ho dovuto prendere importanti decisioni nella mia vita.

Vorrei inoltre ringraziare i miei amici Manuel, Xavi e Niclas, con i quali ho potuto risiedere a Zurigo sentendomi come a casa. Voi mi avete aiutata in numerose circostanze, dandomi la forza di proseguire con determinazione il mio percorso di tesi.

Infine vorrei ringraziare Ermes, per il suo amore, incoraggimento e per avermi saggiamente consigliata in tutte le occasioni di difficoltà. Sei stato la mia motivazione per tutto lo scorso semestre, la persona grazie alla quale sono riuscita a superare numerosi ostacoli. Nessuna parola potrà mai esprimere tutta la mia gratitudine e il mio amore per te.

Contents

Chapter 1

Introduction

1.1 Motivation and Background

What distinguishes autonomous racing from autonomous driving is the focus on challenging hardware and software limits, rather than exploring human factors or real life scenarios. In many autonomous races, the track is not revealed to participants until shortly before the start of the race, making navigation of racecars one of the first important tasks to deal with.

Navigation can be divided into SLAM and localization, which are also the main focuses in this thesis. The first aims at creating a map, while the later locates the car according to a known map and sensor readings. Among many different ways to achieve either tasks, we look for inexpensive methods that don't require complicated a setup or calibration. On top of introducing a fully autonomous racecar, it is also interesting to challenge controllers with indirect measurements and limited computational power.

1.2 Contribution

In summary, the main contributions of this thesis are:

- an autonomous racecar with hardware designed, assembled and mounted on the chassis,
- a fully autonomous pipeline, where real time perception, state estimation, navigation and control are all performed independent of external computational resources,
- experimental results' evaluation and demonstration of each subsystem,
- a potential extension of the localization part of Collaborative Robotic Systems (CRS).

To give a more detailed illustration on the last item, CRS is very well developed for the deployment and testing of different control algorithms in both simulation and on the physical system. For the sake of robustness, the Vicon system is used to obtain full pose estimation of the car. Although being highly accurate, it is expensive and labor intensive to set up. Furthermore, once the setup is done, the track is confined to a limited space. As part of the scope covered in this thesis, we implement localization and mapping methods that allow the car to deal with unknown tracks at arbitrary locations. The localization part potentially extends the autonomous level of CRS to become independent of external localization measurement and method.

1.3 Thesis Structure

In chapter 2, we introduce research relevant to this thesis. Selected hardware as well as how the system is identified in order to complete our implementation is covered in chapter 3. A high level overview of the software is given in chapter 4, whereas algorithms are explained in further details in chapter 5 and chapter 6. Experimental results are presented and discussed in chapter 7.

Chapter 2

Related Work

A wealth of research has been done regarding the topic indoor navigation. Various methods can be implemented ranging from sensing, mapping, path planning and localization. As presented by Federico et al. [1], maps and paths can be created in a hand-drawn manner. After manually sketching up the environment on a tablet and assigning the robot a path to follow, its state is updated via Monte Carlo Localization [2].

For laser scan-based sensing methods, which is also the focus of this thesis, scan-matching is a crucial process of finding the relative pose or transform between two positions where the scans were taken. Iterative Closest Point (ICP) [3] is one of the pioneering scan-matching algorithms, but falls short because of its exhaustive search for point correspondences at each iteration. Polar Scan Matching (PSM) [4] takes advantage of its polar coordinate system, which is naturally obtained by laser scans to estimate their correspondences. However, the scan matcher requires a preprocessing step. Gmapping [5] is an open source indoor planer SLAM algorithm, that includes front and backend systems and uses Rao-Blackwellized particle filters. SLAM frontends estimate the robot position in real-time, whereas backends optimize the pose graph according to previously generated poses.

2.1 F1TENTH Competition

F1TENTH [6] is an autonomous racing organization founded in Pennsylvania. Their goal is to not only bring racing enthusiasts together, but also provide educational opportunities to further stimulate engineering minds. In order to encourage participants to focus on developing their own algorithms, testbeds were provided to set a limit on hardware specifications. The platform is open-source and versatile enough to be used for various research topics, including but not limited to control, reinforcement learning, communication systems.

An interesting and well documented work was done by Daniel Ameida [7], where the thesis focuses on the implementation and tuning of an F1TENTH testbed. While cooperating with Nuno Guedes [8], an Intelligent Transportation System (ITS) for car platooning was introduced, demonstrating the abundance of topics F1TENTH not only covers but also extends to.

The race largely inspired the setup of this thesis. Similar hardware selection, track definition as well as suggested software configuration are used in this work. Further details regarding the setup will be discussed in chapter 3.

2.2 AMZ Racecar

The Formula Student competition has a higher complexity in terms of scale and track definition compared to F1TENTH. In the main race, autocross, instead of walls that mark the borders of the track, cones of different colors are used to distinguish the inner and outer borders of the track. This made it important for the AMZ racing team [9] to incorporate image recognition and 3D point cloud processing into the perception pipeline.

The localization and mapping part of their work is done by particle filter-based methods, namely Monte Carlo Localization [2] and fastSLAM [10]. FastSLAM not only provides efficient computational performance but also has the advantage of dealing with sensor failures, since additionally to cameras and LiDARs, the AMZ racecar is using IMU, GPS, ground speed sensor, motor torque, steering and wheel speed sensors.

As presented in chapter 4, this thesis takes inspiration from the AMZ racecar's high level structure. In additon, throughout the course of this thesis, similar off-line development methods are also used to enhance efficiency.

2.3 CRS Framework

CRS framework was first introduced in a semester project at the Intelligent Control Systems group [11]. It is initiated for the realization of the MPCC (Model Predictive Contouring Control) controller with a kinematic model in experimental setup. Additionally, a PID and a lane keeping feedforward controller are developed to mainly serve as comparisons with the MPCC. Miniature cars at 1:27 scale are used, thus, instead of mounting hardware on the car, computations are done on an external PC. The modularity and extensibility of the system makes it possible to share controllers and state estimators with different hardware configurations, including our pipeline. Furthermore, this thesis verifies its potential to be further integrated into the CRS framework.

Chapter 3

Hardware

In this chapter, each hardware component is introduced in detail. This includes an introduction of the experimental setup, an overview of their connection with each other, and the result of system identification.

3.1 Track Setup

The track is set up in a 4x3 square meter space using puzzle floor mat, which can be easily adjusted for testing various track shapes. It is positioned in the CRS arena in order to conveniently obtain the groundtruth using the Vicon system. However, any space should be possible for running the system as long as the walls are well closed at the LiDAR's scan height. Limits regarding the shape and size of the track will be further discussed in chapter 7.

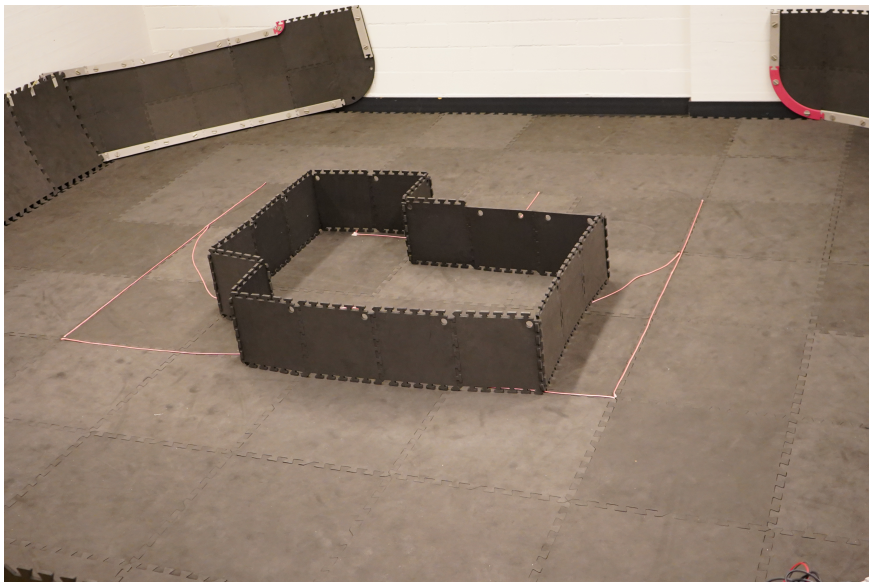


Figure 3.1: Experimental Setup in the CRS arena.

3.2 Hardware Design

All parts of the car are done through laser cutting, because of its fast and precise machining quality. In order to follow the overall development of the car, different versions of hardware configurations have been created over the course. Figure 3.3 and figure 3.2 show the final design.

The upper plate is not extended to the same length as the lower plate, because the computer is mounted on the lower plate. A power jumper is needed to switch between USB power supply and battery. We thus prefer to not cover it from the top. Besides, the heat sink can warm up to 48.5 degrees, so leaving more empty space around it is desired.

