Institute for Dynamic Systems and Control

Benjamin Dupont

# Obstacle detection for autonomous racing go-karts using a dynamic occupancy grid

**Semester Project**

Institute for Dynamic Systems and Control
ETH Zurich

**Supervision**

Dr. Maurilio Di Cicco
Prof. Dr. Emilio Frazzoli

February 2024

IDSC-proj-sp-dupont

# Abstract

This report investigates the advancement of a dynamic occupancy grid for autonomous go-kart navigation, with a focus on real-time processing and dynamic obstacle avoidance. A new approach is introduced to complement or replace the current implementation on the go-kart, addressing robustness and precision issues. This approach includes deploying an occupancy grid in a local frame, coupled with a particle filter for dynamic obstacle tracking and a Kalman filter for state estimation. Despite challenges in motion compensation and dynamic obstacle prediction, evaluations demonstrate the system's relative effectiveness in real-time scenarios which can be considered as a proof of concept for the approach. The implementation, available on the IDSC group's GitHub repository, along with the rest of the code that runs on the go-kart represents a significant step toward robust autonomous navigation with multiple autonomous agents navigating the track simultaneously. Future work will aim at refining computational efficiency and enhancing detection capabilities, by for example focusing on a more quantitative based evaluation and fine tuning of the mutiple parameters available, before moving to the integration of a planner to interface with the controller.

Note: *This document is created with the document class IDSCreport [1].*

**Keywords:** Obstacle avoidance, self driving car, autonomous robot, occupancy grid, particle filter, gokart.

# Contents

# Chapter 1

# Introduction

## 1.1 Motivation

Autonomous driving is revolutionizing the way we think about vehicles and their capabilities, aiming to make them navigate their environments on their own. This goal comes with its fair share of challenges, especially considering the dynamic nature of most driving environments. These environments are filled with moving obstacles and other vehicles, some of which are autonomous, creating a complex scene that the vehicle must navigate through safely and efficiently.

The main challenge lies in enabling the autonomous vehicle, in this case, a go-kart, to understand its surroundings and navigate:

- Autonomously
- Without collisions
- Ensuring the safety of both passengers and external entities
- And, in the specific context of an autonomous go-kart, to achieve these objectives as fast as possible.

## 1.2 Context

This report focuses on the autonomous navigation of a racing go-kart, a scenario that accentuates the demand for agility and precision given the vehicle's inherent design for speed. The study leverages a specifically adapted electric go-kart equipped with a two-wheel rear drive system and an onboard computer for processing and control. Additionally, the go-kart is equipped with proprioceptive sensors such as wheel speed encoders, a steering angle sensor and an Inertial Measurement Unit (IMU), which together facilitate accurate vehicle positioning and orientation tracking. Complementing these are exteroceptive sensors, specifically a LiDAR and a camera, which play a critical role in obstacle detection. In the current phase of the project, the focus is placed on a LiDAR-only pipeline. The camera, though not initially prioritized for integration, is recognized for its potential to significantly enhance the system's environmental perception capabilities. Its incorporation into the system could offer valuable complementary data for obstacle detection and navigation, and is considered for future development phases once integration challenges are addressed.

To ensure operational safety during testing phases, the go-kart is equipped with a remote emergency system, enabling manual override and immediate system shutdown in the event of a malfunction. It is imperative to note, however, that this emergency system is not considered a substitute for the requisite safety features intrinsic to the problem definition of autonomous navigation. Instead, it serves as an additional layer of precaution, to ensure the safety in the development and deployment of the autonomous driving system.

## 1.3 Established framework

The development of the go-kart's current autonomous navigation capabilities represents a collaborative effort, building on the contributions of previous students, PhD candidates, and researchers. The following section briefly outlines the software solutions currently implemented on the go-kart and are depicted in the schematic of Figure 1.1. These serve as both a foundation and motivation for this project, aiming to build upon and enhance these existing capabilities As the current software implementations on the go-kart are explored, it is recognized that the complexity of the concepts may not be immediately accessible to all readers. For those desiring a more detailed exploration of these technicalities, extensive insights are provided through the project's GitHub repository and the documentation linked within the bibliography. The purpose of this overview is to frame the existing challenges and to showcase the opportunities for advancement that are being pursued in this project

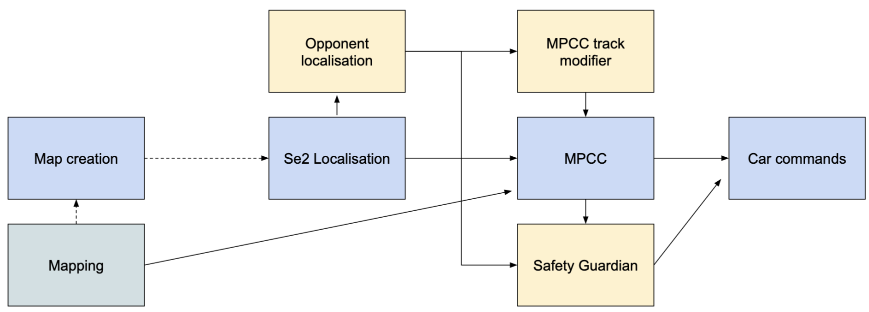



Figure 1.1: Schematic of the pipeline running on the go-kart

**General capabilities:**

- *Mapping:* Utilizes an occupancy grid based on Bayesian methods for creating a map of the environment and its boundaries. This is usually run once and provided to the other packages, however it can also be used in real time on unknown tracks, leveraging slower speeds and reliability.
- *Localization:* Uses the mapped track for precise go-kart localization, which is essential for effective navigation.
- *Controls:* use of MPC (Model Predictive Control) which aims to maximize progress along the track's centerline, minimizing control effort and deviation. It includes a soft constraint for exceeding track limits, balancing speed with safety.

**Obstacle avoidance specifics:**

- *Opponent Localization:* Processes LiDAR point clouds to identify obstacles in the global frame, applying DBSCAN for clustering and the Hungarian algorithm for matching obstacles in successive frames, ensuring accurate obstacle tracking.
- *MPC with Obstacle Avoidance:* Integrates obstacle data into the planning phase, adjusting trajectories to avoid detected obstacles effectively.
- *Safety Guardian:* Works alongside localization and obstacle detection to monitor MPC outputs. If a collision trajectory is detected, it intervenes with predefined safe commands to prevent accidents.

Note that the two latter points are two different strategies to make use of the given obstacles detected from a planning and controls point of view, to effectively be able to navigate around them.

## 1.4 Objective

Motivation for this project stems from identified areas for improvement in the existing systems. Localization, a critical component of autonomous navigation, is currently under enhancement by another student due to issues with the system frequently losing track and struggling to reacquire its position. Additionally, while the current obstacle detection system performs adequately with static obstacles, it faces challenges with erratic bounding box rotations and lacks reliable detection mechanisms for dynamic obstacles. These limitations highlight the need for advancements in sensor accuracy, obstacle detection and classification, and overall system reliability and robustness.

In response to these challenges, this project seeks to refine the accuracy of sensors and enhance the detection and classification of obstacles, particularly addressing the issue of rotating bounding boxes and the insufficient detection of dynamic obstacles. Furthermore, a significant motivation is to reduce the dependency on the existing localization package, which has demonstrated limitations in reliability. By introducing redundancy and improving the robustness of the system, the goal is to achieve a more dependable and efficient obstacle detection solution for the go-kart.