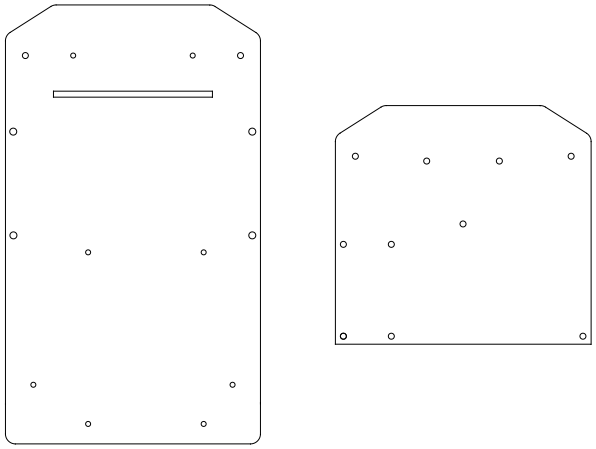


Figure 3.2: Left to right are the CADs for lower plate and upper plate.

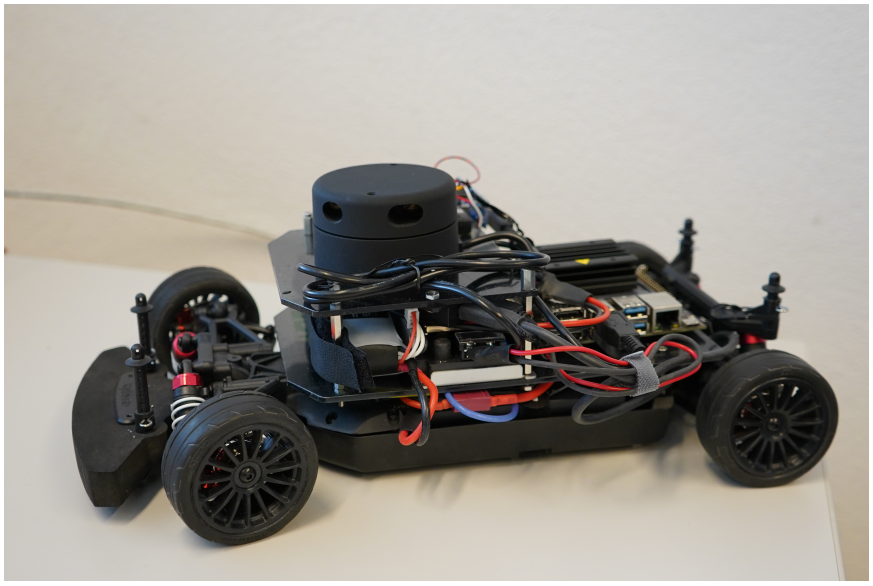


Figure 3.3: Final image of the car fully assembled.

3.3 Hardware Selection

Hardware are selected bearing the goal of prototyping a fully autonomous racecar with minimum budget and specifications in mind. This section presents a schematic of how different hardware components are connected to each other physically. Then, more detailed description is given for better understanding their capabilities and our considerations for picking them.

Electronic Schematic

All hardware mounted and assembled on the chassis are presented in figure 3.4. Hardware included in the grey block are responsible for computation, i.e. mapping, localization and control. Sensor and receiver are included in the blue block. Actuators are included in the orange block. Note that an adapter is used between the LiDAR and Computer, therefore the communication protocol at both ends are different.

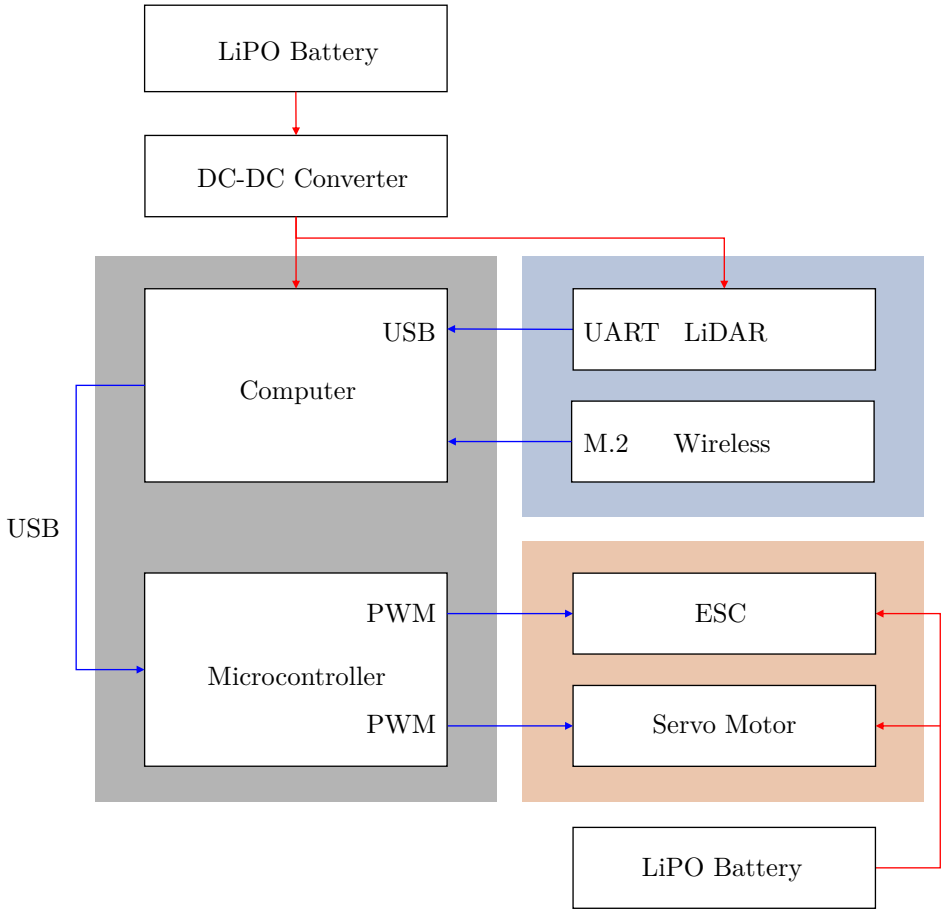


Figure 3.4: Red arrows indicate power connection. Blue arrows are data transmission.

KYOSHO FZ02 Chassis

Kyosho offers a powerful shaft driven four wheel drive chassis powered by 550 size 14T high torque motor. It features oil-filled shocks, grease-filled front and rear gear differentials and performance independent suspension. Full ball bearings are provided to reduce drive loss throughout the drive system.

Figure 3.5: 1:10 scale Acura NSX GT3 from Kyosho and its joystick.

NVIDIA Jetson Nano

According to the F1TENTH competition guidelines, NVIDIA Jetson TX2 was suggested because its computational power properly meets the needs of the scope of their race. MIT RACECAR suggests the Jetson TX1, which has a computational power lying between TX2 and Nano. Similar to TX2 and TX1, Jetson Nano also runs Linux OS, but is slightly downscaled in terms of most specification. However, it is more affordable, portable, and has 4 USB ports, which reduces the need for an external USB hub. The development board also offers various peripherals that are compatible out of the box. Further specification can be found in the NVIDIA Jetson Nano datasheet.

This board is used to process all information from the environment, then send out commands accordingly. In addition, wireless connection is established through SSH communication between the board and a PC. The Intel Wifi 8265NGW is selected to fit our computer's architecture. Throughout the development, it is beneficial to do so for monitoring the car's status and flexibly starting or stopping operations. When it comes to races, startup messages are usually sent in a similar fashion.

YDLIDAR G2

LiDAR, which stands for Light Detection and Ranging, is an accurate and fast remote sensor that generates precise information from the shape of its surroundings. It also operates well under different lighting conditions, due to its active illumination sensor. A LiDAR's price can cover a wide range where the high-end ones are thousand times more expensive than the basic ones. In many robotics applications, LiDAR is used for mapping, localization and obstacle detection.

The YDLIDAR G2 model is a planar LiDAR, meaning that it only has 1 beam that scans at a fixed level and gives us a 2D representation of the environment. According to the datasheet, this LiDAR uses UART communication and comes with a ROS compatible driver, that makes implementation much easier and efficient.

Arduino Nano

The Arduino Nano is a compact microcontroller that supports USB type B connection and features an 8-bit ATmega328 controller with an AVR architecture. The rich amount of libraries and the board's ability to interface with other controllers also made development on Arduino boards an affordable yet competitive choice. In this project, it is used to drive the electronic speed control (ESC) that comes with the chassis using pulse width modulation (PWM) signals, which will be further explained in the next section.

3.4 Communication with Chassis

From Computer to Microcontroller

Serial communication is established by sending steering and torque commands from the computer to the microcontroller via USB. The information is being encoded by adding a letter as an indicator before each value to determine whether they belong to steering or torque. This is later on being decoded on the microcontroller's side and propagated as desired PWM commands for the servo motor and ESC.

From Microcontroller to Chassis

In manual mode, the Kyosho chassis is controlled by its joystick through radio signals as shown in figure 3.5. Two channels of the radio signal receiver are used for steering and torque respectively.

To make the car fully autonomous, the ESC should receive PWM signals from the microcontroller, instead of the RC receiver. The wiring can be found in figure 3.4.

Identification of the correct frequency and duty cycle for both steering and torque is done through connecting the output channels of the receiver to an oscilloscope. Results are shown in figure 3.6 and figure 3.7. Steering is controlled by a servo motor, which is compatible with the arduino library, `ServoTimer2`. Although figure 3.7 identifies the frequency at 60.23 Hz, experimental results have shown that 50 Hz also works well with the system. The reason for using a different timer is to avoid conflict with the torque PWM's interrupts. Torque input has to go through the ESC, which takes PWM signals at 60.23 Hz and high resolution duty cycle between 10% and 20%.

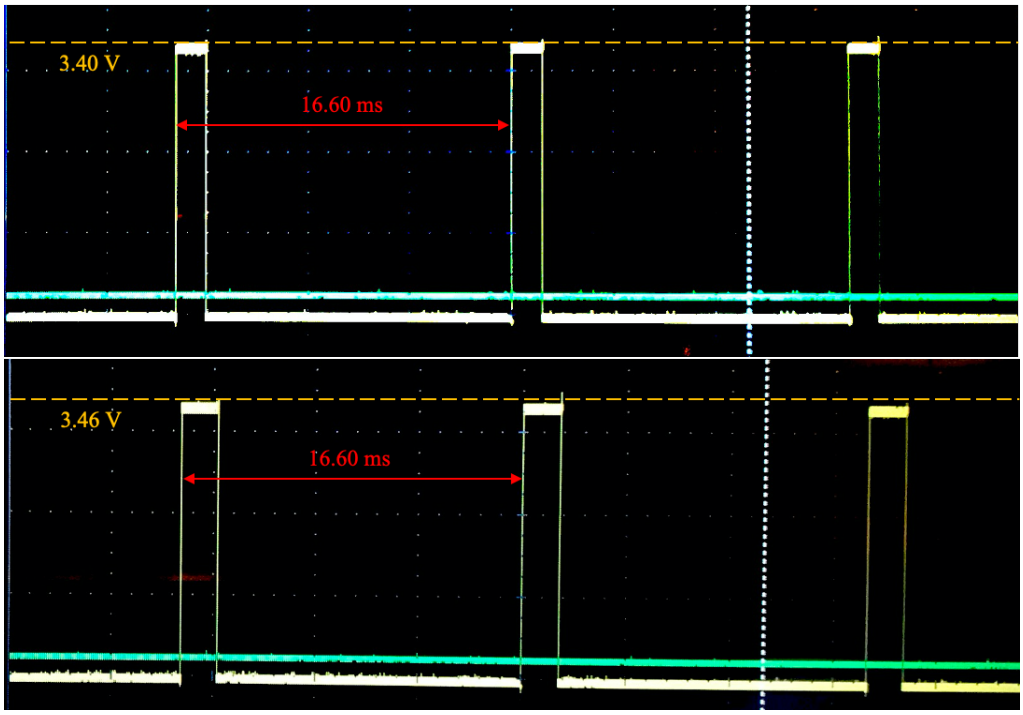


Figure 3.6: Torque PWM identification.

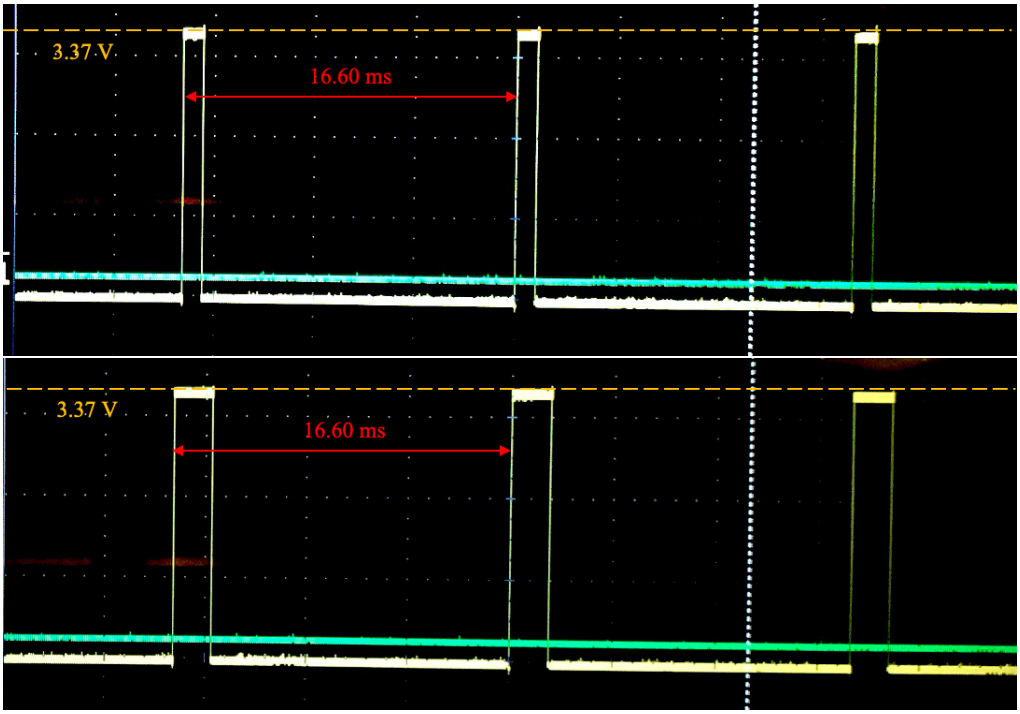


Figure 3.7: Servo PWM identification.

Chapter 4

Architecture Overview

From a high-level system point of view, the car runs on three different modes sequentially: Simultaneous Localization and Mapping (SLAM), track generation and localization. In localization mode, controllers also run in parallel, but in this chapter we stick with this name for not losing our focus and keeping a sense of brevity. Each mode will be introduced in detail in chapter 5, while this chapter rather focuses on the interaction between them.

SLAM

The goal of SLAM mode is to create a map that represents the track. While running in SLAM mode, a wall follow PID controller drives the car slowly along the outer side of the track to let the LiDAR fully scan around the map. Since SLAM is performing localization in parallel, the first return to the origin point serves as a switch to indicate our completion of the first lap. At this point, further map acquisition will not increase the map quality significantly, thus the car stops and switches to track generation mode.

Track Generation

Controllers need references to follow, therefore, a rough middle line is generated along the track using a computer vision based method. The track needs to be further polished due to its inconsistent density and noisiness, by fitting a spline over the rough track. Lastly, the curvature and tangent angle of each point on the track are calculated to provide controllers with more sophisticated information. Once these steps have been completed, the track is saved and the car moves on to the localization mode.

Localization

Three inputs are needed to complete this part, which are map, measurement and motion estimation. Thus, the localization mode starts with a range flow-based odometry estimation. Once a belief of the car's position is obtained using a particle filter-based localization algorithm, the result is further processed by a low pass filter to get a full state estimation of the car. In the end, two different controllers are implemented for closing the loop.

Chapter 5

Localization and Mapping

This chapter presents the algorithms that achieve SLAM and localization. A wall follow PID controller is introduced alongside with the former one. As for the later one, we focus only on localization, leaving the control part to chapter 6.

5.1 SLAM

One of the most important abilities of fully autonomous racing is the model acquisition of the environment to develop operation between the car and the real world race track. The F1TENTH competition defines the map using walls at LiDAR scan height level, therefore, a grid-based map representation is preferred over a feature-based one. As the name suggests, SLAM localizes the car while generating a map simultaneously. It is often considered a "chicken and egg" problem, since the knowledge of a map is required to perform localization. However, mapping needs position input for correct scan matching as well. Bearing the goal of fully autonomous driving in mind, a wall follow PID controller is designed to carefully guide the car through the entire track without the need of any map or localization data.

5.1.1 Hector SLAM

Hector SLAM [12] is a scalable and flexible algorithm for solving the SLAM problem using LiDAR measurements, which are processed as 2D point cloud data. Due to the low consumption of computational resources, it's been proven successfully on many research projects and is also the suggested algorithm in the F1TENTH guidelines.

The Map

Occupancy grid map is used to represent the track, where each grid at map coordinate P_m is assigned a value M to indicate its likelihood of being occupied. To overcome the discrete nature of such a map, bilinear filtering is beneficial for the estimation of both occupancy probabilities and derivatives.