## 1.5 Structure of this Report

The structure of this report is designed to begin with a look into the theoretical and technical details of implementing a new obstacle detection strategy through a dynamic occupancy grid. Subsequently, an evaluation and discussion of the results obtained from this implementation are presented, followed by an exploration of potential future work and evolutionary possibilities for the project. The report concludes with a summary of the findings, leading to the references and bibliography for those interested in going further into the subject.

# Chapter 2

# Technnical implementation

## 2.1 First Approach During Learning Phase: Brute Force

At the very start of this project, a brute force approach is adopted, focusing on familiarization with the codebase, point cloud processing methodologies, and the overall framework. This phase lays the groundwork and motivation for more sophisticated techniques to be applied subsequently.

**Point Cloud Pre-processing (Ground Removal)**

In the preprocessing step, the ground is removed from the point cloud data using a self implemented *RANSAC* algorithm to fit a plane to the data and remove the so called outliers. This critical procedure simplifies the analysis by focusing on points of interest in the point cloud that are more likely to be obstacles the go-kart should avoid, and is essential for reducing the data's complexity and enhancing the efficiency of the clustering process that follows.

**DB-Scan Clustering of the Point Cloud**

Upon preprocessing, the DB-Scan (density based) clustering algorithm is applied to the point cloud. Chosen for its effectiveness in identifying clusters within spatial data based on density, DB-Scan segments the point cloud into distinct objects, potentially representing obstacles in the vehicle's pathway.

**Challenges Encountered**

Despite the approach's straightforward nature, several significant challenges are encountered:

- *Flickering:* The absence of temporal continuity in object detection results in flickering, where obstacles, i.e lidar point clusters, change between frames. This is due to different choices in the DB-Scan clustering algorithm, especially on long objects such as the barriers that can be clustered equally as one big or multiple smaller obstalces, ultimately undermining the reliability of obstacle detection.
- *DB-Scan Parameter Tuning:* The tuning of DB-Scan parameters is found to be unintuitive and lacking in robustness. Optimal settings require extensive trial and error, impractical for dynamic environments or varying conditions.
- *Absence of Memory for Past Measurements:* A lack of accounting for historical data leads to discontinuity in tracking objects over time, reducing the system's effectiveness in dynamic obstacle avoidance.
- *Questioning the Necessity of a 3D Processing Environment:* Processing the entire point cloud in three dimensions appears excessive, especially considering the computational resources required and potential for simplification. Assuming a 2D displacement of the go-kart in a

plane would lead to simplifications which would help the computational complexity greatly without necessarily impacting the robustness of the obstacle detection.

- *Dependency on Accurate Localization:* The effectiveness of this approach heavily relies on precise localization, which is not always reliable or available, further complicating the obstacle detection process.

These challenges highlight the limitations of this simple brute force approach within the context of robust obstacle detection, calling the need for more refined, adaptive, and efficient methods to process and interpret sensor data for real-time obstacle detection and avoidance. However it is important to note that the current obstacle detection approach that runs on the go-kart uses a similar pipeline to the above, whilst adding a few more complications to adress some of the limitations raised.

## 2.2 Dynamic Occupancy Grid

In response to the challenges encountered with the initial brute force approach, a Dynamic Occupancy Grid (DOG) in a local frame is introduced. This method is characterized by its independence from localization, allowing it to be run continuously, which emerges as a crucial aspect for enhancing reliability.

**Key Features of the Dynamic Occupancy Grid Approach:**

- *Continuous Operation Without Reliance on Localization:* The DOG is designed to operate independently of localization, facilitating continuous operation. This independence significantly contributes to the approach's reliability, addressing one of the major limitations identified in the brute force method.

- *Classification of Space into Occupied and Free Areas:* Through the implementation of the DOG, space within the local frame is classified into occupied and free areas. This classification is the groundwork for navigating the vehicle safely through its environment by identifying potential pathways and obstacles.

- *Inability to Distinguish Between Discrete Obstacles:* While the DOG excels in classifying space, it does not distinguish between discrete obstacles. This limitation suggests that while the approach can identify areas of occupancy, it cannot specify the nature or number of obstacles within these areas, in its basic form.

- *Assignment of States to Obstacles for Dynamic Detection:* An innovative aspect of the DOG approach is the assignment of states to obstacles, facilitating the detection of dynamic obstacles. This feature represents a significant advancement in the ability to navigate through environments with moving obstacles, enhancing the system's adaptability and responsiveness. A planner can then easily use this information to make informed choices on the best route to take.

These features of the Dynamic Occupancy Grid approach mark a significant evolution from the initial brute force method, addressing key challenges and laying a foundation for a more sophisticated, reliable, and efficient system for autonomous navigation.

### 2.2.1 General desctiption of an Occupancy Grid

An occupancy grid is conceptualized as a discretized 2D map, where each cell within the grid represents the probability of occupancy. This probabilistic framework allows for a nuanced representation of the environment, distinguishing between occupied, free, and unknown spaces based on sensor data.

**Key Components of an Occupancy Grid:**

- *Discretized 2D Map:* The environment is divided into a grid of cells, with each cell assigned a probability that reflects the likelihood of that cell being occupied by an obstacle or object, as seen in Figure 2.1.

- *Probability of Occupancy:* Probabilities are derived from sensor measurements, with values ranging from 0 (completely free) to 1 (completely occupied), enabling a probabilistic interpretation of the space.

- *Measurement from the Lidar Point Cloud Array:* Data for determining occupancy probabilities is obtained from the Lidar point cloud array. Measurements captured by the Lidar sensor are processed to update the occupancy status of cells within the grid.

- *Interpolation of Free Space:* Free space within the grid is interpolated from the Lidar measurements, allowing for the identification of clear paths. The interpolation process involves estimating the occupancy status of cells between the Lidar points, effectively aiming to reduce the map's sparsity without introducing incorrect probabilities.

The subsequent section will delve deeper into the methodology for interpolating free space, highlighting the technical strategies employed to refine the occupancy grid's representation of the environment.

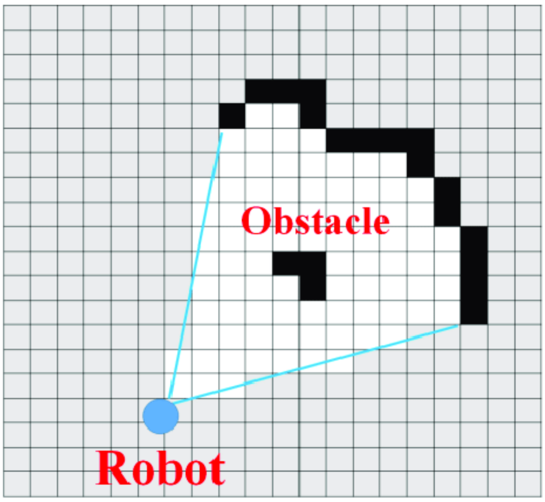



Figure 2.1: General depiction of an occupancy grid

### 2.2.2 Interpolation of free space

The interpolation of free space within the occupancy grid leverages the geometrical properties of the Lidar. It is inferred that when two points in the point cloud, originating from the same angular scan, are identified as free space, the space along the line in between these two points can also be safely considered free space. This inference relies on the fact that the path of the Lidar's laser beams to these points, being unobstructed, indicates the absence of obstacles in the intervening space. This is perfectly depicted in Figure 2.2.