Figure 5.1 presents four coordinates $P_{00\ldots11}$ used to approximate P_m. The interpolation results in

$$\begin{aligned} M(P_m) \approx & \frac{y - y_0}{y_1 - y_0} \left(\frac{x - x_0}{x_1 - x_0} M(P_{11}) + \frac{x_1 - x}{x_1 - x_0} M(P_{01}) \right) \\ & + \frac{y_1 - y}{y_1 - y_0} \left(\frac{x - x_0}{x_1 - x_0} M(P_{10}) + \frac{x_1 - x}{x_1 - x_0} M(P_{00}) \right), \end{aligned} \tag{5.1}$$

and the derivatives' approximation

$$\begin{aligned}\frac{\partial M}{\partial x}(P_m) &\approx \frac{y-y_0}{y_1-y_0}\left(M(P_{11})-M(P_{01})\right)+\frac{y_1-y}{y_1-y_0}\left(M(P_{10})-M(P_{00})\right)\\ \frac{\partial M}{\partial y}(P_m) &\approx \frac{x-x_0}{x_1-x_0}\left(M(P_{11})-M(P_{10})\right)+\frac{x_1-x}{x_1-x_0}\left(M(P_{01})-M(P_{00})\right).\end{aligned} \tag{5.2}$$

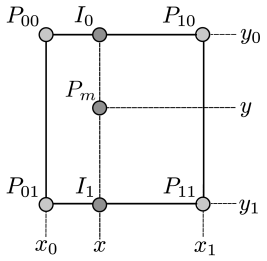


Figure 5.1: Bilinear interpolation of the occupancy values. [12]

Scan Matching

Scan matching is the optimization process where new laser scans are aligned with a previously created map as the car moves in a certain direction that further explores the track. It is even proven that LiDARs have higher accuracy than odometry data, given their high scan rate and low measurement noise. As a result, in the following method, only laser scans are taken into account for the process.

In order to minimize the alignment error between map and laser scan, a rigid body transformation $\xi = (p_x, p_y, \psi)^T$ for a specific scan i is found by minimizing

$$\xi^* = \arg\min_x \sum_{i=1}^{n} \left[1 - M(S_i(\xi))\right]^2, \tag{5.3}$$

where $M(S_i(\xi)) \in [0,1]$ is the map occupancy value, and $S_i(\xi)$ is the scanned coordinates $(s_{i,x}, s_{i,y})^T$ expressed in the world frame as

$$S_i(\xi) = \begin{pmatrix}\cos(\psi) & -\sin(\psi)\\ \sin(\psi) & \cos(\psi)\end{pmatrix}\begin{pmatrix}s_{i,x}\\ s_{i,y}\end{pmatrix}+\begin{pmatrix}p_x\\ p_y\end{pmatrix}. \tag{5.4}$$

Assuming the optimization starts at estimate ξ, $\Delta\xi$ yields a step towards a minimum error according to our goal

$$\sum_{i=1}^{n}\left[1 - M(S_i(\xi+\Delta\xi))\right]^2 \longrightarrow 0. \tag{5.5}$$

The optimization step

$$\Delta\xi = \mathbf{H^{-1}}\sum_{i=1}^{n}\left[\nabla M(S_i(\xi))\frac{\partial S_i(\xi)}{\partial \xi}\right]^T\left[1 - M(S_i(\xi))\right] \tag{5.6}$$

is the so-called Gauss-Newton equation [13], where the hessian is

$$\mathbf{H} = \sum_{i=1}^{n} \left[\nabla M(S_i(\xi)) \frac{\partial S_i(\xi)}{\partial \xi} \right]^T \left[\nabla M(S_i(\xi)) \frac{\partial S_i(\xi)}{\partial \xi} \right].$$

Due to the similarity of this approach to that of a gradient descent, it is vulnerable to getting stuck in local minima. A way to avoid this situation is to use smoother cost function, that is, a coarser map to initiate the alignment process, followed by finer maps generated independently.

5.1.2 Wall Follow PID Controller

The distances of interest of this controller are defined as $dist_{left}$, $dist_{right}$, $dist_{front}$ and are depicted in figure 5.2. If a complete round of scan consists of K ranges, then sampling a range of k scans gives an angle of k*360/K degrees of data. These data are being averaged, then taken as the true distance to those specific directions.

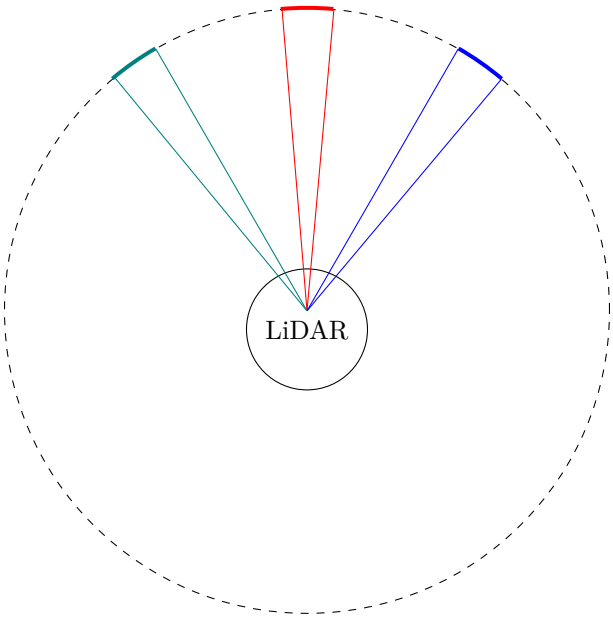
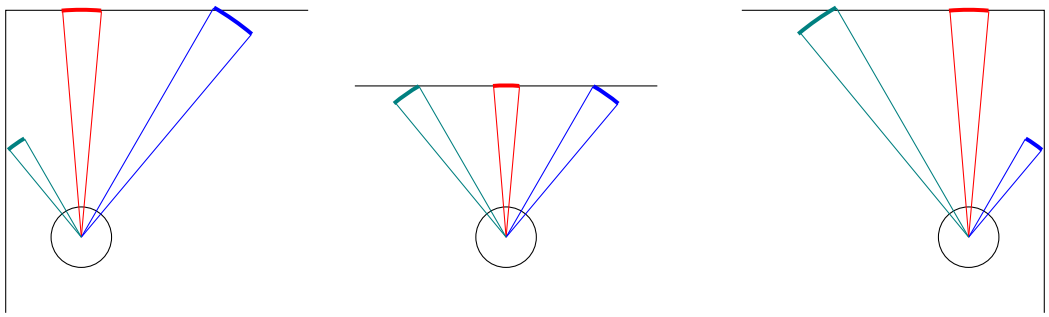


Figure 5.2: Signal filtering of three distances. Dashed circle is a complete range of scan. Green, red and blue ranges are distances of interest, $dist_{left}$, $dist_{front}$ and $dist_{right}$ respectively.

A PID controller is implemented that takes several corner cases into account. The corner cases help meeting the requirement of completing a lap smoothly enough for the map to be created without distortion. Three corner cases are illustrated in figure 5.3, where the torque is reduced when the car gets too close to the wall from different directions. Otherwise, the PID controller keeps the car at a fixed distance to the wall with its upper left or upper right scan.



(a) $dist_{left} < dist_{right}$ (b) $dist_{front} < dist_{stop}$ (c) $dist_{left} > dist_{right}$

Figure 5.3: Illustration of three corner cases with torque reduction due the proximity to the wall. In cases (a) and (c), full right turn and full left turn are made. Case (b) stops the car regardless of any other condition.

5.2 Localization

Adaptive Monte Carlo Localization (AMCL) is a particle filter-based localization algorithm for 2D navigation. Many algorithms used in AMCL are taken from the book Probabilistic Robotics [14]. It has several advantages over an Extended Kalman Filter (EKF), such as the independence to Gaussian noise and the direct use of LiDAR measurements. Compared to its former version, Monte Carlo Localization [2], AMCL effectively reduces computational power by downsizing the size of its particle cloud at the resampling stage. Furthermore, the addition of random particles enables it to deal with robot kidnapping or global localization failures.

One of the essential inputs for any particle filter-based localization method is the motion model. Although a Range Flow-Based 2D Odometry (RF2O) [15] is chosen in this thesis, many other motion models can be obtained through various sensors e.g. wheel encoders, control inputs, IMU or GPS. RF2O uses planar laser scans as its sole input to achieve precise velocity estimation. How an accurate motion model is incorporated for the update of particles in the particle filter will be further explained in section 5.2.2.