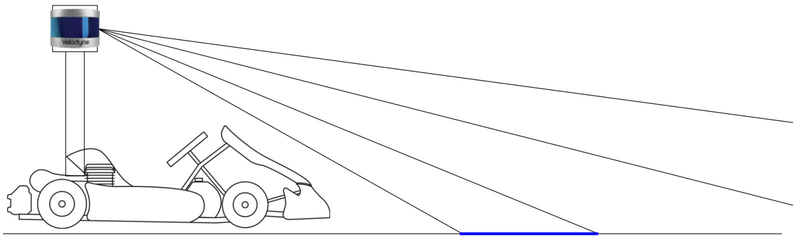


Figure 2.2: Correct interpolation of free space

Furthermore, the process incorporates a nuanced approach by considering a probability distribution for free space. Indeed, considering the special cases where an obstacle is small enough to not be detected by the lidar (see Figure 2.3), the interpolation approach should take into account this possibility and therefore not mark all points as equally free. Points directly measured by the Lidar to be free are attributed a higher probability of being free space, thus suggesting a lower probability of occupancy for these points, whereas points in the middle are slightly less probable of being fully free space. This method not only refines the accuracy of the free space mapping within the occupancy grid but also introduces a probabilistic element, adjusting the level of confidence in free space based on direct measurements versus interpolated inferences. Note that filling the occupancy grid with these values is done in this case using an adaptation of Bresenham's line algorithm, in order to easily find the grid indices for which the interpolating line passes through.

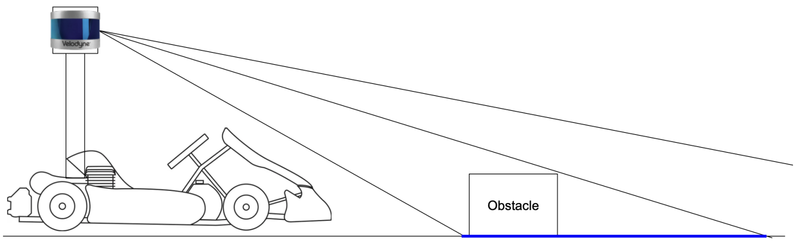



Figure 2.3: Limitation in the interpolation of free space

Finally, in the case where an obstacle is detected (Figure 2.4 below), nothing can really be inferred conveniently without making huge and case specific assumptions. Therefore, only the two discrete points are added to the grid, and the rest stays unknown ($p_{occupied} = p_{free} = 0.5$).

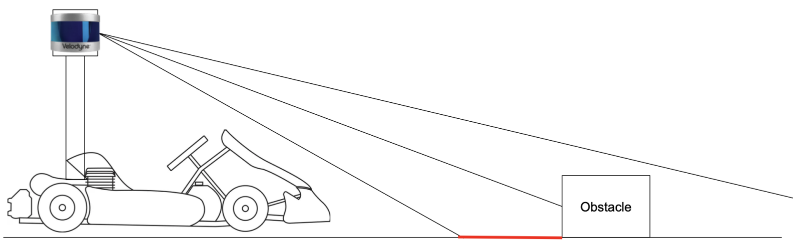



Figure 2.4: No interpolation possible

More importantly, this methodological approach significantly contributes to reducing the sparsity of the occupancy grid while maintaining a relatively precise discretization. By effectively filling in gaps within the grid, the environment's representation becomes denser, offering a more continuous and accurate depiction of navigable spaces. This enhancement is crucial for the system's autonomous navigation capabilities, as it provides a detailed and reliable map of the environment, enabling better decision-making and path planning. This can be seen in Figure 2.5, where the amount of grid cells marked as "free space" is greatly increased, even using a Lidar sensor with a limited number of beams.

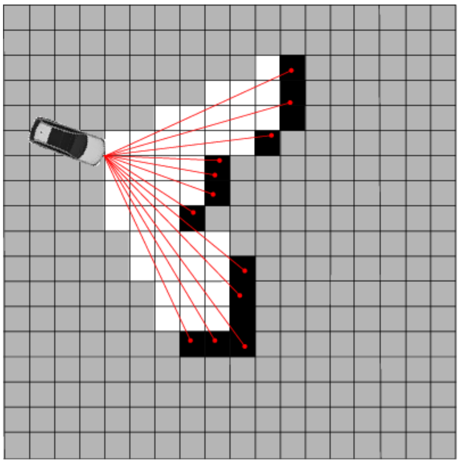


Figure 2.5: Occupancy grid with interpolation of free space

### 2.2.3 Choice of free space

A fundamental issue in processing 3D Lidar point clouds for autonomous navigation is the Lidar's inherent limitation: it does not explicitly return points designated as free space or obstacles. This ambiguity necessitates the development of methods to interpret the point cloud data effectively to distinguish between these critical environmental states.

**Utilizing RANSAC for Point Cloud Segmentation**

One approach to address this challenge involves employing the RANSAC (Random Sample Consensus) algorithm to segment the point cloud, which was used in the preliminary brute force approach. RANSAC operates by iteratively fitting a model to the data points and identifying outliers that do not fit the model. Through this process, surfaces, such as the ground plane, can be distinguished from obstacles that are left as the outliers in our plane hypothesis. However, while RANSAC is effective for model fitting and outlier removal, it is notably computationally expensive, posing challenges for real-time applications, when adding complexity to the full pipeline.

**Simplification using a Static Transform Based on Assumptions**

Given the computational demands of RANSAC, an alternative strategy capitalizes on certain assumptions about the environment and the vehicle's dynamics. Assuming low pitch and roll angles, alongside a predominantly flat racetrack, it becomes feasible to apply a single static transform to compensate for the Lidar's mounting angle and its height relative to the ground. Consequently, free space can be determined as points in the point cloud falling below a certain threshold, such as a z-coordinate less than 15cm (for example), indicating proximity to the ground plane and thus identifying navigable space. It is important to note that these assumptions are valid in the context of the go-kart's operation on a racetrack without loss of generality but may not be applicable in the case of a significantly different use case in other environments.

**Roll and pitch compensation**

To implement this approach, an initial process involves calibrating the static transform. This calibration is achieved by navigating the go-kart around the track, employing RANSAC to average the planes of measurements from multiple point clouds on one or more laps. This procedure allows for the extraction of a robust estimate of the ground plane, from which the static transform parameters can be inferred. By applying this transform to the Lidar data, a consistent method for identifying free space is established, significantly reducing computational overhead and enhancing the system's efficiency in real-time obstacle avoidance. To enhance the RANSAC algorithm's effectiveness in this context, it's tailored with a constraint that forces the plane's normal vector to be strictly vertical. This adjustment ensures the plane model fitted to the data doesn't end up fitting to a wall for example, which would be wrong and affect the averaging. A normalization process accounts for planes with opposite signs and ensures the averaging is not affected by planes that are spatially the same but represented by flipped signed paramters. Subsequently, by averaging the characteristics of these planes across multiple measurements, a reliable static transform for the Lidar is deduced. This transform compensates for the Lidar's mounting angle, facilitating a systematic identification of free spaces without the need of introducing computationally more expensive algorithms for segmentation such as RANSAC, as discussed above.