5.2.1 Measurement Model

Raw laser scans are noisy and subject to failures, therefore a conditional probability density

$$p(z_t|x_t, m) = \prod_{k=1}^{K} p(z_t^k|x_t, m) \tag{5.7}$$

estimates the likelihood of sensor measurement to be z_t at time t, given the sensor's position x_t and a known map m. $z_t = \{z_t^1, ..., z_t^K\}$ denotes the total number of values within a measurement, which in our case is a complete scan that includes all K ranges. Note that (5.7) assumes independence between measurement values k. Non-deterministic sensor readings are processed this way to be suitable for performing localization. Therefore, four types of uncertainties shown in figure 5.4 are taken into account for our measurement model:

a. a Gaussian noise around the true range z_t^{k*} that is expected due to local measurement errors,

b. an exponential distribution induced by unexpected nearby objects closer than the actual range e.g. people walking around the car or components on the car itself,

c. a uniform distribution at maximum scan range due to sensor failure, which often defaults the reading to the maximum range value,

d. a uniform distribution along the entire range due to unexplained factors.

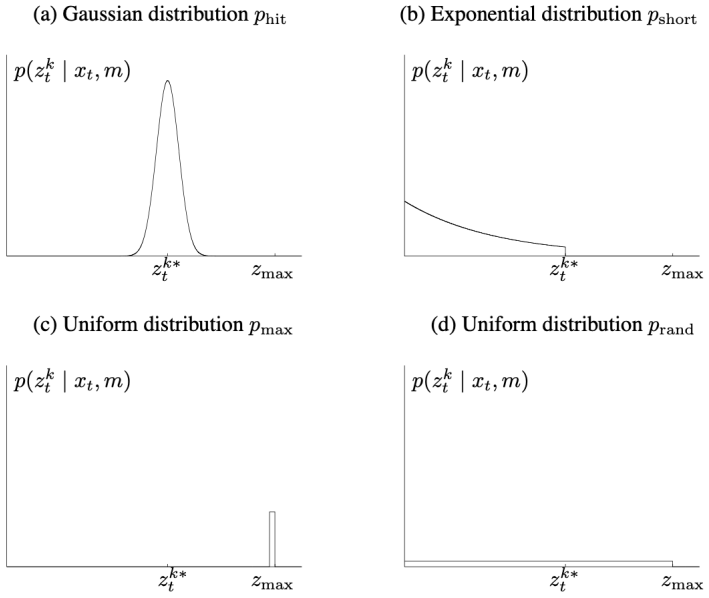


Figure 5.4: Four types of uncertainties. In each diagram, the horizontal axis refers to the sensor's measurement, and the vertical axis refers to each uncertainty model's likelihood. [14]

The aforementioned uncertainty distributions are incorporated into one likelihood using different weights $z_{hit...rand}$, such that

$$p(z_t^k|x_t, m) = \begin{pmatrix} z_{hit} \\ z_{short} \\ z_{max} \\ z_{rand} \end{pmatrix}^T \cdot \begin{pmatrix} p_{hit}(z_t^k|x_t, m) \\ p_{short}(z_t^k|x_t, m) \\ p_{max}(z_t^k|x_t, m) \\ p_{rand}(z_t^k|x_t, m) \end{pmatrix}, \tag{5.8}$$

and that $p(z_t|x_t, m)$ is obtained after iterating through all ranges K as done in (5.7).

5.2.2 Motion Model - Odometry

As an important component of localization, RF2O estimates the planar motion relying only on LiDAR measurements. Assuming a static and rigid environment, the 2D velocity of a laser scanner is estimated by finding the range flow constraint equation [16].

Let $R(t, \alpha)$ be a complete scan dependent on time t and scan coordinate $\alpha \in [0, K)$. In figure 5.5, point P on rigid body is detected at time t. The range of point P at second scan P' can be expressed in terms of a Taylor expansion

$$R(t + \Delta t, \alpha + \Delta\alpha) = R(t, \alpha) + \frac{\partial R}{\partial t}(t, \alpha)\Delta t + \frac{\partial R}{\partial \alpha}(t, \alpha)\Delta\alpha + O(\Delta t^2, \Delta\alpha^2), \tag{5.9}$$

where Δt is the time difference between two consecutive scans. Keeping the first order term and dividing by Δt yields and change of scan within time interval $[t, t + \Delta t]$:

$$\begin{aligned} \frac{\Delta R}{\Delta t} &\approx \frac{\partial R}{\partial t}(t, \alpha) + \frac{\partial R}{\partial \alpha}(t, \alpha)\frac{\Delta \alpha}{\Delta t} \\ &= R_t + R_\alpha \frac{\Delta \alpha}{\Delta t}, \end{aligned} \tag{5.10}$$

where $\Delta R = R(t + \Delta t, \alpha + \Delta\alpha) - R(t, \alpha)$.

Finally, by substituting the average velocity expression into (5.10) such that $\dot{r} = \Delta R/\Delta t$ and $\dot{\alpha} = \Delta\alpha/\Delta t$, the range flow constraint equation becomes

$$\dot{r} \approx R_t + R_\alpha \dot{\alpha}. \tag{5.11}$$

Knowing how point P moves to P' within time Δt is equivalent to knowing how the laser scanner moves respective to the environment.

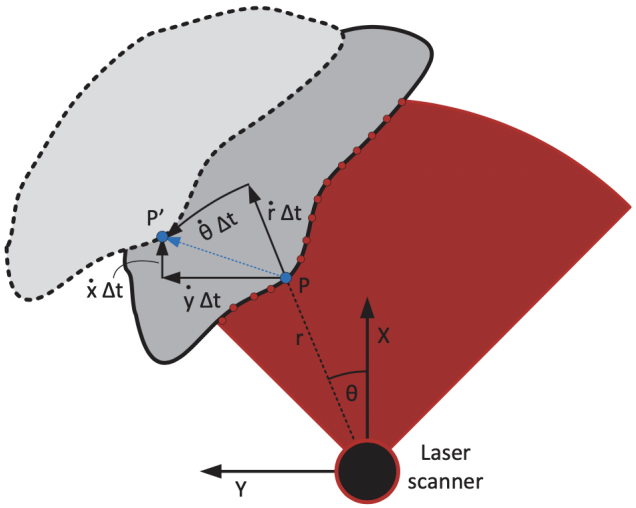


Figure 5.5: Rigid object moving respective to laser scanner from top view. [16]

5.2.3 Particle Filter

An easier way to illustrate how AMCL works is to first look at the Monte Carlo Localization (MCL) in algorithm 5.1.

Only three inputs are needed: a known map m, motion estimation at current time u_t and measurement at current time z_t. Upon initialization, a set of N particles $\bar{\mathcal{X}}_t = x_{t-1}^{[1]} ... x_{t-1}^{[N]}$, which are taken as the prior at this stage, spread around the initial position over a Gaussian distribution. They are assigned with uniform importance. Each particle is updated according to the odometry estimation, that is, the motion model in more general terms. The next step is to assign new weight to each particle according to the measurement model. In our case, that is the belief of how likely a particle represents the true position. The alignment error between laser scans and the map is used for updating each particle's importance weight $w_t^{[n]}$. The smaller the error, the more important a particle becomes, which eventually affects the sampling step, where i new particles are drawn around their priori with probabilities proportional to the weight $w_t^{[n]}$ assigned. The new particle set $\bar{\mathcal{X}}_{t+1}$ obtained after sampling is defined as the posterior at the current time step, and the prior of the next time step.

Algorithm 5.1 MCL

$\bar{\mathcal{X}}_t = \mathcal{X}_t = \varnothing$
for $n = 1$ **to** N **do**
  $x_t^{[n]}$ = **motion_model** $(u_t, x_{t-1}^{[n]})$
  $w_t^{[n]}$ = **measurement_model** $(z_t, x_t^{[n]}, m)$
  $\bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \left\langle x_t^{[n]}, w_t^{[n]} \right\rangle$
end for
for $n = 1$ **to** N **do**
  draw i with probability $\propto w_t^{[n]}$
  $\bar{\mathcal{X}}_{t+1} = \bar{\mathcal{X}}_t + x_t^{[i]}$
end for
return $\bar{\mathcal{X}}_{t+1}$

MCL is able to deal with global localization problems because of its capability of representing various particle distributions. It however, falls short when dealing with robot kidnapping problems once particles converge closely around a single pose. Furthermore, a substantial amount of computational power is required due to the large numbers of particles to be maintained in the memory.

Luckily, the first problem can be solved by artificially injecting random particles to the particle sets. Instead of adding a fixed number of particles at each iteration, the amount and distribution of particles are determined by the average of importance weights

$$\frac{1}{M} \sum_{n=1}^{N} w_t^{[n]}.$$

Intuitively, the average of importance weights indicate how "believable" a measurement is. As a result, a lower measurement likelihood increases the amount of random particles drawn.

To deal with the second problem, we adjust the number of particles to a smaller set once they converge. An adaptive localization algorithm is therefore implemented in algorithm 5.2. It largely improves the computational efficiency. As presented, the first part of algorithm 5.2 is almost the same as that of algorithm 5.1, whereas the sampling part is adjusted as described.