### 2.2.4 Motion Compensation

A big challenge in the deployment of a dynamic occupancy grid in a local frame for autonomous navigation arises from the inherent motion of the go-kart. Given that the DOG is computed in a local frame relative to the vehicle, the motion of the go-kart introduces a difference between the predicted states in the occupancy grid and the actual measurements obtained from the environment. This discrepancy necessitates the implementation of motion compensation mechanisms to ensure that the occupancy grid accurately reflects the current state of the surrounding environment.

To address this challenge, state estimation techniques are employed, specifically through the use of an Extended Kalman Filter (EKF). The EKF provides a framework for estimating the go-kart's velocity $(vx, vy)$, yaw rate $(\dot{\psi})$, and acceleration $(ax, ay)$. By modeling the uncertainty and dynamics of the go-kart's motion, the EKF enables the accurate prediction of the vehicle's state at any given moment, which is critical for effective motion compensation. This approach was inspired by the stated esimation used in the Driverless division of *AMZ Driverless*, as discussed in the paper [2, AMZ Driverless: The Full Autonomous Racing System].

#### Extended Kalman Filter Dynamics

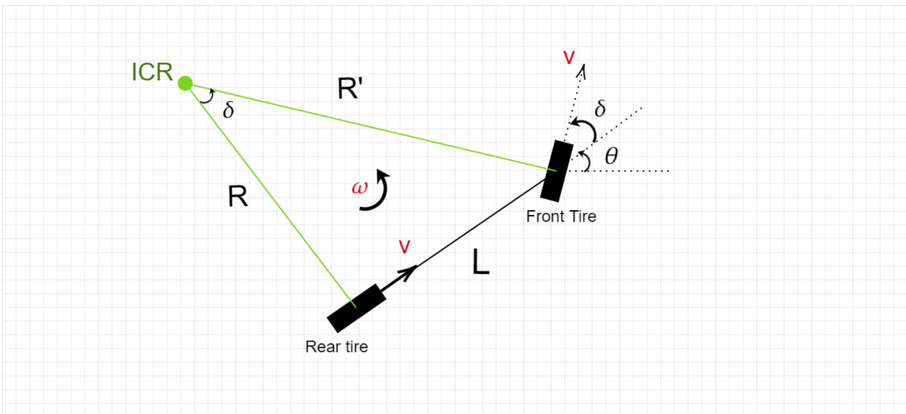



Figure 2.6: Kinematic bicycle model schematic

The motion compensation strategy further incorporates the use of a kinematic bicycle model as seen in Figure 2.6, which offers a simplified yet effective representation of the go-kart's dynamics. This model accounts for the vehicle's motion by considering it as a two-wheeled bicycle, where the front and rear wheels are replaced by single wheels at the center of their respective axles. The kinematic bicycle model, combined with the state estimates provided by the EKF, facilitates the computation of the necessary adjustments to the occupancy grid. These adjustments ensure that the grid accurately represents the environment from the perspective of the go-kart's current position and orientation, compensating for any changes due to the vehicle's motion.

By integrating state estimation via the Extended Kalman Filter with the kinematic bicycle model for motion compensation, the dynamic occupancy grid remains synchronized with the go-kart's movements. The state that is chosen to estimate is the following:

$$\mathbf{x} = \begin{bmatrix} v_x \\ v_y \\ \dot{\psi} \\ a_x \\ a_y \end{bmatrix} \tag{2.1}$$

This choice is made to give an idea of the general velocity and acceleration of the go-kart and can be used by any other package that would require such information.

The continuous time dynamics as given by a kinematic bicycle model are:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \ddot{\psi} \\ \dot{a}_x \\ \dot{a}_y \end{bmatrix} = \begin{bmatrix} a_x - \dot{\psi} \cdot v_y \\ a_y + \dot{\psi} \cdot v_x \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.2}$$

This leads to define the function $f$, reprensenting the state-transition model:

$$f(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} v_x + (a_x - \dot{\psi} \cdot v_y)\Delta t \\ v_y + (a_y + \dot{\psi} \cdot v_x)\Delta t \\ \dot{\psi} \\ a_x \\ a_y \end{bmatrix} \tag{2.3}$$

Finally, in the context of an EKF, the Jacobian matrix $F$ of the state transition model $f$ is needed and given by:

$$F = \begin{bmatrix} 1 & -\dot{\psi} \cdot \Delta t & -v_y \cdot \Delta t & \Delta t & 0 \\ \dot{\psi} \cdot \Delta t & 1 & v_x \cdot \Delta t & 0 & \Delta t \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

For more details on the assumptions and kinematics of the bicycle model used, one can refer to the Medium article written by *Yan Ding* [3].

### Extended Kalman Filter Sensor Model

The sensor model integrates various sensors available on the go-kart, each contributing for estimating the go-kart's motion parameters, and leverages redundancy. This motion estimation is critical for adjusting the dynamic occupancy grid to reflect the vehicle's movement accurately.

- *Wheel Encoders:* Wheel encoders are employed to provide an estimate of the go-kart's speed. By measuring the rotation speed of the wheels, these encoders offer direct insights into the vehicle's travel speed, serving as a foundational component of the motion estimation process.

- *Steering Sensor:* The steering sensor also helps further estimate the speed, combined with the above measurement from the wheel encoders. As these encoders are on the front wheels, the steering angle can give an indication for the relationship between $v_x$ and $v_y$.
- *Inertial Measurement Unit (IMU):* The IMU plays a critical role by providing estimates for both acceleration and angular rates. Note that in this case the IMU is very close to the center of gravity, and no compensation way applied. However in the general case, or to be even more precise in the estimates, this distance from the center of gravity of the go-kart should be take into account and compensated for.
- *Lidar and Iterative Closest Point (ICP):* The Lidar sensor, coupled with the Iterative Closest Point (ICP) algorithm, extend the sensor model's capabilities by offering estimates for the yaw rate and speeds. The ICP algorithm processes the Lidar-generated point cloud data to identify the vehicle's movement relative to its surroundings. This process involves matching consecutive point clouds to estimate the go-kart's change in position and orientation, providing an additional layer of data for motion estimation.

The integration of these sensors and processing techniques into the sensor model ensures a robust framework for motion estimation. By harnessing data from wheel encoders, steering sensors, the IMU, and Lidar through ICP, the system can accurately predict and compensate for the go-kart's movements using multiple sources which is great for redundancy and thus reliability. This comprehensive sensor model is indispensable for maintaining the integrity and relevance of the dynamic occupancy grid, thereby enhancing the go-kart's autonomous navigation capabilities by ensuring that environmental representations remain synchronized with the vehicle's real-time state.

All this is summarized in the following measurement vector used in the EKF (note that $v_{avg}$ is the average between the linear velocities obtained from the two front wheel encoders):

$$\mathbf{z} = \begin{bmatrix} v_{avg} \\ \delta_{steer} \\ a_{x,IMU} \\ a_{y,IMU} \\ \dot{\psi}_{IMU} \\ v_{x,ICP} \\ v_{y,ICP} \\ \dot{\psi}_{ICP} \end{bmatrix} \tag{2.5}$$

This leads to define the observation model from algebraic relations between the measured variables and the states:

$$h(\mathbf{x}) = \begin{bmatrix} \sqrt{v_x^2 + v_y^2} \\ \arctan\left(\frac{v_y}{v_x}\right) \\ a_x \\ a_y \\ \dot{\psi} \\ v_x \\ v_y \\ \dot{\psi} \end{bmatrix} \tag{2.6}$$

Again, in the context of an EKF, the Jacobian is required and defined as the following observation matrix $H$:

$$H = \begin{bmatrix} \frac{v_x}{\sqrt{v_x^2+v_y^2}} & \frac{v_y}{\sqrt{v_x^2+v_y^2}} & 0 & 0 & 0 \\ -\frac{v_y}{v_x^2+v_y^2} & \frac{v_x}{v_x^2+v_y^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{2.7}$$

**Extended Kalman Filter Prediction and Update equations**

Below, the equations of the EKF's implementation are refreshed, and for more details, the reader is encouraged to check out external literature such as the dedicated Wikipedia webpage [4].