Algorithm 5.2 AMCL

$\bar{\mathcal{X}}_t = \mathcal{X}_t = \varnothing$
for $n = 1$ **to** N **do**
  $x_t^{[n]}$ = **motion_model** $(u_t, x_{t-1}^{[n]})$
  $w_t^{[n]}$ = **measurement_model** $(z_t, x_t^{[n]}, m)$
  $\bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \left\langle x_t^{[n]}, w_t^{[n]} \right\rangle$
  $w_{avg} = w_{avg} + \frac{1}{M} w_t^{[n]}$
end for
for $n = 1$ **to** N **do**
  if w_{avg} is not good enough **then**
    $\bar{\mathcal{X}}_{t+1} = \bar{\mathcal{X}}_t + \text{random}(x_t)$
  else
    draw i with probability $\propto w_t^{[n]}$
    $\bar{\mathcal{X}}_{t+1} = \bar{\mathcal{X}}_t + x_t^{[i]}$
  end if
end for
return $\bar{\mathcal{X}}_{t+1}$

Chapter 6

Control and Estimation

In this chapter, we use a computer vision-based method to generate a reference track for the car to follow. Then, we introduce a kinematic and tire model to assist the control input calculation. Lastly, two controllers are implemented and compared: a PID controller and a lane keeping feedforward controller. Due to the simplicity of the former one, we focus on the later one in section 6.4.

6.1 Track Generation

6.1.1 Computer Vision

A grid based-map is chosen for presenting the environment, therefore, it is possible to access this map as an image such that every coordinate corresponds to a pixel. A grid can take three types of values: occupied, empty, unknown. Transforming a map to the image shown in figure 6.1 assigns black, white and grey to each corresponding pixel. To assure scalability, the dimension of the map is 2048x2048, where the resolution from pixel to pixel is 0.05 m. As mentioned in section 3.1, the track is set up in a limited space, holding the region of interest within a smaller area. Cropping and resizing the region of interest is proven to give a better performance.

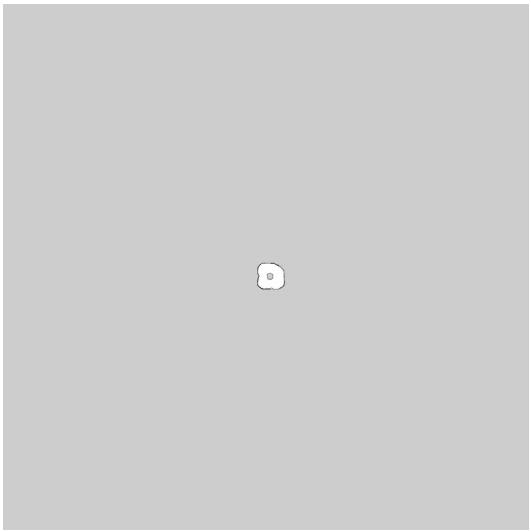


Figure 6.1: 2048x2048 map as an image, were white pixels are empty, black pixels are occupied and grey pixels are unknown.

Many explanation in this section are heavily dependent on visualization, thus, experimental results are shown to better explain the process. The following steps are made to reach a rough track representation:

Thresholding

Only white pixels are left in this step, allowing the image to be sharpened and the next step to only focus on pixels representing empty space.

Distance Transform

Distance transform normally applies to binary images only, which is why the first step is needed. It iterates through all pixels in the image and searches for the nearest "feature point". In this case, those are the black pixels that mark the boundaries of the track. The larger the distance, the higher value is assigned, and vice versa.

0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0

→

0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0
0	1	2	2	2	2	1	0
0	1	2	3	3	2	1	0
0	1	2	3	3	2	1	0
0	1	2	2	2	2	1	0
0	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0

Figure 6.2: Distance transform of an 8x8 image, with zeros being feature points and ones being the transformed pixels. Note that a "chessboard" distance metric [17] is applied in this example.

Skeletonization

First, the result of distance transform is normalized such that the middle-most pixels around the track take up the highest values, in other words, become white. Then, these pixels undergo a sequence of dilation and erosion [18] to generate a smoother and thinner track contour.

Figure 6.3: From left to right: Cropped and resized region of interest, binary image after thresholding step, distance transform and final rough track.

The track contour pixels are then being extracted and transformed from pixel coordinate to map coordinate. However, a more homogeneous density is desired as well as a smoother track. Therefore a spline fitting method is introduced in section 6.1.2 for further processing.

6.1.2 Spline Fitting

B-Splines are piecewise low-degree polynomial interpolant that contain certain control points and are specified by Bernstein basis function. Instead of fitting a high-degree polynomial over a large

set of points, as done in normal splines, B-splines guarantee a lower interpolation error. The fitted result is shown in figure 6.4.

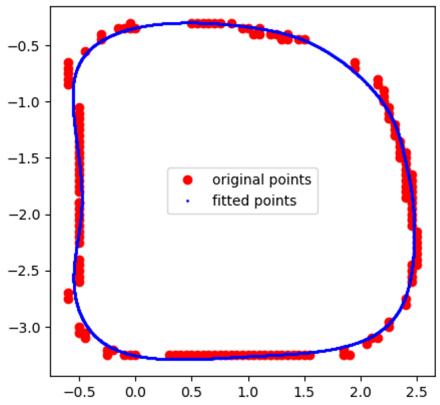


Figure 6.4: Spline fitting result. Axes are presented in meters.

Since less points are considered for fitting a piece of polynomial, points need to be ordered along the direction of the track. It doesn't matter whether it goes clockwise or counterclockwise, as long as the order is correct. A method that works for convex tracks as considered in this case is to transform all points into a polar coordinate and sort them according to angular values.

6.1.3 Curvature and Tangent Angle

For both calculations of curvature and tangent angle, a circle is fitted using three consecutive points in the cartesian space. Assuming the three points in the cartesian space are A, B and C with coordinates (x_1, y_1), (x_2, y_2) and (x_3, y_3) respectively.

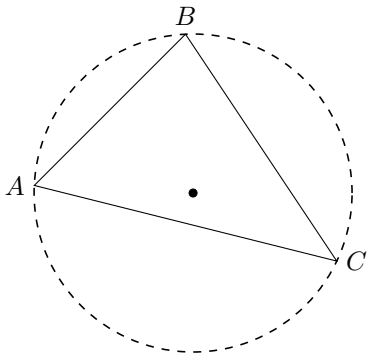


Figure 6.5: Three points A, B and C are presented in the cartesian space.

Curvature

Menger curvature κ deals with a triple of points in n-dimensional space. It can be calculated as the reciprocal of radius R, that is $\kappa = 1/R$, with

$$R = \frac{\sqrt{[(x_2 - x_1)^2 + (y_2 - y_1)^2] \cdot [(x_3 - x_2)^2 + (y_3 - y_2)^2 \cdot [(x_1 - x_3)^2 + (y_1 - y_3)^2]}}{2 \cdot |(x_2 - x_1) \cdot (y_1 - y_2) - (y_2 - y_1) \cdot (x_3 - x_2)|}. \tag{6.1}$$

Tangent Angle

The circumcenter coordinate (x_c, y_c) is needed for the tangent angle calculation. The intersection of two bisector lines to two arbitrary sides of the triangle gives us the circumcenter expressed in terms of the three points in figure 6.5:

$$\begin{aligned} x_c &= \frac{(y_3 - y_1)(y_2 - y_1)(y_3 - y_2) - (x_2^2 - x_1^2)(y_3 - y_2) + (x_3^2 - x_2^2)(y_2 - y_1)}{-4\mathbf{A}} \\ y_c &= \frac{x_2 - x_1}{y_2 - y_1} \cdot \frac{x_2^2 - x_1^2 + y_2^2 - y_1^2}{2(y_2 - y_1)} \cdot x_c, \end{aligned} \tag{6.2}$$

where $\mathbf{A}$ is the shoelace formula expressed as

$$\mathbf{A} = \frac{1}{2}\left(\begin{vmatrix} x_1 & x_2 \\ y_1 & y_2 \end{vmatrix} + \begin{vmatrix} x_2 & x_3 \\ y_2 & y_3 \end{vmatrix} + \begin{vmatrix} x_3 & x_1 \\ y_3 & y_1 \end{vmatrix}\right).$$

As presented in figure 6.6, the tangent angle θ can then be easily calculated by

$$\theta = \arctan\left(\frac{y_c - y}{x_c - x}\right) + \frac{\pi}{2}. \tag{6.3}$$

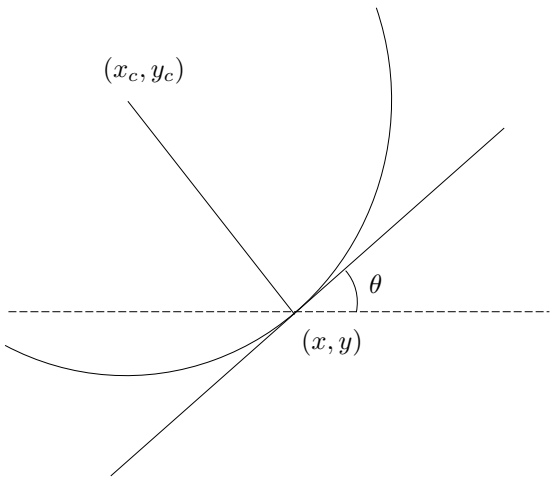


Figure 6.6: Tangent angle calculation.