- Predicted State Estimate: $\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_{k-1})$
- Predicted Covariance Estimate: $P_{k|k-1} = F_{k-1}P_{k-1|k-1}F_{k-1}^T + Q_{k-1}$
- Innovation or Measurement Residual: $y_k = z_k - h(\hat{x}_{k|k-1})$
- Innovation (or Residual) Covariance: $S_k = H_k P_{k|k-1} H_k^T + R_k$
- Near-optimal Kalman Gain: $K_k = P_{k|k-1} H_k^T S_k^{-1}$
- Updated State Estimate: $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k y_k$
- Updated Covariance Estimate: $P_{k|k} = (I - K_k H_k) P_{k|k-1}$
- State Transition and Observation Matrices Definitions: $F_k = \left.\frac{\partial f}{\partial x}\right|_{\hat{x}_{k-1|k-1}, u_{k-1}}$ $H_k = \left.\frac{\partial h}{\partial x}\right|_{\hat{x}_{k|k-1}}$

### 2.2.5 Particle filter for dynamical obstacles

**Integration of Measurements and Dynamic Obstacle Management**

With the sensor model in place, the system now receives measurements at every new timestep. Initially, the approach to update the dynamic occupancy grid involves compensating for the go-kart's movement, decaying the probabilities of occupancy using a factor less than 1, and incorporating new measurements from the Lidar into the grid. While this approach is simple, it is an easy way to use prior information in the occupancy grid, without relying uniquely on the measurement and allows the grid to describe a larger part of the environment whilst still acknowledging in a probabilistic way its certainty about occupancy using past observations.

**Challenges with Dynamic Obstacles**

A significant challenge arises when accounting for dynamic obstacles within the environment. The initial method, while effective for static environments, lacks the capacity to predict and adapt to obstacles that change position over time.

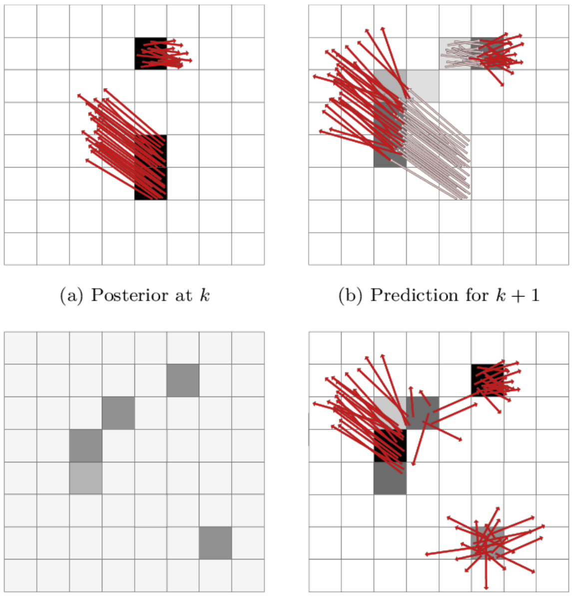
To address the limitations of handling dynamic obstacles, the occupancy grid is made dynamic. Rather than relying on a simple decay factor, a particle filter is introduced to enhance the grid's dynamism and predictive capabilities.

**Particle Filter Integration**

The particle filter operates by assigning particles throughout the occupancy grid, with each particle having a weight and a specific grid index association. These particles are not merely placeholders; each possesses a state defined by $(x, y, v_x, v_y)$ representing potential obstacle positions and movements.

This setup enables the system to go beyond static representations, allowing for the prediction of obstacle trajectories based on the linear velocities of the particles, as this is depicted in Figure 2.7. These predictions are crucial for anticipating the future positions of dynamic obstacles, thereby enabling proactive adjustments to the go-kart's path planning and navigation strategies.

Note that this approach and implementation is inspired by the paper from [5, Dominik Nuss and others], and the interested reader is encouraged to go through it for more detailed information.



(a) Posterior at $k$ (b) Prediction for $k+1$

(c) Measurement grid at $k+1$ (d) Posterior at $k+1$

Figure 2.7: Update process of the occupancy grid using the particle filter

### Advantages of Particle Filter Use

By employing a particle filter within the dynamic occupancy grid, the system gains a powerful tool for managing dynamic obstacles. This approach allows for a nuanced understanding of the environment, where obstacles are not only detected but their future movements are anticipated. The linear velocity of each particle, indicative of an obstacle's direction and speed of movement, becomes a key factor in the system's predictive capabilities, enhancing the safety and efficiency of the autonomous navigation system.

### Details of the implementation

As seen in Figure 2.8, the particle filter consists of multiple blocks that are called at every new measurement from the Lidar.

- *Initialisation:* In the initialization phase, particles are generated to represent hypotheses of the state space, each with an initial weight based on the prior distribution. This foundational step establishes the basis for subsequent predictions and updates within the particle filter framework and is only needed on the first call of the particle filter to setup everything such as allocate memory and initialise all the arrays.
- *Motion Compensation:* The motion compensation mechanism adjusts particles' states to account for the observed movement of the vehicle, ensuring that the predicted states remain consistent with the dynamics of the environment as the vehicle navigates through it. This is done using the estimate of the velocity of the go-kart as discussed above, and integrating the state.

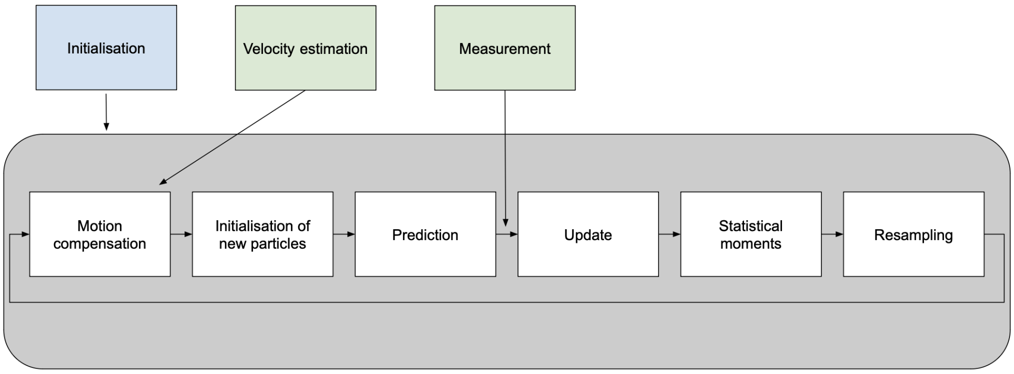



Figure 2.8: Particle filter pipeline schematic

- *Initialisation of New Particles:* New particles are introduced to represent emerging features or obstacles within the environment, enhancing the filter's capability to dynamically adapt to changes and maintain an accurate representation of the surrounding space.
- *Prediction:* During the prediction step, the future states of particles are estimated using a motion model, which incorporates the dynamics of the vehicle and the probabilistic nature of movement, to forecast the potential evolution of each particle's state.
- *Update:* The update process involves adjusting the weights of particles based on the likelihood of the new measurements given the predicted states, thereby refining the particles' representation of the environment in light of new sensor data.
- *Statistical Moments:* Statistical moments are calculated from the distribution of particles to provide estimates of the system's current state and its confidence, offering a quantitative understanding of the vehicle's position and the environment.
- *Resampling:* Resampling is conducted to address the issue of particle degeneracy, selecting a new generation of particles based on their weights to focus computational resources on the most probable hypotheses, thereby ensuring the particle set effectively represents the state space.

### 2.2.6 Final overview of the dynamic occupancy grid

With every block of the pipeline in place, the dynamic occupancy grid can be depicted by the schematic in Figure 2.9. The main inputs are the measurements from the lidar scan and the odometry sensors such as the steering angle sensor, the wheel encoders and the IMU. The lidar scan is processed into a measurement grid to segment the points in free space and occupied space with associated probabilities while the raw odometry measurements are used in the Kalman filter for velocity estimation. With this preprocessing, the new measurement is passed to the particle filter and the prior of the previous occupancy state is motion compensated thanks to the state estimation. The measurement grid is used to compare with the prediction from the new particles. This leads to a newly updated occupancy grid with particles associated to the obstacles to represent the speed of dynamic obstacles, which can then be used by a planner to effectively navigate the space with as much information as possible and avoid the obstacles.

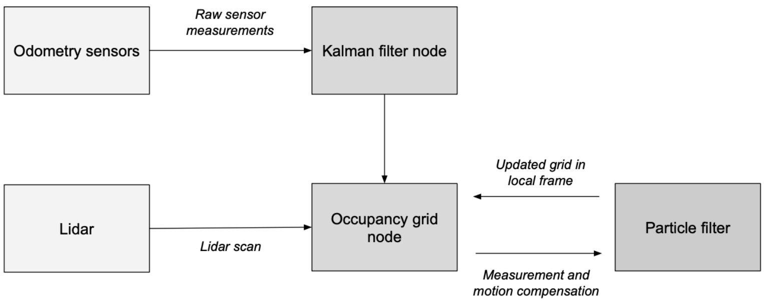



Figure 2.9: Overview of the dynamic occupancy grid pipeline

# Chapter 3

# Results and Discussion

In this chapter, the evaluation of the implemented approach focuses on qualitative analysis. Due to time constraints, an in-depth quantitative analysis and comprehensive tuning of parameters were not feasible within the project's timeline. However, the main goal of ensuring real-time operation, specifically a 20Hz update rate compatible with Lidar data acquisition, has been successfully met. The achievement of real-time processing underscores the practical viability of the approach. The exploration of parameter tuning, while not extensively covered in this phase, is identified as a crucial next step for enhancing the system's performance.

## 3.1 Evaluation of the Kalman filter for state estimation

In the context of motion compensation for the dynamic occupancy grid in a local frame, the importance of analyzing the Kalman filter's velocity estimation is crucial. As previously discussed in the theoretical segment, the Kalman filter's capacity for precise velocity estimation transposes directly to the accurate adjustment of the occupancy grid, by translating the grid of the environment into the local frame of the moving go-kart. Given the absence of ground truth data, the velocity estimations outputs from the Kalman filter are compared against the poses provided by the localization algorithm, which may not necessarily reflect the most accurate ground truth measurements. This method of evaluation is chosen to assess qualitatively the estimates of the Kalman filter and check for inconsistencies.

In the plot in Figure 3.1, the longitudinal velocity ($v_x$) and yaw rate ($\dot{\psi}$) estimates from the Kalman filter are very similar to those derived from the localization package. A side note is the voluntary difference in the measurements' covariance for the ICP of the Lidar point cloud, which is significantly increased in the plot's second half, to showcase the reduced noise as a consequence. This suggests a reduced necessity for this estimate, which seems to be of bad quality, and can be caused potentially by the measurement frequency (20Hz for ICP versus 200Hz). The $v_x$ estimation from the Kalman filter demonstrates a close correlation with the localization data, although a slight delay is visible of around 0,05 seconds, which should definitely be further investigated in a more thorough analysis. On the other side, the yaw rate estimation aligns almost perfectly, without noticeable lag. The $v_y$ estimation, however, shows more pronounced discrepancies, which should probably also need further investigation and the necessity of ground truth data for accurate assessment. This analysis underlines the importance of refining the velocity estimation processes to enhance real-time motion compensation within the dynamic occupancy grid which would in turn result in better obstacle detection capabilities, especially in highly dynamic environments.

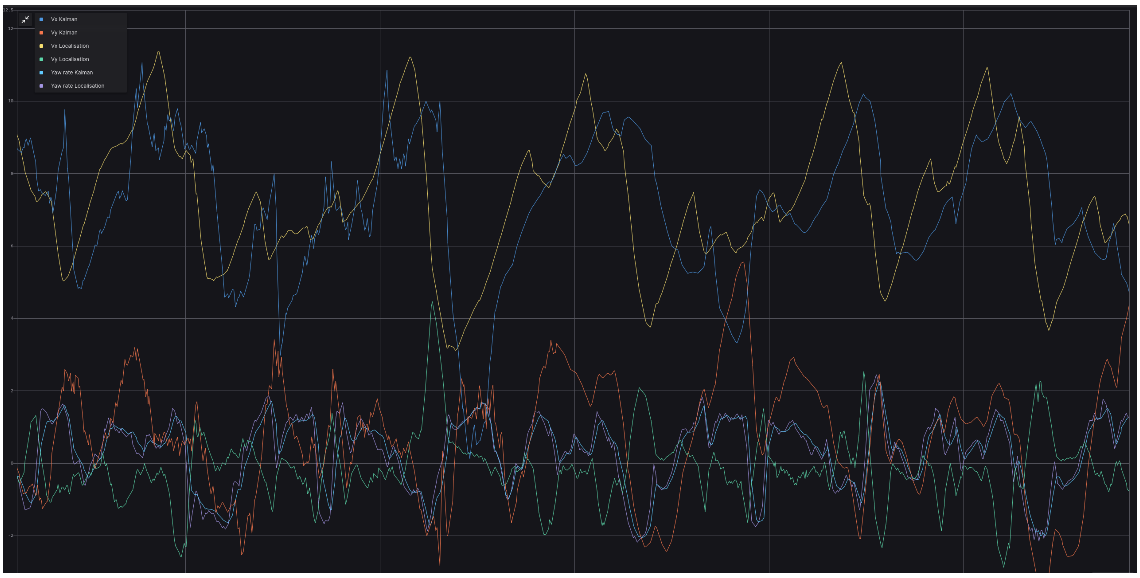



Figure 3.1: Velocity estimation telemetry comparison

## 3.2 Dynamic occupancy grid in the static case

In the static case analysis of the dynamic occupancy grid (static go-kart), as depicted in Figure 3.2, the grid is able to distinguish between free space (indicated by lighter areas on the map) and obstacles as expected. Particles in Figure 3.3 are observed to aggregate effectively around occupied spaces, with a little noise. The approach having numerous tunable parameters, the current setup may not represent the optimal configuration. Particularly, the velocity estimation for dynamic obstacles, as inferred from particle velocity values, appears somewhat imprecise, indicating potential issues with convergence. Addressing this might involve increasing the number of particles and reducing the noise in their state estimation, which is currently limited by the computation capacity on the CPU.