6.2 Kinematic Model

This section shows the derivation of the kinematic bicycle model. Two parameters that affect our model are the distance from the car's center of gravity to its front and rear wheels respectively, as shown in figure 6.7. By assuming that the front steering wheels of a 4 wheel car share the same center of rotation through different steering angles, a front-steer car can be reduced to a bicycle model using Ackermann steering condition:

$$\cot(\delta) = \frac{1}{2}[\cot(\delta_1) + \cot(\delta_2)], \tag{6.4}$$

where δ_1 and δ_2 are the steering angles of the two front wheels.

The slip angles α_r and α_f, illustrated in figure 6.7, are expressed in terms of translational and rotational velocities v_x, v_y and ω as

$$\begin{aligned} \alpha_r &= \arctan\left(\frac{v_y - \omega l_r}{v_x}\right) \\ \alpha_f &= \arctan\left(\frac{v_y + \omega l_f}{v_x}\right) - \delta. \end{aligned} \tag{6.5}$$

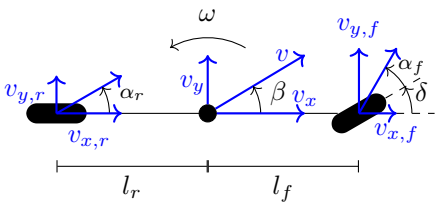


Figure 6.7: Illustration for the calculation of the slip angle. [11]

6.3 State estimation

The pose estimation of the localization algorithm presents orientation in terms of quaternions. However, only yaw angle is of relevance in our set up, so the orientation is transformed into roll, pitch and yaw. Besides the angle transformation, velocity estimate is also needed. A straightforward backwards difference is used and then processed with a 3rd order Butterworth low-pass filter for noise reduction.

6.4 Lane Keeping with Feedforward Controller

Two terms are considered as parts of the controller: a lookahead error e_{la}in the feedback controller as shown in figure figure 6.8, and a feedforward term δ_{ff}taking the track's curvature into account. When combining both terms, the steering input becomes $\delta_{steer} = \delta_{ff} + \delta_{fb}$ where δ_{ff} is the feedforward input and δ_{fb} the feedback.

Let κ^* be the desired curvature of the track, which is equivalent to a desired angular rate ω^*. Equations (6.5) can be modified to express steering angle in terms of curvature and car parameters:

$$\delta_{ff} = \alpha_r - \alpha_f + \kappa^*(l_r + l_f). \tag{6.6}$$

Optionally, a damping term can be added to the feedback term such that

$$\delta_{fb} = k_p \cdot e_{la} + k_d \cdot \dot{e}_{la}, \tag{6.7}$$

where

$$\dot{e}_{la} = \dot{\omega} - \dot{\theta}.$$

$\dot{\omega}$ is the car body's angular rate and $\dot{\theta}$ is the arc angle changing rate of the track.

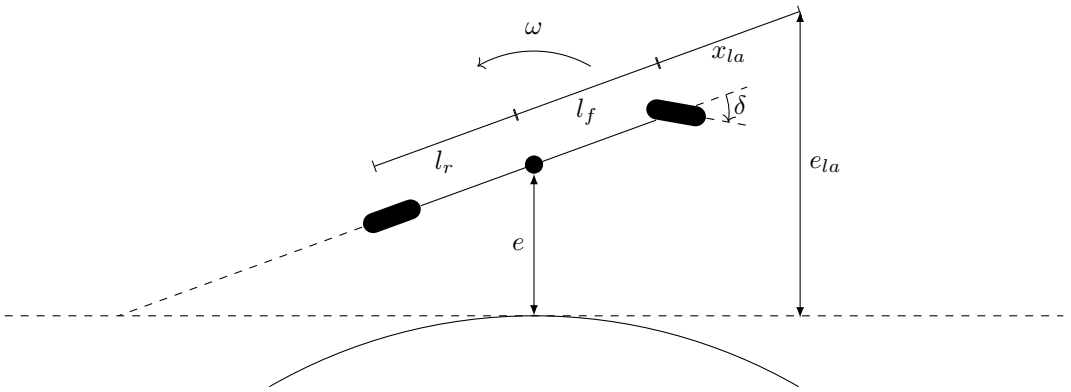


Figure 6.8: Illustration of the look-ahead error e_{la}. [11]

Chapter 7

Results and Discussion

In this chapter, the localization performance is evaluated. We also discuss the scalability of our track generation method. Then, a comparison is made between the two controllers implemented on our experimental setup. Every part is concluded with possible ways of improvement.

7.1 Path Generation

Most of the path generation results are already shown in section 6.1.1 and section 6.1.2 for the sake of explaining the concepts.

7.1.1 Track Scalability

The region of interest (ROI) cropping, which is the process that cuts the first step of figure 6.3 from figure 6.1, performs well under the assumption that we have knowledge of the map's scale. However, an adaptive method is desired if the size and starting point of the map is unknown. A simple solution can be implemented. The idea is to iterate through rows or columns of pixels from all four directions, and draw a rectangular bounding box according to the minimum ROI that includes all explored grids of the map.

7.1.2 Curvature and Tangent Angle calculation

One important factor for spline fitting is the smoothing parameter. There is a balance to strike between the track's smoothness and its ability to guarantee accurate navigation. From experimental results, a smoothing factor of 2.5 works well with the current setup. It is shown in figure 7.1 that spline fitting is indeed beneficial for both curvature and tangent angle calculation. The data is ready to use without requiring additional filtering or smoothing.

7.2 AMCL

The Vicon system is taken as the ground truth to evaluate the accuracy of AMCL. In the evaluation experiment, the car is being manually driven around the track twice, with the first lap being slow and the second being fast. Figure 7.2 shows that the aggressiveness has a significant impact on the localization accuracy. An offset in the faster lap occurs due to LiDAR distortions. Experiments also show that this error can be accumulated as more laps are driven.

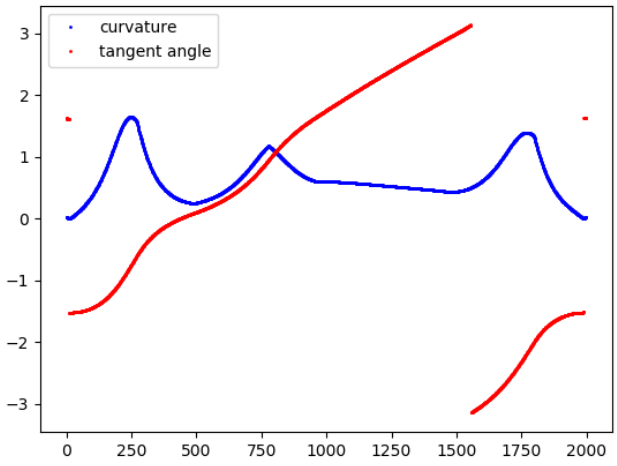


Figure 7.1: Curvature and tangent angle visualization over 2000 points

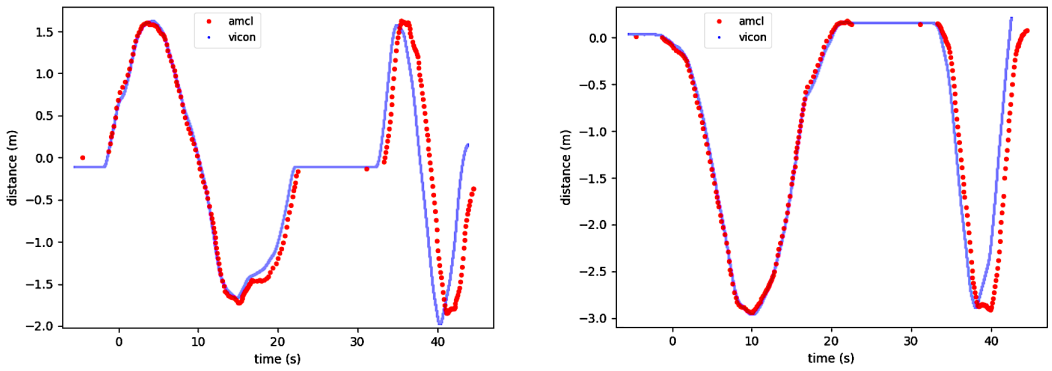


Figure 7.2: Left figure: x-axis evaluation; Right figure: y-axis evaluation

LiDAR scans the environment in circular motion, therefore, a drastic change in the yaw direction can distort the newer ranges taken into a scan. A LiDAR with faster scan rate can successfully improve this issue. However, as mentioned in section 3.3, upscaling LiDAR specifications can indicate a consequential increase of budget. Therefore, another preferred method is to add an IMU to a preprocessing step. Distortion can be compensated given a known motion estimation as shown in [19]. Furthermore, inaccurate odometry estimation also affects the localization performance, making sensor fusion another important task to be done to enhance RF2O output.