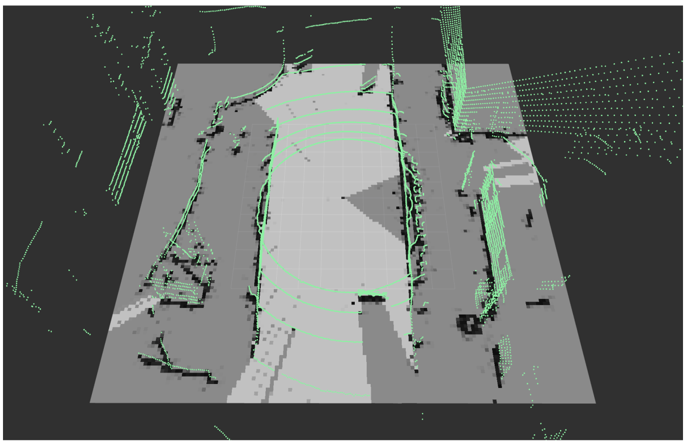


Figure 3.2: Dynamic occupancy grid with stationary go-kart

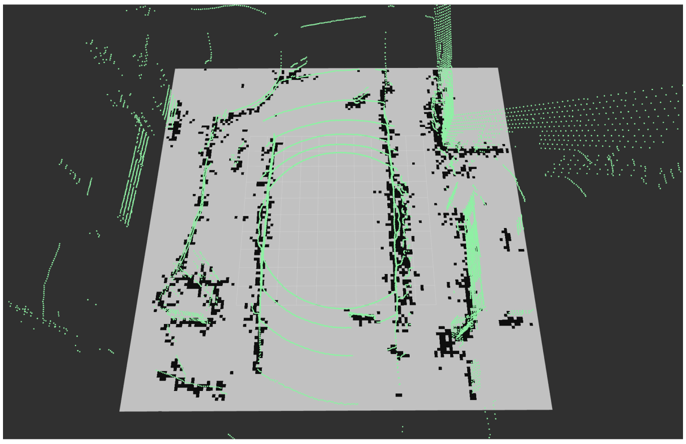


Figure 3.3: Particle grid with stationary go-kart

## 3.3 Dynamic occupancy grid in the dynamic case and its limitations

In the dynamic case where the go-kart navigates the track, the dynamic occupancy grid and particle grid demonstrate notable performance, as shown in Figures 3.4 and 3.5. The grids effectively map the environment based on real-time measurements, with particles converging satisfactorily under moderate maneuvers. This is possible thanks to the new born particles which are able to quickly adapt and track new obstacles that enter the space spanned by the grid.

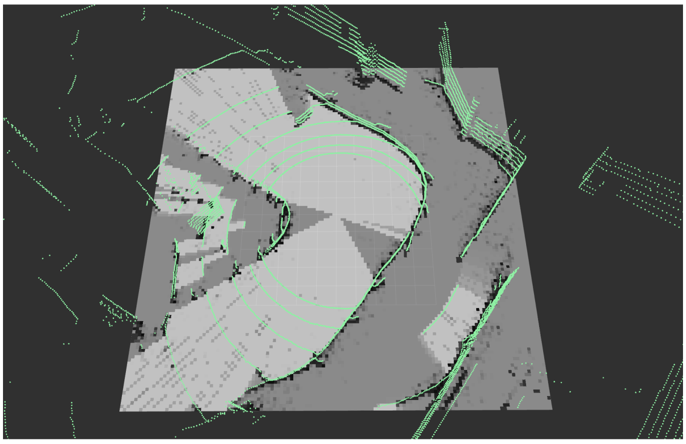


Figure 3.4: Dynamic occupancy grid with dynamic go-kart

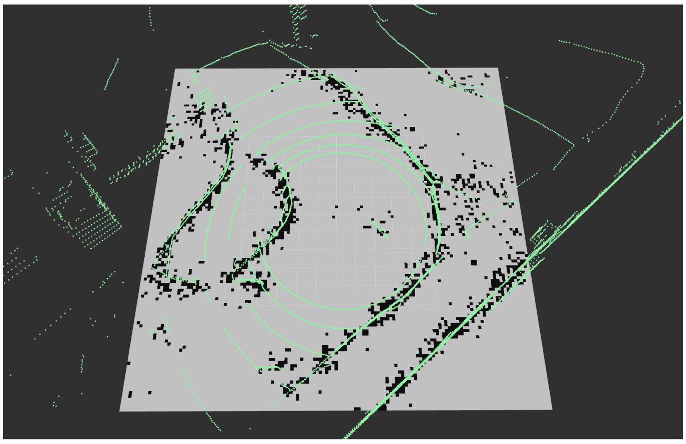


Figure 3.5: Particle grid with dynamic go-kart

However, challenges arise during abrupt movements, particularly in high yaw corners, where particle estimation of movement becomes less precise, as depicted in Figure 3.6. Additionally, the assumption of minimal pitch and roll proves inconsistent, occasionally leading to free space being inaccurately marked as obstacles, a limitation highlighted in Figure 3.7. This discrepancy is less pronounced near the go-kart, mitigated by angular geometric considerations. It is also important to note that the sparsity issue adressed with the interpolation of the Lidar points is quite effective as very few grey squares in the "free space" zone are present. However increasing the grid's resolution might heavily affect this. Additionaly, it can be seen that the interpolation often is not applied, especially to the right of the kart, since the first Lidar measurement immediately hits an obstacle. Despite the presence of this big zone of unknown space, it isn't a significant concern as any potential obstacles there are not really relevant on the planning side.

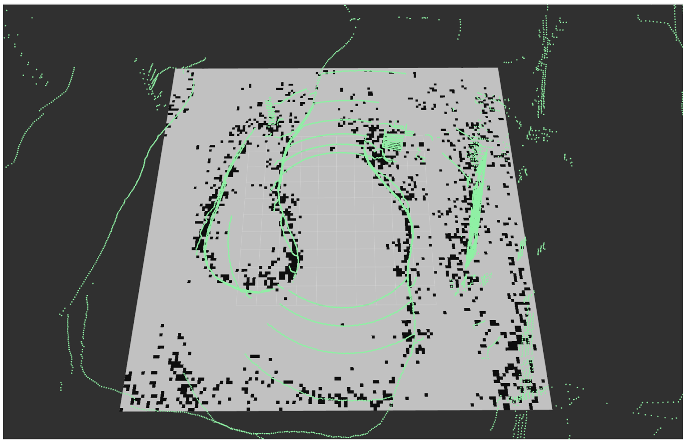


Figure 3.6: Particle grid limitation - Highly dynamic case

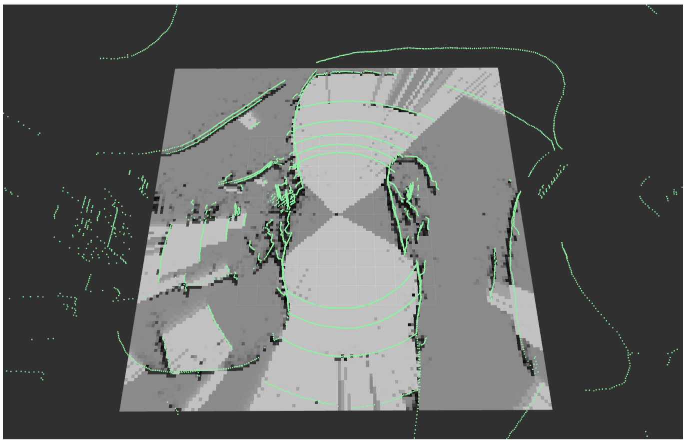


Figure 3.7: Occupancy grid - Limits of the pitch and yaw assumption

# Chapter 4

# Evolution and further work

As the project progresses, several possibilities for further development and enhancement have been identified. These focus on optimizing runtime and complexity, refining motion planning strategies, and implementing several improvements across the system.

**Runtime and Complexity**

- *GPU Implementation:* A significant improvement in runtime and computational efficiency can be achieved by adapting the code to run on a Graphics Processing Unit (GPU). GPUs offer the capability for parallelized computation, which is particularly beneficial for the particle filter used in the dynamic occupancy grid. By leveraging GPU acceleration, the system could handle an increased number of particles, by several magnitudes, and manage larger grid sizes without compromising performance. This is particluarly adapted to particle filters since every particle's initialisation or prediction is independant from the others so can be parallelized.

**Motion Planning**

- *Integration of a Local Frame Planner:* Incorporating a planner that operates within the local frame and accounts for known obstacles presents an opportunity to enhance motion planning. This planner would work alongside, or as an alternative to, the existing Model Predictive Control (MPC) trajectory, allowing for dynamic adjustments to the go-kart's path based on obstacle avoidance and optimal navigation strategies. It would also be possible to run alongside the mapping or localization package to allow redundancy and robustness, which was one of the issues identified as the motivation behid using this local frame approach.

- *Path Modification and Velocity Adjustment:* Further work could explore interfacing with the local frame planner to modify the planned path in real-time or adjust the velocity vector based on the relative velocity to obstacles (This can be thought of as a repelling force from the obstacles).

**Improvements**

- *Quantitative Analysis and Parameter Tuning:* The system could be significantly enhanced through more formal, quantitative analysis and the fine-tuning of its numerous parameters. A systematic approach to parameter optimization would help refine the balance between performance and computational efficiency and would hopefully demonstrate the dynamic obstacle tracking capabilities.

- *Complex Motion Models for Dynamic Obstacles:* Expanding the motion model to include additional states for particles would allow for more accurate prediction of dynamic obstacles. This enhancement could lead to more sophisticated anticipation of obstacle movements, improving the system's ability to navigate complex environments. This could also lead to the identification and classification of the obstacles based on their features in order to fit specific motion models.

- *Redundancy and Sensor Fusion:* Implementing redundancy through the integration of other obstacle detection methods, such as sensor fusion with camera data, could improve the reliability and robustness of the system. Combining Lidar data with visual segmentation from cameras would offer a more comprehensive view of the environment, enriching the measurement provided to the grid.

# Chapter 5

# Conclusion

This project presents a novel approach to autonomous navigation for go-karts, emphasizing a dynamic occupancy grid that operates independently of localization. By focusing on a more fundamental, low-level method of detection—where every object within the sensor's field of view is treated as a potential obstacle—this system offers a more robust framework for obstacle detection compared to traditional methods reliant on precise localization.

#### Benefits Over Other Approaches

The primary advantage of this methodology is its independence from localization, mitigating the risks associated with localization failures or inaccuracies. This attribute enhances the system's robustness, ensuring reliable detectioin and navigation even in environments where localization data might be compromised. Furthermore, by adopting a low-level approach to obstacle detection, the system ensures comprehensive coverage, treating every detected object as an obstacle and thereby minimizing the risk of oversight.

#### Identified Bottlenecks and Areas for Improvement

Despite these benefits, several bottlenecks have been identified that could limit the system's scalability and integration into fully autonomous navigation solutions:

- *Computational Limitations on CPU:* The current implementation's computational demands are significant, particularly for the particle filter used in the dynamic occupancy grid. To accommodate increased complexity and a larger number of particles, a transition to GPU-based computations is necessary.

- *Dependence on Accurate Motion Estimation:* The system's effectiveness is heavily reliant on precise motion estimation. Inaccuracies in estimating the go-kart's movement could lead to mismatches between the predicted and actual environmental states, affecting obstacle detection and navigation planning.

- *Lack of Integration with Autonomous Navigation Planners:* Currently, the system does not interface with any navigation planners, limiting its ability to be tested in a fully autonomous go-kart. This integration is crucial for assessing the system's performance in real-world autonomous navigation scenarios.

- *Reliability of Planning within the Occupancy Grid:* Planning paths within the occupancy grid, while innovative, introduces uncertainty due to its probabilistic nature. The accuracy of this approach, compared to deterministic methods, remains an area for further investigation and validation.

**Moving Forward**

Addressing these bottlenecks presents opportunities for significant advancements in the project. Transitioning computational processes to GPUs, enhancing motion estimation algorithms, developing interfaces with autonomous navigation planners, and rigorously testing the reliability of planning within a probabilistic occupancy grid are critical steps toward realizing a fully autonomous navigation system for go-karts.

In conclusion, this project lays the groundwork for a robust and localization-independent approach to obstacle detection in autonomous navigation for the go-kart. While challenges remain, the identified pathways for improvement and development hold the promise of overcoming these limitations, paving the way for more reliable and effective autonomous navigation solutions.

# Bibliography

[1] A. Ritter, P. Elbert, and C. Onder, *How to Use the IDSCreport LaTeX Class*, Version 1.8.0, Institute for Dynamic Systems and Control (IDSC), ETH Zürich, Switzerland, Aug. 2023.

[2] J. Kabzan, M. de la Iglesia Valls, V. Reijgwart *et al.*, "AMZ Driverless: The Full Autonomous Racing System," *arXiv:1905.05150 [cs.RO]*, May 2019. [Online]. Available: https://doi.org/10.48550/arXiv.1905.05150

[3] Y. Ding, "Simple understanding of kinematic bicycle model," https://dingyan89.medium.com/simple-understanding-of-kinematic-bicycle-model-81cac6420357, Feb 2020.

[4] Wikipedia contributors, "Extended kalman filter — Wikipedia, the free encyclopedia," 2023. [Online]. Available: https://en.wikipedia.org/wiki/Extended_Kalman_filter

[5] D. Nuss, S. Reuter, M. Thom, T. Yuan, G. Krehl, M. Maile, A. Gern, and K. Dietmayer, "A random finite set approach for dynamic occupancy grid maps with real-time application," *arXiv:1605.02406 [cs.RO]*, May 2016. [Online]. Available: https://doi.org/10.48550/arXiv.1605.02406

Institute for Dynamic Systems and Control

Prof. Dr. R. D'Andrea, Prof. Dr. E. Frazzoli, Prof. Dr. Lino Guzzella, Prof. Dr. C. Onder, Prof. Dr. M. Zeilinger

**Title of work:**

Obstacle detection for autonomous racing go-karts using a dynamic occupancy grid

**Thesis type and date:**

Semester Project, February 2024

**Supervision:**

Dr. Maurilio Di Cicco
Prof. Dr. Emilio Frazzoli

**Student:**

Name: Benjamin Dupont
E-mail: bdupont@student.ethz.ch
Legi-Nr.: 19-815-976
Semester: HS 2023

**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, 15.2.2024: ______________________