7.3 Controller

Figure 7.3 presents a comparison between a PID controller and the feedforward controller from section 6.4. As shown in the figure, the feedforward controller largely outperforms the PID controller in terms of lap smoothness. Additionally, it reaches a lap time of 16 seconds, whereas the PID controller takes 24 seconds to complete a lap.

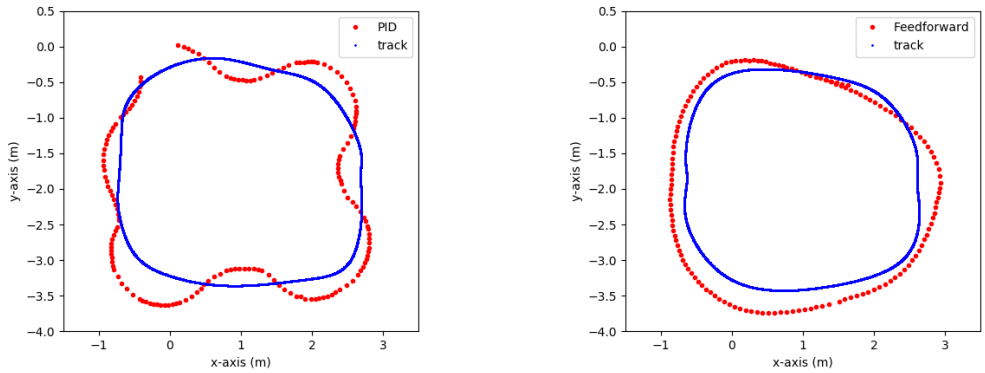


Figure 7.3: Path comparison of two controllers: PID and feedforward.

The lane keeping feedforward controller takes advantage of its knowledge about the curvature of the track, therefore, not over-steering when errors increase avoids oscillation that occur in the case of a PID controller implementation. As shown in the results, however, it reacts slower than desired, which is why the car mostly stays on the outer side of the track. One possible way to improve the feedforward controller's performance is using a more powerful computer, since the delayed reaction is believed to stem from the slow position update. Another possible way is to embed a motion model inside the controller itself, using known control inputs.

In the current setup, only a fixed torque is given. If experiments can be done on a bigger track with straight and curved segments, a longitudinal controller that slows the car down when approaching corners can be further explored. This would also be beneficial to observe oscillatory behaviors.

Better system identification can improve the car's performance. We assume linear mapping between the applied steering command and the effective steering angle. This is not necessarily true. In the CRS implementation [11] for example, the steering map is a piece-wise linear function.

7.4 Conclusion

A fully autonomous car has been created over the span of the project. With a planar LiDAR being its only sensor, it explores an unknown environment, saves the map, and finds a track around the map to follow. When following the track, two controllers are tested and compared to each other. The entire system runs on ROS, where the localization part can be seen as a potential extension of the CRS framework.

We gained deeper understanding of the system limitations, especially while implementing the two controllers. The performance of the chassis is yet to be explored, but this will require more computational power, which is easily saturated when performing localization. Adhering to our goal of keeping expenses low, computational resources of the NVIDIA Jetson Nano can be further exploit by parallelization or taking advantage of its GPU. Moreover, the 8-bit microcontroller should be replaced by a 32-bit one, in order to deploy longitudinal control. Besides better hardware, an IMU should be added, where an EKF can be used for providing a better state estimation.

For future work, we envision the implementation of an MPCC controller, which, however, requires more efficient computation and would only be necessary if an average velocity of 1.5 m/s or above is achieved. Whether we want to opt into achieving higher performance regardless of budget limitations can also be decided as future development goes.

Bibliography

[1] F. Boniardi, A. Valada, W. Burgard, and G. D. Tipaldi, "Autonomous indoor robot navigation using a sketch interface for drawing maps and routes," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2016.

[2] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, *Monte Carlo localization for mobile robots*, Computer Science Department, Carnegie Mellon University, May 1999.

[3] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," in *International Journal of Computer Vision*, October 1994, pp. 119–152. [Online]. Available: https://link.springer.com/article/10.1007%2FBF01427149#citeas

[4] A. Diosi and L. Kleeman, "Fast laser scan matching using polar coordinates," in *The International Journal of Robotics Research*, October 2007, pp. 1125–1153. [Online]. Available: https://www.researchgate.net/publication/220122650_Fast_Laser_Scan_Matching_using_Polar_Coordinates

[5] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2005, pp. 34–46. [Online]. Available: https://ieeexplore.ieee.org/document/4084563

[6] M. O'Kelly and V. S. Babu, *F1/10: An Open-Source Autonomous Cyber-Physical Platform*, Department of Electrical and Systems Engineering,Department of Computer Science,Dept. of Physics, Informatics Mathematics, University of Pennsylvania, University of Virginia, University of Modena Reggio Emilia, 2019.

[7] D. Ameida, *Implementing and Tuning an Autonomous Racing Car Testbed*, Research Centre in Real-Time Embedded Computing Systems, CISTER, 2019.

[8] N. Guedes, *A Robotic Platooning Testbed for Cooperative ITS Components*, Research Centre in Real-Time Embedded Computing Systems, CISTER, 2019.

[9] J. Kabzan, C. Ehmke, V. Reijgwart, and M. Prajapat, *AMZ Driverless: The Full Autonomous Racing System*, AMZ Driverless, Autonomous Systems Lab (ASL), Automatic Control Laboratory (IfA), ETH Zürich, Switzerland, May 2019.

[10] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem*, School of Computer Science, Computer Science Department, Carnegie Mellon University, Stanford University, Apr. 2007.

[11] C. Küttel, *Model Predictive Control for Autonomous Racing with Miniature Cars*, Institute for Dynamic Systems and Control (IDSC), ETH Zürich, Switzerland, Oct. 2020.

[12] S. Kohlbrecher, J. Meyer, O. von Stryk, and U. Klingauf, "A flexible and scalable slam system with full 3d motion estimation," in *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. IEEE, November 2011.

[13] B. D. Lucas and T. Kanade, *An Iterative Image Registration Technique with an Application to Stereo Vision (IJCAI)*, Computer Science Department, Carnegie Mellon University, Apr. 1981.

[14] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, Stanford University, University of Freiburg, University of Washington, 1999.

[15] J. Mariano, G. Monroy, and J. González-Jiménez, "Planar odometry from a radial laser scanner. a range flow-based approach," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 4479–4485. [Online]. Available: http://mapir.isa.uma.es/mapirwebsite/index.php/mapir-downloads/papers/217

[16] H. Spies, B. Jähne, and J. L.Barron, "Range flow estimation," in *Computer Vision and Image Understanding*, March 2002, pp. 209–231. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1077314202909707

[17] V. Ćurić, *Distance Functions and Their Use in Adaptive Mathematical Morphology*, Faculty of Science and Technology 1137, Uppsala University, 2014.

[18] "Opencv morphological transformations." [Online]. Available: https://docs.opencv.org/3.4/d9/d61/tutorial_py_morphological_ops.html

[19] P. Merriaux, Y. Dupuis, R. Boutteau, P. Vasseur, and X. Savatier, *LiDAR point clouds correction acquired from a moving car based on CAN-bus data*, 2017.

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institute for Dynamic Systems and Control
Prof. Dr. R. D'Andrea, Prof. Dr. E. Frazzoli, Prof. Dr. C. Onder, Prof. Dr. M. Zeilinger

Title of work:

Localization and Mapping for Close Loop Autonomous Racing

Thesis type and date:

Master's Thesis, January 2021

Supervision:

Dr. Andrea Carron
Prof. Dr. Melanie Zeilinger

Student:

Name: Patricia Sung
E-mail: pasung@student.ethz.ch
Legi-Nr.: 19-911-841
Semester: HS 2020

Statement regarding plagiarism:

By signing this statement, I affirm that I have read and signed the Declaration of Originality, independently produced this paper, and adhered to the general practice of source citation in this subject-area.

Declaration of Originality:

https://www.ethz.ch/content/dam/ethz/main/education/rechtliches-abschluesse/leistungskontrollen/declaration-originality.pdf

Zurich, 7.2.2021: