Dynamic Occupancy Grid Mapping with Velocity Measurements by a Particle Filter Approach

Teeuwen, D.H.E.

Award date:
2021

Master's Thesis

Dynamic Occupancy Grid Mapping with Velocity Measurements by a Particle Filter Approach

DC 2021.082

Author:

D.H.E. Teeuwen
0851711

Supervisors:

prof. dr. H. Nijmeijer
dr. ir. T.P.J. van der Sande
dr. ir. R.F. Tigrek

TU/e - Department of Mechanical Engineering
Dynamics and Control Group

Eindhoven, September 8, 2021

Declaration concerning the TU/e Code of Scientific Conduct for the Master's thesis

I have read the TU/e Code of Scientific Conduct[i].

I hereby declare that my Master's thesis has been carried out in accordance with the rules of the TU/e Code of Scientific Conduct

Date 8/9/21

Name D.H.E. Teeuwen

ID-number 0851711

Signature 

Insert this document in your Master Thesis report (2nd page) and submit it on Sharepoint

[i] See: http://www.tue.nl/en/university/about-the-university/integrity/scientific-integrity/
The Netherlands Code of Conduct for Academic Practice of the VSNU can be found here also.
More information about scientific integrity is published on the websites of TU/e and VSNU

Abstract

This work aims to research and develop a dynamic environment mapping approach. The occupancy grid map is used as the framework in this approach. It creates a map by tessellation of the environment in grid cells, in which occupancy is represented as a probability per cell. Measurements of the environment are converted to occupancy probabilities by a shaped function, derived from an inverse sensor model, and updated in an log-likelihood update equation. Additionally, a particle filter (PF) is used to approximate the occupancy as a non-linear distribution over the area surrounding the vehicle. The PF functions according to an update and correction principle. Each particle is assigned a position and velocity state, which are corrected in weighting and resampling steps by range and velocity measurements. An occupancy estimate is derived in probabilities, from the amount of particles per grid cell. Both outputs, from the direct mapping and PF components, are combined to create the dynamic occupancy grid map.

The proposed dynamic environment mapping approach is developed in simulation by implementation in a 1D setup. It is evaluated in several stages, by defining simulation scenarios and generating measurements by sensor models. Some scenarios are created to analyse functionality, quality and performance of the algorithm, each simulating a specific challenge to environment mapping. Evaluation metrics are introduced to objectively quantify the performance, by comparing the generated maps against ground truths. First, the functional analysis is used to determine if the algorithm works as intended. It concludes that it does so, but has two limitations in representation of accelerating objects and objects on which detections are lost. Secondly, a quality analysis is performed to show the benefits of the different components and additions to the proposed mapping algorithm. Lastly, a parameter study is performed to indicate the sensitivity of parameters to the performance of the algorithm. The latter analysis results in recommendations to improve performance of the algorithm.

To conclude, the proposed concept has been shown to estimate occupancy well in the simulated scenarios. Due to the inclusion of velocity measurements, the prediction by the PF becomes more accurate. By combing the direct mapping and filter outputs, the resulting dynamic occupancy grid map is able to represent occupancy in unobservable areas when detections of objects are missed or lost. The research concludes with recommendations on how to improve the current implementation and the steps required for implementation of the concept.

Acknowledgements

I would like to take this opportunity to thank several people who have helped me complete my thesis in these unprecedented times, during the Covid-19 pandemic. Due to the restrictions imposed, most of the work had to be done from home. In spite of that, I received plenty of support and understanding during this time.

First of all, I am extremely grateful for the support provided by my supervisors dr. ir. Tom van der Sande and dr. ir. Firat Tigrek, who gave me the opportunity to work on this most interesting topic and have guided me by our weekly discussions. Their comments and feedback provided me with valuable insights, helping me along in my research, as I could learn from their expertise and experience.

I would like to extend my sincere thanks to prof. dr. Henk Nijmeijer, my main supervisor. He was able to provide me with suitable, critical assessments on the progress of my research and give valuable insights during our regular meetings. I am grateful for his timely and honest feedback during this study.

Thanks should also go to my family, girlfriend and friends, who have provided me with support and understanding during this research. Both with fruitful discussions, but also by happy distractions which greatly helped keep me motivated. Without them I would not be were I am now and I would like to sincerely thank all for that.

Contents

List of abbreviations

1D, **2D**,..	**One**-, **Two**-,.. **D**imensional
ADAS	**A**dvanced **D**river-**A**ssistance **S**ystem
ADS	**A**utomated **D**riving **S**ystem
BOF	**B**ayesian **O**ccupancy **F**ilter
C-DMAT	**C**ooperative **D**ual **M**ode **A**utomated **T**ransport
CVCO	**C**onstant **V**elocity **C**onstant **O**rientation
EKF	**E**xtended **K**alman **F**ilter
FNR	**F**alse **N**egative **R**ate
FOV	**F**ield **O**f **V**iew
FPR	**F**alse **P**ositive **R**ate
GM-PHD	**G**aussian **M**ixture **P**robability **H**ypothesis **D**ensity
GOSPA	**G**eneralized **O**ptimal **S**ub**P**attern **A**ssignment
GPS	**G**lobal **P**ositioning **S**ensors
GPU	**G**raphical **P**rocessing **U**nit
GT	**G**round **T**ruth
i-CAVE	**I**ntegrated **C**ooperative of **A**utomated **Ve**hicles
IMM	**I**nteracting **M**ultiple **M**odel
IMU	**I**nertial **M**easurement **U**nit
LiDAR	**Li**ght **D**etection **a**nd **R**anging
ME	**M**ap **E**rror
MS	**M**ap **S**core
PDF	**P**robability **D**ensity **F**unction
PF	**P**article **F**ilter
PFS	**P**arametric **F**ree **S**pace
PHD	**P**robability **H**ypothesis **D**ensity
PHD/MIB	**P**robability **H**ypothesis **D**ensity **M**ulti-**I**nstance **B**ernoulli
RADAR	**RA**dio **D**etecting **A**nd **R**anging
RBPF	**R**ao-**B**lackwellized **P**article **F**ilter
RCS	**R**adar **C**ross **S**ection
ROC	**R**eciever **O**perating **C**haracteristic
SMC-BOF	**S**equential **M**onte **C**arlo **B**ayesian **O**ccupancy **F**ilter
SNR	**Signal**-to-**N**oise **R**atio
TNR	**T**rue **N**egative **R**ate
TPR	**T**rue **P**ositive **R**ate
TU/e	**E**indhoven **U**niversity of **T**echnology
UKF	**U**nscented **K**alman **F**ilter

Nomenclature

The notations that are kept throughout this document are listed below.

List of symbols

m	grid map
C	notation of a grid cell
L	grid cell length, in 1D
r	range
θ	angle
v	velocity
z	measurement vector
σ^2	measurement variance
λ	log odds ratio
ζ	decay factor in update equation of direct mapping
N	set of particles
x	state vector of particles
ν	random process noise
w_r	weighting criteria based on position
w_v	weighting criteria based on velocity
w_ρ	weighting criteria based on amount of particles per grid cell
w_{total}	particle weight
α	decay factor in total weight equation of the PF
γ	measurement residual
Λ	likelihood for associating measurements
p	probability density function
$\mathcal{N}(x; \mu, P)$	normal distribution over x with mean μ and covariance matrix P

Subscripts

t	time-step
k	discrete time-step
z	indication that a variable is received by measurements
d	detection
fa	false alarm
ref	reference value
o	indicator for 'occupied', as the occupancy state
f	indicator for 'free' as the occupancy state
GT	indicator for a value from the Ground Truth map

Superscripts

c	grid cell index
i	particle index
A	indicates association to a measurement

1 Introduction

A recent survey by the U.S. Department of Transportation [1] has found that the majority (94 percent) of the crashes involving light vehicles are caused by human driving error. Of those cases the majority of critical reasons, i.e. a main contributor in the casual chain leading to the accident, are classified as recognition (41 percent) or decision (33 percent) errors. Similarly, performance (11 percent) and non-performance (7 percent) errors, such as overcompensation, poor vehicle control or sleep, are large contributors. This data shows significant potential to increase traffic safety by avoiding human driving errors through automating driving tasks. As such Advanced Driver-Assistance Systems (ADAS), Automated Driving Systems (ADS) and related technologies are an active field of research. Additionally, ADS offer the potential to improve driver comfort, traffic throughput and fuel efficiency. However, the full potential of this technology cannot be realised unless the robustness of the state-of-the-art environment perception techniques in this field are improved further [2].

This assignment aims to contribute to development of automated driving by contributing to the Integrated Cooperative Automated Vehicles (i-CAVE) [3] project of the Eindhoven University of Technology (TU/e). The i-CAVE project addresses current transportation challenges regarding throughput and safety within an integrated approach to automated and cooperative driving. Within this project a Cooperative Dual Mode Automated Transport (C-DMAT) system is researched, enabling both automated and manual driving and thus allowing for maximum flexibility. Besides these enabling technologies, focus is put on fault tolerance and fail safety, wireless communications, human factors and others addressing transition of control between manual and automated driving and the response of other road users. i-CAVE aims to tackle the main challenges of automated driving, i.e. achieving high levels of safety and reliability through rigorous technological design, combined with integration between automated and manual driving to obtain maximum flexibility and user acceptance. A demonstration platform will be used for the integration and evaluation of mapping and localisation techniques, cooperative vehicle control algorithms and fleet management methods. In addition it allows for a close-to-market transport system, which can be commercialised by the transport industry, by applying results in their road maps.

The i-CAVE project consists of seven work packages, each focused on a specific part of the project. Central in the project is the living lab, or test vehicle, which is used by all other six to implement and evaluate their work. This research will be involved in the cooperative vehicle control and the communication groups, as well as the demonstrator group for potential implementation. Within the former group, controllers for cooperative and automated vehicles are developed, based on in-vehicle and inter-vehicle interaction. It includes developing vehicle control algorithms on vehicle and platoon levels, path planning algorithms and vehicle state estimation of both vehicle states and its surroundings. The communication group addresses implementing Radio Detecting And Ranging (RADAR)-based communication, allowing ADAS to be used as both radar sensors and communication devices, realising a more robust and synergistic approach to sensing and communication for safe high-speed automated and cooperative driving.

This study will focus on developing an environment mapping approach for the i-CAVE project, with the available sensor set. In this chapter an introduction to environment perception and mapping approaches is given in Section 1.1 by providing a general background of relevant theory. Then Section 1.2 identifies the current challenges in this field and details the problem description. In Section 1.3 the identified challenges are addressed by formulating a research question and intermediate objectives. Here the main contributions of this research are highlighted. Finally, the outline of this report is detailed in Section 1.4.

1.1 Environment perception and mapping

Autonomous vehicles require a good understanding of their surroundings to execute safe and efficient driving tasks. Therefore they are equipped with a vast array of hardware and sensors, as redundancy is important for robustness and reliability. Data of these sensors is interpreted to extract meaningful environmental features and map the surrounding environment, such that the ADS can use this information to plan and execute their tasks. This process is referred to as environment perception and mapping. This section will give a general introduction to the theory behind the process, by introducing the available sensor set and relevant hardware in Section 1.1.1 and a general introduction to environment mapping methods in Sections 1.1.2 through 1.1.4.

1.1.1 Sensors and hardware

As introduced the i-CAVE project features a living lab, which consists of test vehicles. The selected vehicle is a Renault Twizy, equipped with a selection of hardware as illustrated in Fig. 1.1. The vehicle is equipped with a set of exteroceptive sensors, which is a common term in robotics [4] and refers to sensors used to acquire information from the vehicle's surrounding environment. Examples of this kind of sensor include cameras, laser range finders, radar and ultrasonic sensors. This section will detail the characteristics of each sensor type on the vehicle and important considerations for fusing data, i.e. combining data of different sensors, based on an overview of the state art by Zegelaar et al. [5]. The latter is recommended as data of these sensors can compliment each other by improving accuracy and increasing reliability.

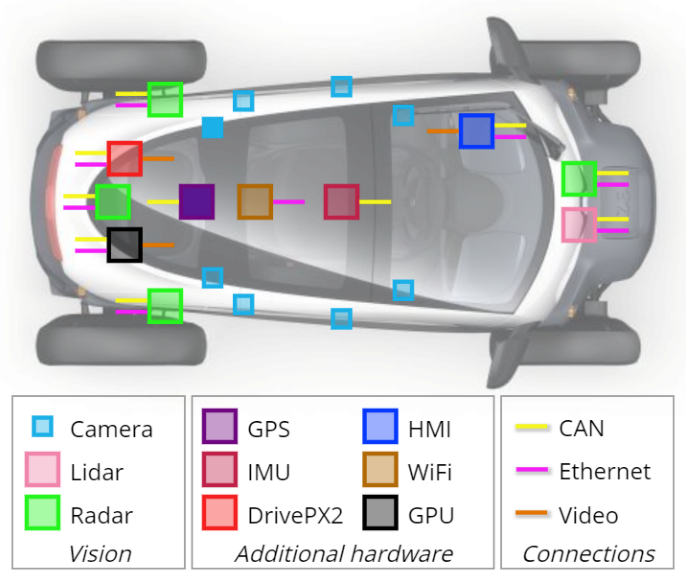


Figure 1.1: Overview of the sensor and hardware layout of a Renault Twizy test vehicle.

Next to the exteroceptive sensor set, the test vehicle features sensors to measure values internal to the system. These are referred to as the proprioceptive sensor set [4], and include components such as an Inertial Measurement Unit (IMU) and Global Positioning Sensors (GPS). Using these measurements the location of the ego vehicle can be determined between subsequent environment scans, for example. Additionally, the vehicle features crucial hardware, such as a Real-Time Graphical Processing Unit (GPU) and a Nvidia DrivePX2 to process the received measurements.

Radar sensing used in automotive applications provides good radial distance and velocity measurements in a 2D plane, but it has a mediocre angular resolution and typically no vertical information. It is hardly affected by adverse weather or poor lighting conditions, with ranges up to 300 meters. It returns reflection points with radial distance and velocity relative to the equipped vehicle, based off the return time of the electromagnetic waves and by the Doppler effect [5]. More advanced radar systems with capability to measure in a 3D space are available nowadays, but these are kept out of the scope of this research.

The commercial-off-the-shelf technology in camera sensing offers ranges up to 100 meters. A trade-off between cost, resolution range and image size is to be expected. In general the technology offers high angular resolutions, but mediocre radial distance and speed measurements. The downside of this sensor type is that measurements are greatly affected by adverse weather and poor lighting conditions. The test vehicles are equipped with monocular cameras and 2D image processing is applied. This leaves room for improvement as more accurate, 3D information can be generated by combining data of multiple viewpoints. However, 2D computer vision is already an established field with remarkable state of the art algorithms which are able to generate accurate information from monocular images [2]. In this study it is assumed that the available camera data is already processed and that detections are received by their respective coordinates in a bird's eye view.

Additionally, a Light Detection and Ranging (LiDAR) sensor is considered as an extension to the exteroceptive sensor set. Comparatively to the other sensors, lidar is a 3D technology that offers good radial, angular and vertical distance and is useful in constructing 3D environment maps. The sensor is affected by adverse weather, but not by bad lighting conditions [5]. It outputs point cloud data, which accommodates easy fusion with radar as it outputs a similar data. Compared to radar technology it is generally more expensive and requires higher computation effort to process measurements. However, unlike radar it is generally unable to provide velocity measurements of detections.

Radar and lidar sensors are very useful in covering the shortcomings of cameras. Depth information, i.e. distance to objects, can be measured effectively to retrieve 3D information with these sensors, and they are not affected by illumination conditions [2]. Additionally, radar provides Doppler velocity measurements and its resolution and accuracy of the data are quasi-constant over the entire Field Of View (FOV), which is an advantage of this technology. Although radar data will never get a comparable data density like optical sensors, settling time and convergence of filters and dynamic parameters like relative speed will be faster, more mature and robust [6].

1.1.2 Environment mapping

When designing the mapping algorithm, a distinction can be made between mapping the static and dynamic environment around the vehicle. The static environment includes all entities and objects around the ego vehicle that have no relative motion with respect to the fixed world. Entities and objects with such relative motion are considered part of the dynamic environment. For example, all traffic in a driving scenario is considered part of the dynamic environment. This section introduces environment mapping frameworks in which both the static and dynamic environment are included. Representation of the dynamic environment can be improved by providing filter estimates, which is detailed in the next sections.

For any environment perception approach it is important to decide on a well suited map representation. In robotics map representations can be roughly divided in two categories, namely topological and metric maps. Topological maps are simplified, high-level representations of the environment, which provide information about spatial relations between entities without providing exact locations in space. A great example of this are the transit maps used in subway systems throughout the world, these maps illustrate the routes and stations in an easy to understand overview, without correctly indicating exact locations or

distances. Metric maps, on the other hand, provide specific, detailed geometric information about the environment and are therefore more useful to autonomous vehicles [7]. These maps are better suited to accommodate for exact measurement information and include uncertainties, e.g. from measurements or localisation of the ego vehicle, by providing precise locations and distances.

Another distinction can be made between local, vehicle-centric, and global, world-centric mapping. Local maps are given relative to a coordinate frame fixed to the ego vehicle, whereas global maps are given relative to a world-fixed reference frame. An example of the latter is a global spatial reference system like GPS coordinates, e.g. WGS 84. World-centric mapping is mainly employed for tasks like navigation or path planning, while vehicle-centric approaches are used for piloting tasks such as collision avoidance [7]. Both approaches have to consider changes in vehicle location over time, however local maps only have to consider the relative motion of the vehicle. Additionally, in this framework the coordinate transformations between sensor positions and the map's coordinate frame remain constant, as the sensors are attached to the vehicle. A downside of local maps is that these tend to struggle to distinguish between parts of the environment that are similar, e.g. with comparable velocity to the ego vehicle or closely grouped objects, and that they are of little use after the vehicle has left a particular area [8]. Especially compared to global maps, as local maps generally only make a single pass through an environment. A global map can be made from local maps, however this is challenging due to uncertainties from localisation and measurements.

A common metric approach is to represent the environment in a spatial lattice map, with occupancy states defined for each cell. Such maps are aptly named occupancy grid maps and were introduced by A. Elfes [9], as shown in Fig. 1.2. The main advantage of these maps is the ease of incorporating measurement and pose uncertainties in a multi-view and multi-sensor scenario. A basic approach is through inference theory by Bayesian estimation mechanisms employing stochastic sensor models [10]. In this theory multiple estimates of the occupancy probability for each cell are fused by their respective sensor measurement uncertainties, characterised through a Gaussian noise model with zero mean and a specified variance. Alternatives based on the Dempster–Shafer theory as the credibilist approach [11] are also commonly used. Occupancy grid algorithms are not limited to Gaussian noise, but can accommodate noise of any distribution. Furthermore, the algorithm is incremental with relatively simple update equations that incorporate new information in a current map through addition of log ratios.

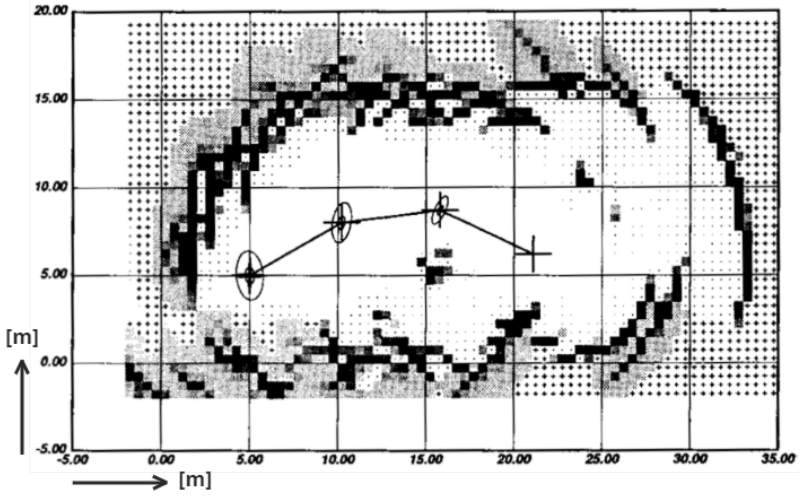


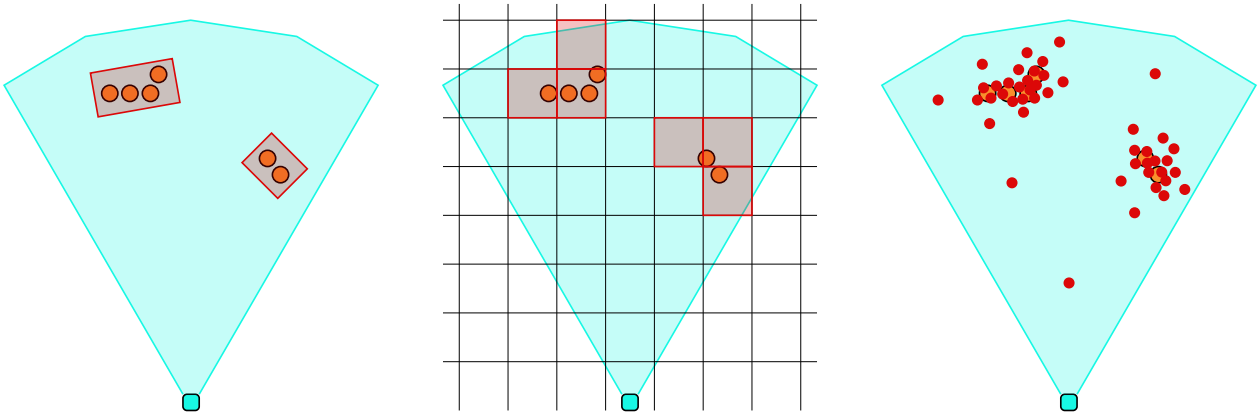
Figure 1.2: A global 2D occupancy grid map, centered around the ego vehicle in a bird's eye view, made from subsequent environment scans by a moving robot [9]. The representation indicates occupancy probability by shades of grey and pose uncertainty around the vehicle, *(+)*.

Commonly, occupancy maps only contain information at a certain height from the ground in a 2D map format. However, with the availability of vertical information more comprehensive maps can be made. Different choices are available which vary in complexity and computational requirements, like 2.5D elevation maps [12, 13], which incorporate height information and corresponding variance as additional states in the discrete grid. This information could be used indicate occluded areas with more confidence. Also multi-level or even full 3D maps [14, 15] are thoroughly researched alternatives.

1.1.3 Dynamic environment representation

The previous section detailed how to represent measurements in a suitable environment map. The provided information to build these maps generally only consists of direct measurements. This can be complimented by estimation of the movement of the dynamic objects in the environment. Therefore this section and the next will give background on how filtering techniques can be used to generate estimates of the dynamic objects in the environment. First, this section will detail different representations of the dynamic environment. Then the next section will give background information on filter methods that can be used to estimate propagation of this dynamic environment over time.

Dynamic objects in the environment can be distinguished from static environment due to their motion relative to the fixed world. Broadly, there are two approaches in representing dynamic objects in the environment, the parametric and the object-free non-parametric approach. In this work, methods which assign dimensions to objects in their representation are considered parametric. Therefore, object detection and tracking approaches are considered as parametric, while particle- or grid-based approaches are considered non-parametric. Figure 1.3 gives a simplified overview of all three representations.



(a) Object-based (b) Grid-based (c) particle-based

Figure 1.3: Different representations of the dynamic environment, presented in a simplified scenario by the sensor and its FOV (blue), object detections (orange) and the representation (red).

The object-based representation in Fig. 1.3a groups measurements together by data association, identifies them as an object and assigns a bounding box with specific dimensions, qualifying it as a parametric representation. In contrast, a grid-based representation as in Fig. 1.3b identifies cells corresponding to detections and assigns states as properties to these cells, e.g. velocity or occupation. The grid can be updated by migration matrices based on propagation of these dynamic states. Similarly, a particle-based representation as in Fig. 1.3c maintains a particle population over the whole area, with dynamic states like position and velocity assigned to each particle. Particles similar to the detected objects survive through subsequent update, weighting and resampling steps to represent these objects. The main advantage of the

non-parametric representations is that they can represent objects of all shapes and sizes, while parametric representations are limited to predefined object classes.

The object-based representation can be considered more computationally intensive, as it requires algorithms to identify, group and label objects. Their main advantage lies in the possibility to apply specific process models to each of the predefined object classes and the availability of a wide variety of filter methods. These can be used to realise an accurate estimation of their motion. However, this is limited to the amount of object definitions available and relies on correct identification of the object. Non-parametric approaches do not require the object detection and identification steps, but represent measurements directly in their approach. In these representations each grid cell or localised clusters of particles represent occupancy, which is propagated over time by the assigned dynamic states.

1.1.4 Estimation by filters

If measurements are available on the dynamic states of the environment, estimation techniques can be applied to propagate its occupancy. Estimation is often done through filter methods of which there is a large variety. This section will provide an introduction to the most common methods.

The general Bayesian filter algorithms work according to the prediction and update principle, where a prediction about the future states is done using a process model and corrected by measurements. This prediction is an estimation of the future states, which is given as a distribution. Filter approaches can broadly be put in two classes, Gaussian and non-parametric filters. Where the former share the basic idea that beliefs are represented by multivariate normal distributions, while the latter does not rely on a fixed functional form of the posterior [16]. Instead, they approximate posteriors by a finite number of values, each roughly corresponding to a region in state space. Non-parametric approaches can be sub-divided in histogram filters and Particle Filters (PF).

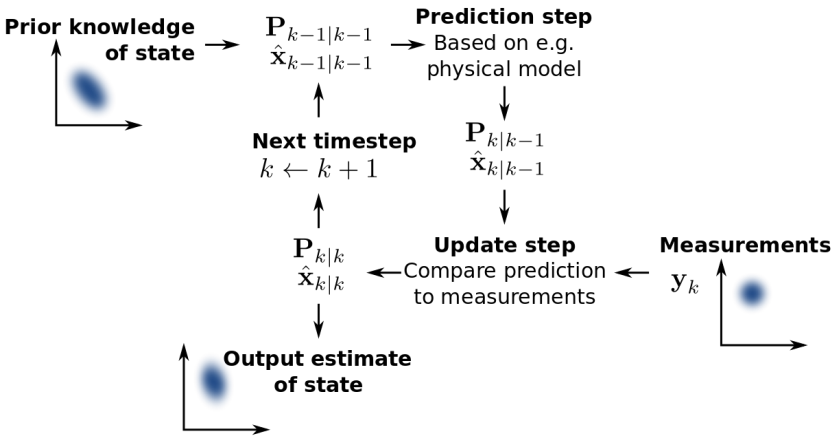


Figure 1.4: Illustration of the basic steps of Kalman filtering; the prediction and update steps. The Kalman filter keeps track of the estimated state of the system and the variance or uncertainty of the estimate. The estimate is updated by a state transition model and measurements. $\hat{x}_{k|k-1}$ denotes the estimate of the sytem's state at time step k before the k-th measurement y_k has been taken into account; $P_{k|k-1}$ is the corresponding uncertainty [17].

The most well known Gaussian filter approaches are the various Kalman filters. The standard Kalman filter is an optimal estimator for linear processes, functioning according to the prediction and update principle as shown in Fig. 1.4, and is often applied to target tracking applications with reasonable success [18]. However, dynamic objects rarely show linear behaviour and thus different approaches can be necessary. Instead, the Extended Kalman Filter (EKF) or the Unscented Kalman Filter (UKF) [19] can be used to estimate non-linear processes. The EKF can handle non-linear functions in its state transition or observation models by linearising through computing the Jacobian matrix around the current estimates. However, for highly non-linear functions the EKF does not perform well either and the UKF can be used. This adaptation gathers a set of sample points of the prior distribution and propagates them through the non-linear functions to get sample points of the posterior. From those propagated points the mean and covariance estimates of the posterior distribution are derived.

Alternatively, several variations of non-parametric filter approaches are available, each with their own advantages. Histogram or grid-based filters like the Bayesian Occupancy Filter (BOF) [20] are applicable to grid-based environment representations. Alternatives like the Sequential Monte Carlo BOF (SMC-BOF) [21] run on a similar grid-based representation, but make use of PFs to estimate the velocity and occupancy of cells. PF approaches are often applied to unstructured, non-parameteric representations of the environment, like the Rao-Blackwellized PF (RBPF) in the research to multiple object tracking in traffic scenarios by Vatavu et. al. [22]. Similarly, a particle implementation of the Probability Hypothesis Density (PHD) filter [23] and its variations, the Gaussian Mixture PHD (GM-PHD) [24] and the PHD/Multi-Instance Bernoulli (PHD/MIB) [25] filters have been used in studies to multi-target tracking applications, with simulated results. Additionally, hybrid filter schemes are available that can switch between several process models for objects, like the Interacting Multiple Model (IMM) filter [26]. These are often applied to object-based, parametric environment representations.

To estimate the movement of the dynamic environment over time some form of velocity information is required. This can be obtained by direct measurements or through deduction by optical flow methods or temporal lattice maps. The latter can capture the time-varying nature of the environment by comparing subsequent environment scans [27, 28], but is dependent on sensor and processing frequencies and is prone to misidentifying fast- and slow-moving objects, as well as previously undetected static objects as moving objects. [29]. However, these are common methods to derive dynamic information from camera and lidar data. Alternatively, direct velocity measurements are available from suitable sensors. An example is from a radar sensor by the Doppler effect, which is available in the i-CAVE project.

1.2 Challenges

The previous sections gave an introduction to the i-CAVE project and a brief theoretical background of environmental perception and mapping. In this section the current challenges in this field are discussed. Challenges relevant to the project are identified and a problem description for this research is given.

First the available exteroceptive sensor set was introduced in Section 1.1.1. Processing raw radar, camera or lidar data are all extensive fields of study. However, in this work it is assumed that processed data is available, so processing raw data of these sensors is kept outside of the scope of this work. But in short, the current challenges in that field are in improving the state-of-the-art image processing algorithms [2], especially 2D algorithms for monocular cameras, in improving accuracy and reducing hardware costs.

For any environment mapping approach it is important to consider which framework to use to represent the environment, as introduced in Section 1.1.2. Finding an optimal environment representation is still an active field of research and representations seeking to improve quality, performance or both are continuously proposed. In his study M. Schreier [30] compared various representations with different levels of ab-

stractness on multiple criteria to asses the quality of each and compare their feasibility for real-time use. Quality is assessed by comparing mapping properties like level of detail, represented dimension, capability of handling and describing moving objects and free space describability versus performance indicators like compactness, online performance and sensor independence. The study concludes by stating that sparse dynamic object maps combined with dense 2D grid maps seem to provide a sufficient level of detail to realise a large variety of future ADAS and ADS applications. However, the author considers the free space representation by discrete cells sub-optimal and proposes his own continuous Parametric Free Space (PFS) map to increase compactness for transmission and interpretation. Instead of modelling the world by discrete cells, the PFS map models the free space by a combination of a parametric curve and geometric primitives in a continuous, 2D bird's eye view.

Mapping a dynamic environment from a moving vehicle requires extra care and is an active field of research. The representation of the dynamic environment by parametric or non-parametric approaches, as introduced in Section 1.1.3, requires consideration in the development of an environment mapping approach. Additionally, the position estimation process of the ego vehicle was identified as a challenge early on [31]. Nowadays it is often solved through odometry approaches, by estimates derived from GPS data or by matching features of multiple environment scans over time [32]. In all of these approaches pose uncertainty is encountered, which is important to take into account for localisation and mapping [8]. Lastly, estimating the propagation of the dynamic environment poses a challenge. Section 1.1.4 gave an introduction to various estimation methods and object tracking approaches. Current research in the field is focused on improving longevity of the estimates, mainly in long-term tracking, which is often made difficult by occlusions or cluttering of multiple objects.

1.3 Research objectives and contributions

The main goal of this study is to research an environment mapping framework to support ADAS and ADS algorithms in the i-CAVE project. More specifically, an approach to map the local environment around a moving vehicle, using data from radar sensors and cameras. The mapping framework must be able to support multi-view and multi-sensor data from several sensor types, while incorporating measurement and pose uncertainties. Additionally, it should properly represent and estimate the occupancy resulting from the dynamic environment, without using target detection and tracking methods. Lastly, the research should focus on using available velocity measurement information to improve the mapping. This section will cover the research objectives and the contributions of this study.

For the considered application a metric representation, as detailed in Section 1.1.2, is deemed the most suitable. This approach can process data from different sources and incorporate uncertainties in a straightforward manner. The occupancy grid mapping provides a suitable candidate framework. It creates a map by tessellation of the environment in grid cells, in which occupancy is represented as a probability per cell. It was already introduced in Section 1.1.2 as a well established and thoroughly researched environment mapping framework. The framework is widely used in robotic applications and is also extensively researched for use in ADS [33, 34]. Research on it remains on detection and representation of free space [35, 36], fusion in multi-view and multi-sensor scenarios [37, 38, 39, 40, 41] and methods for more efficient, online computation methods [42]. This ensures that plenty of literature on the topic is available. Considering this and the conclusion reached by M. Schreier in his evaluation [30], the occupancy grid mapping framework is chosen as a starting point to develop the environment mapping algorithm.

The dynamic environment will be represented by non-parametric approaches due to their flexible nature to represent object of all shapes and types. Instead of representing actual objects, the environment is represented as an occupancy. This occupancy results from the positioning and movements of the dynamic objects. These are either directly measured or are estimated by filter methods and derived from the chosen

representation. Most of the introduced filter methods rely on the update and correction principle. In this update step a process model is often applied to propagate objects or in this case the dynamic environment by estimates of their velocity. This research will choose a suitable filter method and focus on using velocity measurements to improve the accuracy of the applied estimation method. The Doppler velocity measurements of the available radars are used to this end.

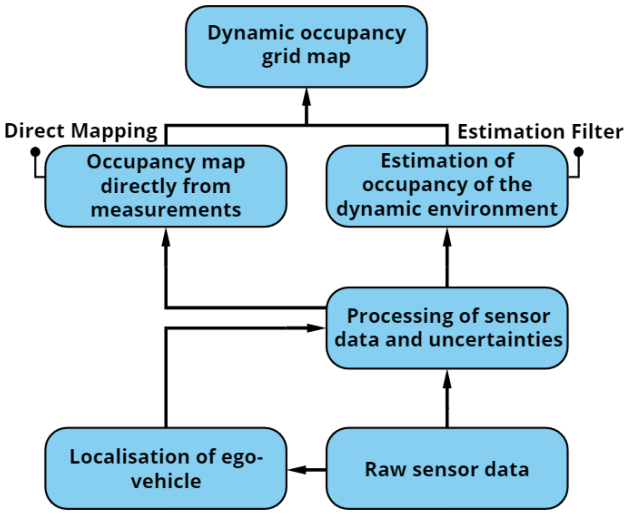


Figure 1.5: Outline of intermediate goals to realise the dynamic occupancy grid map.

By combining the occupancy grid map with the filter estimate, a dynamic occupancy grid map is created. In this representation the information of the occupancy grid map, resulting directly from measurements, is combined with occupancy estimates from a non-parametric filter approach, representing the dynamic environment. The estimation from these filters can be improved by including velocity measurement information, to make more accurate predictions. Figure 1.5 shows the broad outline of the envisioned environment mapping approach. As in the occupancy grid method, each cell will be represented by a probability of being occupied. This study will focus on studying a concept of the described method and focus on the inclusion of velocity measurement information in the filter method. As such, the main research question is formulated as:

Main research question:

"Can dynamic environment mapping on a moving vehicle be improved by incorporating velocity measurements?"

Related to the main research question, several contributions of this research are identified. These will serve as intermediate goals and milestones to the realisation of the main objective of studying a concept dynamic environment mapping approach.

Research contributions:

- Development of a concept dynamic environment mapping approach.
- Offering a mapping framework capable of processing multi-view and multi-sensor data of several sensor types, while incorporating measurement and pose uncertainties.
- Development of a concept non-parametric filter approach, that includes velocity measurement information to improve prediction of approximated occupancy in the environment map.
- Analysis of this concept non-parametric filter approach on performance and accuracy.

1.4 Report outline

This report will continue in Chapter 2 by explaining the concept environment mapping algorithm and the simulation setup to analyse it. The chapter starts by giving an overview of the simulation setup and explains each of the components of the algorithm in the remaining sections. To analyse the algorithm an extensive evaluation is setup, which is explained in Chapter 3. Here the specific scenarios are listed, each with their challenges to the algorithm, and the evaluation metrics are explained, which enable objectively analysing the performance. Then in Chapter 4 the results of the simulations are evaluated. Based on these results recommendations on improvements for the algorithm are given. The report concludes with the main findings and recommendations in Chapter 5.

2 Environment mapping algorithm

This chapter details the concept environment mapping algorithm, which is developed in simulation to analyse it in specific and reproduce-able scenarios. Section 2.1 details this by giving an overview of the simulation setup, explaining how the scenarios are generated and how a sensor model is used to simulate detections. Then the main components of the algorithm are detailed in Sections 2.2 through 2.4. First the direct mapping and estimation filter components are explained. Then in the last section the combination of the occupancy estimates from the filter method with the occupancy grid map is explained, by which the dynamic occupancy grid map is created.

2.1 Simulation environment

The simulations will be performed in the Matlab® programming language. First an overview of the different components in the setup is given in Section 2.1.1. More elaborate explanations of the components are given in other sections of this chapter. Additionally, this section describes how the environment scenarios are simulated in Section 2.1.2 and how measurements are generated from these scenarios by a sensor model in Section 2.1.3, which is specifically programmed for simulation of the algorithm. Lastly, an extension of the scenario setup is discussed. The described simulation is setup in a 1D format and the extension proposes a method to simulate multiple, simultaneous object detections in this dimension.

2.1.1 Overview setup

An overview of the simulation setup is given in Fig. 2.1, in which three sets of components are encircled. The main components are in the algorithm part, which are also the components indicated when referring to the algorithm in the remainder of this report. They are the direct mapping component, the estimation filter and a component which combines both outputs to a dynamic occupancy grid map. Both former components have to be initialised, which is done through creating a grid map with unknown occupancy and establishing an uninformed prior distribution for the chosen estimation filter. Then there are the scenario simulation and sensor models components, which combined generate the measurements for the algorithm.

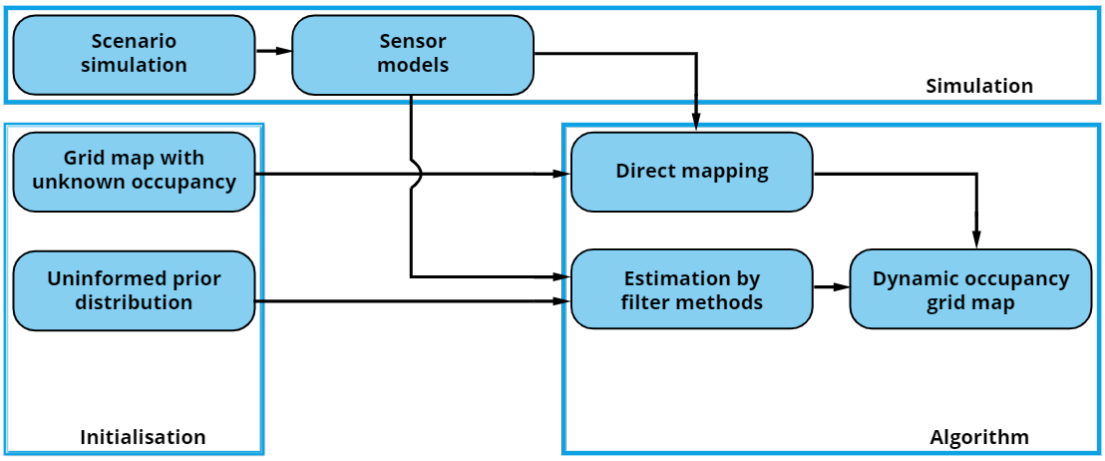


Figure 2.1: Overview of the components in the simulation setup.

A scenario is initialised by defining the velocities and initial positions of objects in the area surrounding the vehicle. These objects are propagated over time and specific velocities and initial positions can be assigned to create desired behaviour. The objects are observed by sensor models, which generate the measurements. These are used by the algorithm components to create the dynamic occupancy grid map. An occupancy grid map is updated in the direct mapping component, by incorporating measurements in the previous map through an update equation. In the estimation filter a prior distribution estimating the occupancy of the environment is updated with the new measurements as well. Both outputs are then combined to the dynamic occupancy grid map, which is the end result of the algorithm.

2.1.2 Scenario generation

Simulation scenarios are defined by describing object positions and movements in the area surrounding the vehicle. The scenarios are setup by assuming a stationary ego-vehicle and initialising dynamic objects around it. The first simulations are performed in a 1D scenario setup, so the results are easier to visualise and analyse. Figure 2.2 aids in understanding the 1D setup, which can be considered as taking a narrow slice out of a 2D sensor FOV and considering only that angle. In the left-hand side of Fig. 2.2 a top-down, 2D view of the area in front of a vehicle is given. This area is mapped in polar coordinates by a range and angle, via the coordinates (θ, r). The narrow 1D beam is assumed to span only a small angle, θ_i, and can be described by the range, r, only. It is in this 1D setup that the first simulations are performed.

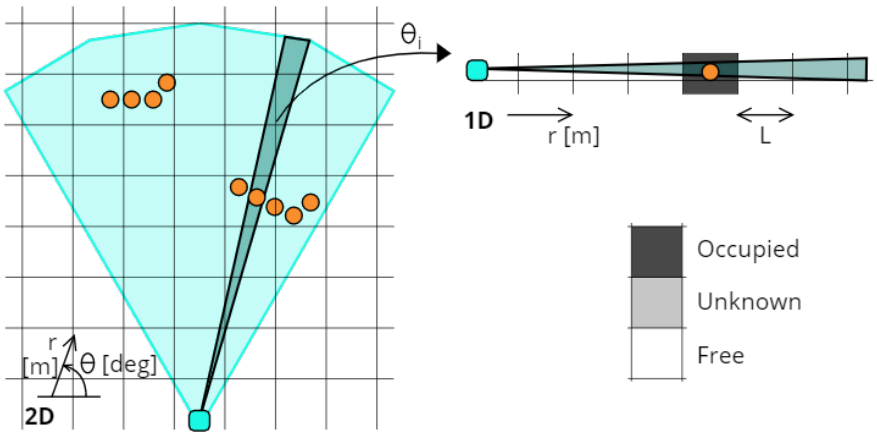


Figure 2.2: Illustration of a 1D scenario setup, by sensor FOV (blue) and detections (orange).

The objects in the scenarios are given properties to simulate specific behaviour, by assigning dimensions as well as velocity and acceleration profiles. Several object types are defined, which are given in Appendix B in Tab. B.2 for reference. Assigning dimensions is important to the generation of detections by the sensor models, which is explained in the next section. The defined velocity and acceleration profiles are used to create specific behaviour in the scenarios, i.e. by defining object's movement.

The scenarios are generated by initialising objects in the environment, with predefined or randomised position and velocity. A basic scenario is displayed in Fig. 2.3, in which a single object travels through the area at a constant velocity. The van in the example moves through the range at a velocity of 8.0 $[\frac{m}{s}]$ and is deleted when it moves out of the range. A single radar is used to generate detections. Results of simulating with this scenario can be analysed easily, as the actual position of the object can be checked in a straightforward manner. In Section 3.1 more complex scenario setups are explored and explained.

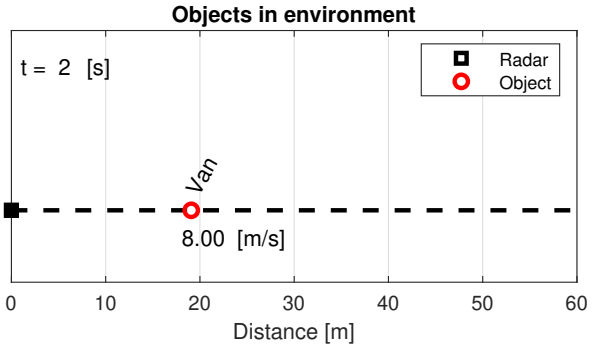


Figure 2.3: Example of the constant velocity scenarios, 2 $[s]$ in simulation. The snapshot shows the radar beam at the vehicle, i.e. the coordinate system origin, in black. All generated objects are displayed in red and have their type and velocity printed.

2.1.3 Sensor models

Sensor models are implemented to generate measurements from the simulated scenarios. A radar sensor model is explained here, as it is the first one to be implemented. Data from other sensors can be implemented in a similar manner, with models specific to each sensor type. When running the algorithm on the test vehicle, input data is assumed to be delivered as processed. So as detections by position coordinates with velocity information, if available. The output of the model is similar, however is only for simulation.

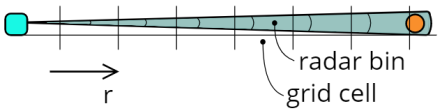


Figure 2.4: Illustration of radar bins and grid cells for a radar measurement beam in 1D.

The radar model generates measurements based on probabilities for detection and false alarm. These are evaluated along the single measurement beam at regular time intervals. This can be visualised as in Fig. 2.4, where the beam is split in bins, independent of the underlying grid cell structure. The size of these radar bins is specific to the sensor used. The model determines if a detection occurs at each bin along the range, according to the logic in Tab. 2.1, which is also given by the pseudo-code in Alg. 2.1. If a bin is occupied by an object, a detection occurs with probability p_d and this detection is missed with probability $1 - p_d$. The bins closest to the radar origin are evaluated first and only the closest detection is returned. Additionally, for each bin a false detection can occur with probability p_{fa}, i.e. the probability of false alarm. If no detections occur along the beam, no measurements are passed on to the algorithm.

Probabilities	Detection	No detection
Occupied position	p_d	$1 - p_d$
Empty position	p_{fa}	$1 - p_{fa}$

Table 2.1: Probabilities for occurrence of detections in the radar sensor model, based on the probabilities of detection (p_d) and false alarm (p_{fa}) and their compliments.

Algorithm 2.1 Pseudo-code of the generation of detections by the radar sensor model

```
function [DetectionList] = RADAR SENSOR MODEL(map, ObjectList)
    ObjectList = SortFCN(ObjectList)                ▷ Sorts objects on distance to radar origin
    for i = [1 : 1 : r_end / BinSize] do            ▷ Evaluate for each radar bin along the range
        if ObjectList(r_bin[i]) ≠ Empty then        ▷ Check if there is an object at the current bin
            Calculate p_d for target
            if rand(1) < p_d then
                ADD ObjectList(r_bin[i]) to DetectionList
            end if
        else if rand(1) < p_fa AND ObjectList(r_bin[i]) = Empty then
            ADD false detection to list, by randomised position
                and velocity within the current bin
        end if
    end for
end function
```

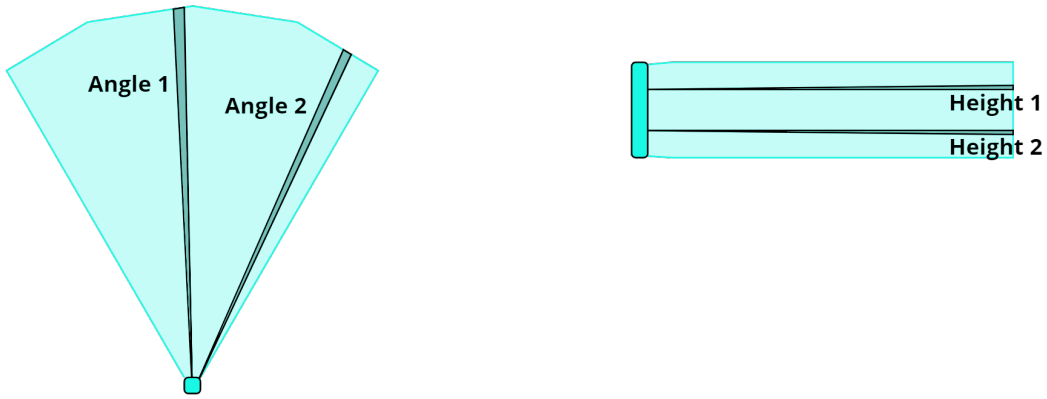
The sensor model calculates a p_d for each target based on its range, r, i.e. distance to the sensor's origin and the target's visible Radar Cross Section (RCS), calculated from its dimensions as specified in Tab. B.2. It does this by estimating the required Signal-to-Noise Ratio (SNR) to detect a target with a reference RCS at a reference r with specified reference p_d and p_{fa} values. This is done by Schnidman's equation [43, p. 337], which is a series of equations that yield an estimate of the SNR. The required SNR for the actual target with its RCS at range r is scaled by Eq. (2.1), as an approximation of the radar range equation. Then the p_d value is extracted from the Reciever Operating Characteristic (ROC) curve corresponding to the given p_{fa} at the calculated SNR value [43, pp 298–335]. The p_d is calculated at each time step due to the change in r of the target, however the p_{fa} value is assumed to be constant over the whole measurement range. All the mentioned reference values are design parameters specific to the radar sensors used.

$$SNR = SNR_{ref} + 40 \cdot log\left(\frac{r_{ref}}{r}\right) + 10 \cdot log\left(\frac{RCS}{RCS_{ref}}\right) \tag{2.1}$$

2.1.4 Extension

The limitations of the 1D setup is that only a single measurement can be generated at any time step. In scenarios with multiple actors a sensor should register multiple, simultaneous detections. Therefore, an extension to the simulation setup is proposed in this section. This extended setup will be referred to as 1D+ and will enable to test the algorithm with multiple, simultaneous detections.

In a 2D setup multiple measurements would naturally result from measurements at different angles, e.g. as in Fig. 2.5a. However, the goal is to test the current implementation without extending the algorithm to a higher dimension. As such, more measurements need to be added to the 1D setup. A proposal is made to extend the setup with an additional measurement beam on top of the first one. This can be thought of as having multiple measurement beams at different heights, as in Fig. 2.5b. Measurements are along the same direction, however they do not interfere. By adding height as property to the objects, some are only detected by one of the beams. This generates multiple simultaneous measurements in 1D, however it does not capture complete information on the height of objects. Therefore, this approach is referred to as 1D+ and is not considered as a 2D, as it lacks complete information in the second, height, dimension.



(a) Two measurement beams in a top-down projection of the FOV of the sensor, i.e. in a 2D plane.

(b) Two measurement beams in the 1D+ setup, considered as measurements at different heights in the FOV.

Figure 2.5: Illustration of multiple measurement beams in a 1D+ setup, compared to 2D.

In Fig. 2.6 an example is given of the 1D+ setup, with two radar measurement beams arbitrarily placed at different heights. As the two objects have different heights, the top measurement beam (2) will only detect the truck in this example. The bottom beam (1) will only return detections of the sedan as it is closer. So in this setup multiple measurements can be taken simultaneously. Additionally, it enables simulation of more challenging scenarios, to better analyse the algorithm. For example, in the situation in Fig. 2.6 the sedan will overtake the truck in the future due to its higher velocity. This then fully occludes the sedan from view of both measurement beams. However, due to previous measurements on the sedan, a filter estimate should be retained on the occupancy due to this object. This is an example of a more challenging scenario that can be tested in the extended 1D+ setup.

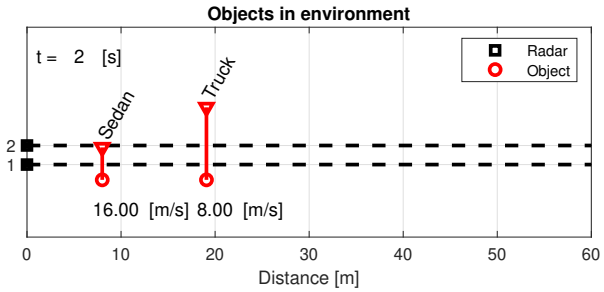


Figure 2.6: Example of a 1D+ scenario setup, 2 $[s]$ in simulation. The setup shows the radar beams at the vehicle, i.e. the coordinate system origin, in black. The two generated objects are displayed in red and have their type and velocity printed.

2.2 Direct mapping

Now the simulation environment is introduced, the components of the mapping algorithm can be explained. This section details the direct mapping component which processes measurement information to an occupancy grid map. By tessellation of the surrounding environment with grid cells, a grid map is created. Measurements are converted to occupancy probabilities of corresponding grid cells. These probabilities are cumulatively updated by an update equation, to estimate the occupancy of the environment.

2.2.1 Initialisation

The first step in the algorithm is to discretise the area to be mapped into a regular grid of cells, c. In 1D this means dividing into segments of specified lengths, L. In this dimension the grid cells are aligned along the one coordinate axis and fully described by their centre position and length. An example of this is shown in Fig. 2.7, for both a 2D and 1D grid map, m.

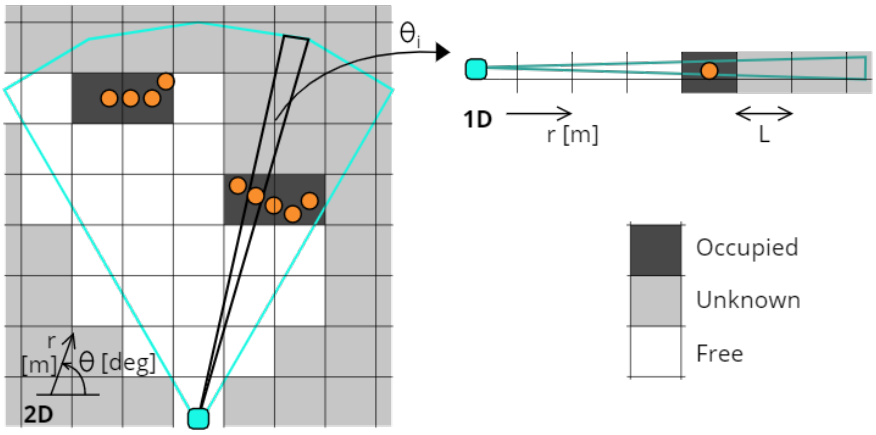


Figure 2.7: Illustration of an occupancy grid map in both 1D and 2D, by sensor FOV (blue) and detections (orange). Occupancy states of grid cells are indicated by gray scale.

As indicated in Fig. 2.7, each grid cell is assigned an occupancy state, i.e. occupied, free or unknown. These states are derived from the occupancy probabilities assigned to each cell, based on updating the grid with measurements. How measurements are translated to these probabilities and are updated over time is detailed in the remainder of this section. As during initialisation no measurements are available yet, all cells are assigned an "unknown" occupancy. This corresponds to an occupancy probability of 50 [%]. When the algorithm is run and measurements are received, these probabilities will be updated cumulatively.

2.2.2 Occupancy probabilities

After initialising the grid map can be updated with measurement information. This is done according to an iterative process. The first step is to convert measurement information to occupancy probabilities for the corresponding grid cells. The mapping algorithm receives pose estimates and measurements of the vehicle's surroundings as inputs, denoted by $x_{1:t}$ and $z_{1:t}$, respectively. These are known for the first time step up to the current time t. The measurements are obtained from sensors mounted on the vehicle and are given in coordinates relative to the vehicle's pose. As they have a fixed relative position, the coordinate transforms between sensor and vehicle pose remains constant throughout the simulation. For now the algorithm assumes a static vehicle, i.e. x_t is constant, but changes in pose should be incorporated later on.

The definition of an occupancy grid by A. Elfes [9] states that the grid representation employs a multidimensional tessellation of space into cells, where each cell stores a probabilistic estimate of its state. In this case that state is the occupancy probability. In the discussion below, the occupancy grid is modelled as a Markov random field of order 0, so the individual cell states can be estimated as independent random variables [9]. This is referred to as the conditional independence assumption. It can be justified conceptually as, in general, there are no casual relationships between states of different cells. Additionally, from an engineering point of view the resulting map is adequate for the considered task. Without this assumption, the problem would change into an estimation problem requiring to estimate the conditional probabilities of

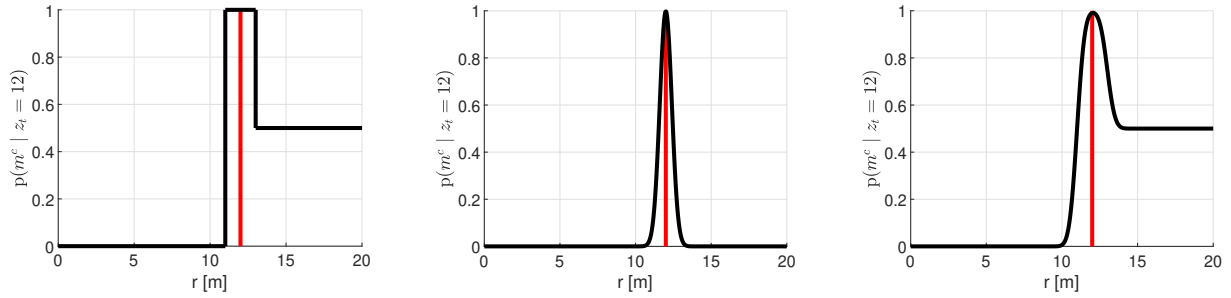
all possible configurations over the full grid. For 2D or higher dimensional maps, this would result in huge amount of possible grid configurations and contribute to slowing down the algorithm significantly.

If a Ground Truth (GT) map is made of the environment, it would result in a binary map with either occupied or free states for each grid cell. Instead the mapping algorithm uses occupancy probabilities to include uncertainties from measurements and other sources. The map can be generated with either probability for a cell to be occupied or free, as indicated by the subscripts o and f, respectively. Both probabilities are compliments of each other and are noted as in Eq. (2.2). For example, the probability for a cell, c, to be occupied is $p_o(m^c|z_{1:t}, x_{1:t})$, while for free it is $p_f(m^c|z_{1:t}, x_{1:t})$. However, only the probability for occupied, p_o, is used, as either one gives sufficient information to derive the state of a grid cell. Additionally, an unknown state was mentioned for initialising the grid cells. This state is defined for when both probabilities are equal, i.e. $p_o(m^c|z_0, x_0) = p_f(m^c|z_0, x_0) = 0.50$ and the state cannot be derived.

$$p_f(m^c|z_{1:t}, x_{1:t}) = 1 - p_o(m^c|z_{1:t}, x_{1:t}) \tag{2.2}$$

2.2.3 Inverse sensor model

As explained previously, measurements are generated from the first observed obstacle along the beam. All cells intersected by that beam must be updated with a new occupancy probability. This can be done by employing an inverse sensor model, which incorporates measurement uncertainties. The proposed method in this section is based on the inverse sensor model with Gaussian noise as proposed by D. Joubert [8]. A major advantage of using this model is its flexibility. The proposed model shapes the occupancy probabilities along the beam based on a convolution of an ideal inverse sensor model and a Probability Density Function (PDF) by a Gaussian distribution. However, the function can be shaped to any noise distribution. Therefore, it can be adapted to a large spectrum of sensor types. Additionally, the model is worked out for a single beam, which can be extended to higher dimensions in a straight-forward fashion.



(a) PDF of the ideal inverse sensor model

(b) PDF with a Gaussian distribution

(c) PDF of the inverse sensor model with noise

Figure 2.8: Different methods to compute occupancy probabilities along the measurement beam.

Consider the ideal inverse sensor model as shown in Fig. 2.8a. In this example an obstacle is detected at 12 $[m]$, i.e. $z_t = 12$, so according to the sensor the first detected object is located at that distance. Then the inverse sensor model uses a piecewise defined function, $g(r)$, to compute the occupancy probability along the measurement range, r. The ideal model returns a probability value corresponding to certainly free, $p(m^c|z_t) = 0$, to all cells in front of the measurement, a value corresponding to certainly occupied,

$p(m^c|z_t) = 1$, to the cell containing the obstacle, and a value corresponding to unknown $p(m^c|z_t) = 0.5$ to all cells further away than z_t, as the first object may not be the only one along that beam [8]. Here $g(r)$ is:

$$g(r) = \begin{cases} 0, & \text{if } r < z_t - \frac{L}{2} \\ 1, & \text{if } z_t - \frac{L}{2} \leq r < z_t + \frac{L}{2} \\ 0.50, & \text{otherwise} \end{cases} \tag{2.3}$$

The ideal model does not properly reflect measurement uncertainties in the computed occupancy probabilities. Consider a measurement in which the noise is normally distributed around the observed value. In this case the occupancy probability can be computed using a PDF with a Gaussian noise distribution, i.e. $f(r; z_t, \sigma^2)$, as is shown in Fig. 2.8b. This is expressed as:

$$f(r; z_t, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{(r-z_t)^2}{2\sigma^2}} \tag{2.4}$$

Here the standard deviation, σ, can be obtained from sensor specifications. This representation of noise can be incorporated in the ideal inverse sensor model, resulting in an inverse sensor model that adequately represents measurement noise. To realise this, the work by D. Joubert [8] presents a convolution of the ideal inverse sensor model and the PDF. The standard convolution between two functions, $f(t)$ and $g(t)$ is expressed as:

$$f(t) * g(t) = \int_{-\infty}^{\infty} g(\tau) f(t - \tau) d\tau \tag{2.5}$$

The convoluted function can be derived with Eqs. (2.3) and (2.4). This results in the Gaussian inverse sensor model $(f * g)(r)$.

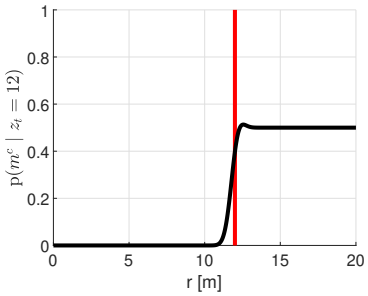
$$(f * g)(r) = -\frac{1}{4}\text{erf}\left(\frac{r - z_t - \frac{L}{2}}{\sqrt{2}\sigma}\right) + \frac{1}{2}\text{erf}\left(\frac{r - z_t + \frac{L}{2}}{\sqrt{2}\sigma}\right) - \frac{1}{4}\text{erf}\left(\frac{-z_t}{\sqrt{2}\sigma}\right) \tag{2.6}$$

In this formula the error function is used, which is defined as:

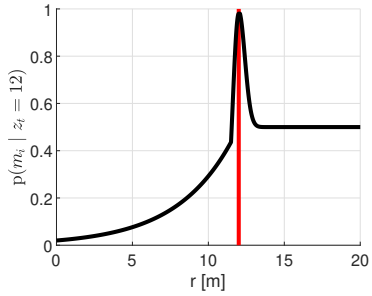
$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \tag{2.7}$$

The convoluted function $(f * g)(r)$ takes the grid cell size, L, and standard deviation, σ, as input parameters. These parameters influence the shape of the Gaussian inverse sensor model, especially the height and width of the peak around the measurement value. In the example of Fig. 2.8c the occupancy probability is plotted according to this model. Here a coarse cell size of $L = 2.0$ $[m]$ is used, with reasonable values for the standard deviation.

When the inverse sensor model is implemented with a finer grid cell size, e.g. $L = 0.2\ [m]$, the shape changes significantly and results in the shape of Fig. 2.9a. In this shape the occupancy probability is too low around the measurement and in general the shape is not as desired. Therefore, a shaped function is proposed to recreate the shape of Fig. 2.8c, with parameters as used in the algorithm used in this study.



(a) PDF of the inverse sensor model with noise



(b) PDF of the shaped function

Figure 2.9: Computation of occupancy probabilities, with algorithm parameters.

In the proposed function, the occupancy probability p_o is shaped by a piecewise function, similar to Eq. (2.3). It combines an exponential, a normal distribution and ends in a constant value. The pseudo-code of this shaping function is given in Alg. 2.2. It computes the exponential in $y_1(r)$, where values a and b are chosen such that the exponential reaches a value of 0.50 at the detected range value, i.e. $r = z_t$. A Gaussian PDF is used to shape a normal distribution around the detection in $y_2(r)$, which is scaled such that its peak value equals 1.0. Lastly, $y_3(r)$ scales the PDF of $y_2(r)$ to have a peak value of 1.0 with a minimum value of 0.50. Then for $r \leq z_t$ the maximum of $y_1(r)$ or $y_2(r)$ is taken as output, while for $r > z_t$, $y_3(r)$ is taken. Finally, output $y(r)$ gives the shaped occupancy probability over the full range.

Algorithm 2.2 Shaped function for the occupancy probability in pseudo-code

function $[y]$ = SHAPE PROBABILITY(r, z_t, σ)
  $y_1(r) = b \cdot e^{(a \cdot r)}$
  $y_2(r) = f(r; z_t, \sigma^2)$ ▷ Normal distribution, by Gaussian PDF (Eq. (2.4))
  $y_2(r) = \frac{y_2(r)}{\max(y_2(r))}$
  $y_3(r) = 0.50 \cdot y_2(r) + 0.50$

  $y(r) = \max(y_1(r), y_2(r))$
  for $r = [r_0, \ldots, r_{end}]$ **do**
    if $z_t < r$ **then**
      $y(r) = y_3(r)$
    end if
  end for
end function

Figure 2.9b shows the resulting occupancy probability by the shaped function. In the shape the different elements can be recognised. Due to this shape the occupancy probability is higher at low range values in front of the detection then by the inverse sensor model of Fig. 2.8c. However, this will not pose a problem

due to the cumulative updating, which is described in the next section. The normal distribution is shaped around the detection as desired. In contrast to the results by inverse sensor model as proposed by D. Joubert, in Fig. 2.9a, the occupancy probability reaches a peak value of 1. So the shaped function can recreate a shape similar to the inverse sensor model, with the parameters that will be used in this algorithm.

To illustrate how the shaped occupancy probability is computed during simulation a screenshot of a 1D scenario is given in Fig. 2.10. The scenario setup is equal to the setup given in Fig. 2.6, but with only a single measurement beam. Objects are displayed in the top plot of the figure. Only the closest object is detected by the radar, as the objects behind are occluded from view. That measurement is displayed in blue in the second plot from the top, together with an indication of the radar bins. In the third plot the shaped occupancy probability function is displayed by the green line. This occupancy probability is used to derive the occupancy map in the fourth plot. The next section will detail how this map is derived from a cumulative update equation.

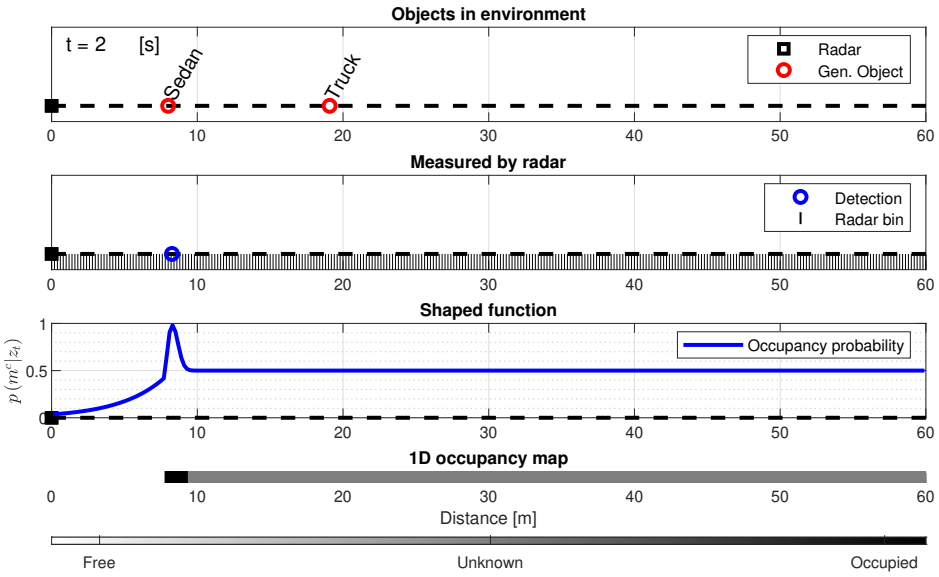


Figure 2.10: Example of a generated occupancy map in the 1D setup. The top plot shows the actual objects, the second plot shows detections with indication of the radar bins, the third plot the shaped occupancy probability function and the occupancy map is derived in the fourth plot.

2.2.4 Update equation

The occupancy probabilities of grid cells in the map, m^c, are updated cumulatively. This is done by an update equation derived from the Bayes' rule, expressed in log-likelihood ratios, which is derived here. It uses the conditional independence assumption between cells, introduced in Section 2.2.2, to enable computation of the occupancy probability for individual cells. Following the work of D. Joubert [8], an additional assumption is made to derive the update equation. Measurements, z_t, are given relative to the vehicle's pose, x_t. By translating the measurements into a global coordinate system, the pose information is included in the data. Therefore, the occupancy probability $p_o(m^c|z_{1:t}, x_{1:t})$ can be reduced to the equivalent notation:

$$p_o(m^c|z_{1:t}, x_{1:t}) \equiv p_o(m^c|z_{1:t}) \tag{2.8}$$

By employing Bayes' rule and assuming independence between measurements, these equations are derived:

$$\begin{aligned} p_o(m^c|z_{1:t}) &= \frac{p_o(z_t|m^c, z_{1:t-1})p_o(m^c|z_{1:t-1})}{p(z_t|z_{1:t-1})} \\ &= \frac{p_o(z_t|m^c)p_o(m^c|z_{1:t-1})}{p(z_t|z_{1:t-1})} \end{aligned} \tag{2.9}$$

$$p_o(z_t|m^c) = \frac{p_o(m^c|z_t)p(z_t)}{p_o(m^c)} \tag{2.10}$$

Substitution of Eq. (2.9) into Eq. (2.10) results in the equation for the probability of a cell to be occupied based previous measurements, $p_o(m^c|z_{1:t})$, as in the equation below. The equation for the probability of a cell to be free, $p_f(m^c|z_{1:t})$, is derived similarly as they are compliments of each other.

$$p_o(m^c|z_{1:t}) = \frac{p_o(m^c|z_t)p(z_t)p_o(m^c|z_{1:t-1})}{p_o(m^c)p(z_t|z_{1:t-1})} \tag{2.11}$$

$$p_f(m^c|z_{1:t}) = \frac{p_f(m^c|z_t)p(z_t)p_f(m^c|z_{1:t-1})}{p_f(m^c)p(z_t|z_{1:t-1})} \tag{2.12}$$

By dividing Eq. (2.11) by (2.12) and taking natural logarithms on both sides [8], the log-likelihood update equation below is obtained. Logarithms are taken to avoid truncation errors for probabilities close to 0 or 1. It also has the advantage that multiplications of probabilities become additions of log-likelihood ratios.

$$\log\left(\frac{p_o(m^c|z_{1:t})}{p_f(m^c|z_{1:t})}\right) = \log\left(\frac{p_o(m^c|z_t)}{p_f(m^c|z_t)}\right) + \zeta \log\left(\frac{p_o(m^c|z_{1:t-1})}{p_f(m^c|z_{1:t-1})}\right) - \log\left(\frac{p_o(m^c|z_0)}{p_f(m^c|z_0)}\right) \tag{2.13}$$

The left-hand side term in Eq. (2.13) is the cumulative log odds ratio representing the posterior probability of occupancy for the respective grid cell. New measurements are added by their probability as the first term on the right-hand side. The posterior probability resulting from previous measurements is combined in the second term. This term includes a decay factor, ζ, decreasing the contribution of previous measurement information over time, with the aim of better representing occupancy from dynamic objects. As a start, this factor is set to a value of $\zeta = 0.5$. Lastly, as introduced in Section 2.2.1, the probabilities assigned to cells when initialising the environment map represent an unknown occupancy state, with value 0.5. These initial values are subtracted in the last term on the right-hand side.

The notation λ_t^c, which denotes the log odds ratio of grid cell c at time t, is introduced. With this notation the ratio can be written as below and the update equation, can be written more brief. The log odds ratio can be converted to a posterior occupancy probability, $p_o(m^c)$. If a posterior occupancy probability is mentioned in the remainder of this work, it indicates the one below.

$$\lambda_{1:t}^{c} = \log\left(\frac{p_o(m^c|z_{1:t}, x_{1:t})}{p_f(m^c|z_{1:t}, x_{1:t})}\right) \tag{2.14}$$

$$\lambda_{1:t}^{c} = \lambda_{t}^{c} + \lambda_{1:t-1}^{c} - \lambda_0 \tag{2.15}$$

$$p_o(m^c) = \left(\frac{e^{\lambda_{1:t}^{c}}}{1 + e^{\lambda_{1:t}^{c}}}\right) \tag{2.16}$$

Table 2.2 summarises the correspondences between the posterior occupancy probabilities and the log odd ratios, with their respective interpretations. For implementation the log odd ratios are limited to a chosen threshold, such that the ratios do not go to $\pm\infty$. Without limiting this, the cumulative updated ratios react slow to changes. For each cell in the map an occupancy state is assigned, according to the interpretations as in Tab. 2.2. These states are visualised in the occupancy map by shades of grey, as in the fourth plot of Fig. 2.10. The legend at the bottom of this figure shows the correspondence between the grey scale and occupancy states.

$p_o(m^c)$	$\lambda_{1:t}^{c}$	**Interpretation**
0	$-$threshold	free
0.5	0	unknown
1	threshold	occupied

Table 2.2: Correspondence between occupancy probabilities and log odds ratios with their respective interpretations. Here $p_o(m^c)$ is the posterior probability of cell c being occupied. [8]

2.3 Particle filter

The previously described occupancy grid mapping algorithm directly uses measurements and translates these to an occupancy probability. This is susceptible to uncertainties in measurement information. By providing state estimation through a filter method, these uncertainties can be corrected. This estimation can be realised by propagation of the occupancy with the available velocity information, as Doppler radar provides both range and velocity measurements. Therefore, this research proposes to use an estimation filter to approximate the occupancy of the environment, as outlined in Section 1.3. This estimated occupancy can then be merged with the map resulting from direct mapping to generate a dynamic occupancy grid map. This section will detail which estimation filter is chosen and how this method estimates the occupancy of the dynamic environment.

When driving around city or highway scenarios, the environment is filled with a wide variety of actors. Instead of representing actual objects, this estimation method aims to represent the environment by distributed occupancy probabilities in a non-parametric representation, as introduced in Section 1.1.3. This allows for the method to represent occupancy from objects of all shapes and sizes. Since the estimated state, the joint occupancy distribution of the whole dynamic environment, will form a complex, non-linear PDF, a filter capable of handling is required. Thus, the problem at hand requires an estimation method capable of approximating complex, non-linear distributions.

Section 1.1.4 introduced several methods and highlighted the PF class as it can approximate large non-Gaussian PDFs without requiring complex stochastic calculus. The PF belongs to a class of non-linear filters, with a broad variety in different types. In this research the Sequential Importance Resampling (SIR) filter, introduced under the name bootstrap filter by Gordon et al. in 1993 [44], is applied. Dense clusters of particles represent occupancy in the proposed approach. Thus the amount of particles per grid cell can be used to derive an occupancy probability similar to the occupancy grid map. However, these clusters are not bound to any underlying grid structure, making this approach independent of the chosen grid map resolution. This section will give an overview of the steps in the filter.

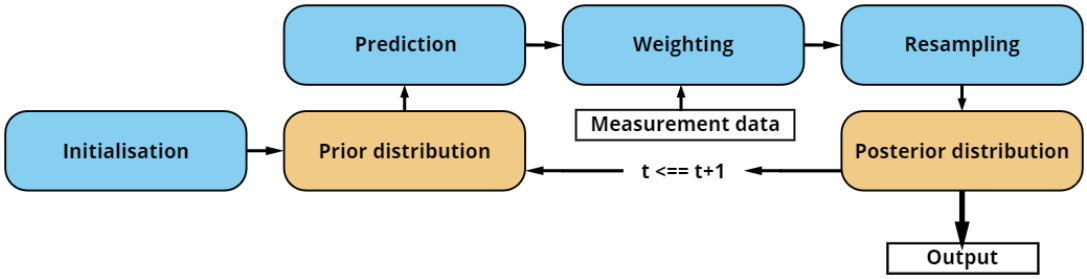
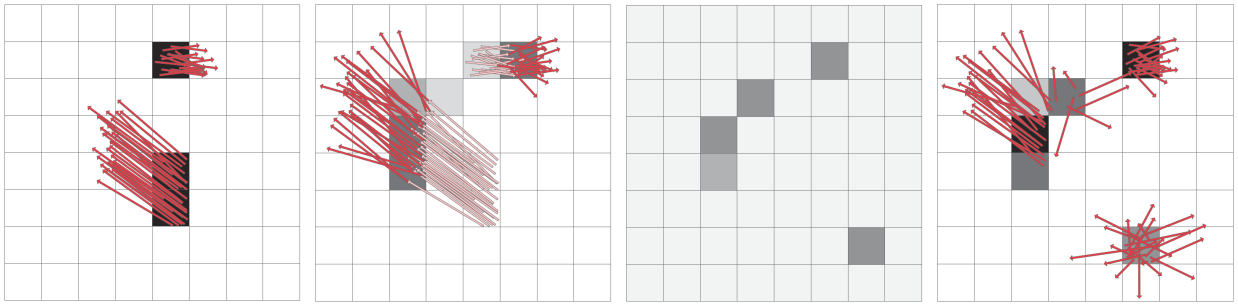


Figure 2.11: Block diagram of the general particle filtering structure.

The PF functions according to an update and correction principle, by the general structure as displayed in the block diagram of Fig. 2.11. A particle population is initialised by an uninformed prior, similar to initialising the occupancy grid map with unknown occupancy states as in Section 2.2.1. This particle population is updated by a process model in the prediction step, obtaining an a new posterior. Using a weighting and resampling scheme this posterior is corrected by received measurements to approximate the current occupancy of the dynamic environment in its distribution. By repeatedly updating and correcting, the posterior will accurately resemble the occupancy of the dynamic environment. Each of the steps, initialisation, prediction, weighting and resampling is explained in detail in this section.



(a) Posterior at time t (b) Prediction (c) Measurements (d) Posterior at time $t+1$

Figure 2.12: A visualisation of particles on an underlying 2D grid structure during different stages of an iteration of the PF, taken from Danescu et. al. [45]. Particles (red) are shown by vectors, representing position and associated velocity. The individual grid cells show occupancy probability visualised in gray scale, similar to Fig. 2.7.

Figure 2.12 gives an example of an iteration of the PF in a 2D setup. At the start of any iteration, the posterior distribution of the previous time step becomes the new prior. Each particle has a position and velocity attribute, which are represented by the arrow's origin for position and direction and magnitude for velocity. In the prediction step these states are updated, as is done between Figs. 2.12a and 2.12b. This prediction is corrected by measurements, which in this example are represented as an occupancy grid, by probabilities per cell, as in Fig. 2.12c. Through weighting and resampling the new posterior is computed. In this example that posterior is shown in Fig. 2.12d. Cells that were predicted without occupancy, but that had measurements on them are assigned new particles with a randomised position and velocity within the cell. A method based on this structure, but with adaptations is used in this research.

2.3.1 Initialisation

During initialisation of the filter a population of particles is generated. The total amount of particles used in this population, N_{total}, is fixed. This amount is chosen, based on performance requirements and computational restrictions and is researched in Section 4.2. A list is used to keep track of the particles by indices, i, which registers two dynamic states per particle, position and velocity, in a state vector, x, and two additional parameters, its weight, w, and corresponding grid cell index, c. Each particle is placed at a random position, r, in the 1D setup with a random initial velocity, v. An initial weight is assigned to each particle according to Eq. (2.18), which is equal for all particles. The particle index, i, is used to keep track of the particles through the update and correction cycles.

$$x^i = \begin{bmatrix} r^i \\ v^i \end{bmatrix} \tag{2.17}$$

$$w^i = \frac{1}{N_{total}} \tag{2.18}$$

2.3.2 Prediction

During the prediction step the dynamic states of all particles of the prior distribution of the current time step are updated, which are equal to the particles of the posterior of the previous time step, as according to Fig. 2.11. Updating is done through a process model with a Constant Velocity Constant Orientation (CVCO) assumption. The dynamic states of each particle are updated according to model below, which is similar to the CVCO process model proposed by [21], but simplified to conform to the 1D states.

$$x^i_{k+1} = \begin{bmatrix} 1 & T_{meas} \\ 0 & 1 \end{bmatrix} x^i_k + \nu \tag{2.19}$$

The CVCO model is given in a discrete representation, as the measurement samples are received at intervals. In this discrete notation, the current time will be referred to as k. By using the measurement time, T_{meas}, the particle states are updated to the next time step, $k+1$. This measurement time is the duration of such interval and is assumed constant. Its value can be determined from sensor properties. A random process noise, ν, is added to the updated states to add stochastic diffusion to the model. These quantities are randomly drawn from a Gaussian distribution of zero mean with covariance derived from sensor properties, which is equivalent to the use of a state transition covariance matrix in a Kalman filter [45].

The updated population represents the new posterior distribution and has to be corrected by the measurement update. This correction is done in two steps. First the particles are assigned a weight, representing the confidence in each particle of it correctly representing the actual occupancy. This is done by several weighting criteria. Secondly, the particle population is resampled. A selection of the updated population is duplicated to a new set, which represents the posterior PDF. This process is referred to as resampling and the selection is made based on assigned weights. Both the weighting and resampling steps are explained in more detail in the next sections.

2.3.3 Weighting

The weight assigned to a particle represents its likelihood of correctly approximating the occupancy due to the dynamic environment. Three weighting criteria are all applied and used to calculate a combined weight per particle. All of these criteria are listed below and are explained in more detail later on.

1. w_r^i - Assigns weight based on the correspondence between the position of a particle and the position of a matching detection.
2. w_v^i - If the particle is located close to a matching detection, weight is assigned based on the similarity between the particle's velocity and the corresponding velocity measurement of that detection.
3. w_ρ^i - Assigns weight based on the number of particles in a grid cell, assigning higher weights to dense clusters of particles.

The first weighting criterion assigns more value to particles positioned close to a detection. If more detections are received simultaneously, weights are assigned for each one separately and normalised afterwards. To assign the position weight, w_r^i, a normal distribution is computed for the range measurement, with the measured range, r_z, as mean and the specified sensor noise, σ_r, as its variance. The weight assigned is the probability of drawing the particle's position from that distribution.

$$w_r^i \sim \mathcal{N}\left(r^i; r_z, \sigma_r^2\right) \tag{2.20}$$

The second criterion, based on the similarity in velocity, enables the use of velocity measurement data to improve the estimate of the PF. The state vector of the particles consists of position and velocity states, as shown in Eq. (2.17). This measurement information can be used to correct the velocity estimate of particles, corresponding to the underlying dynamics of the approximated occupancy.

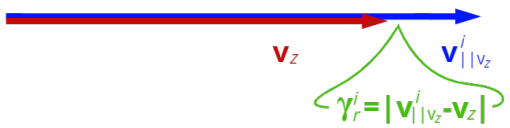


Figure 2.13: Visualisation of a measured velocity vector v_z, a particle's velocity vector $v^i_{||v_z}$ and the measurement residual, γ_r^i between them. These are used to calculate w_v^i according to [21].

$$ {}^{A}p_v^i = \max\left(0, \frac{L - \left(r^i - r_z\right)^2}{L}\right) \tag{2.21} $$

$$ \Lambda^i = \frac{1}{\sqrt{2\pi \cdot \sigma_{v_z}^2}} e^{\left(\frac{-\gamma_r^2}{2\sigma_{v_z}^2}\right)} \tag{2.22} $$

To ensure that only relevant velocity measurements are used to assign weight to a particle, an association probability, ${}^{A}p_v^i \in [0,1]$, is defined. On a per particle basis, this probability considers the location of received detections and evaluates if they are relevant to a particle. For this it only considers measurements close to that particle, i.e. in grid cells surrounding the particle's location. In this algorithm a distance of $1 \cdot L$ is used as the search domain. Within this domain, measurements that are further away receive a smaller probability according to Eq. (2.21).

Additionally, a likelihood, Λ^i, is used to quantify the resemblance between the measured velocity, v_z, and the particle's estimated velocity, v^i. A measurement residual, γ_r^i, is calculated between these velocities, which is simply the difference in magnitude in 1D, as indicated in Fig. 2.13. For a 2D setup the estimated velocity would first have to be projected in the direction of the measured velocity, hence $v^i_{||v_z}$, before the residual can be calculated. Then the likelihood is calculated according to Eq. (2.22), using the noise variance, σ_{v_z}, derived from sensor specifications. This gives the likelihood of the particle's velocity to be representative for the associated occupancy. The weight based on the velocity similarity, w_v^i, is then calculated from multiplication of the association probability and the likelihood.

$$ w_v^i = {}^{A}p_v^i \cdot \Lambda^i \tag{2.23} $$

The third weight, w_ρ^i, assigns a value according to the density of particle clusters. This is to give confidence in the estimated occupancy from previous detections. By assigning weight to clustered group of particles the estimation should preserve particles from previous estimates even if detections on the corresponding, represented dynamic environment are missed. The weight is calculated as the ratio between the amount of particles in a grid cell, N^c, to the total amount of particles in the population, $\sum_c N^c$. Therefore, every particle in the same grid cell receives an equal weight.

$$ w_\rho^i = \frac{N^c}{\sum_c N^c} \tag{2.24} $$

After computing the individual weighting criteria, they are combined to create a single weight per particle. To ensure equal contributions of each of the criteria, the summation of the weights per criterion is normalised to be equal to 1. Individual weights of each criterion are therefore divided by the summation of all weights in that criterion. This results in the normalised weights, $\bar{w}^i$. Per criterion all the normalised weights add up to 1, such that each criterion has an equal contribution to the total weight.

$$\bar{w}^i = \frac{w^i}{\sum_i w^i} \tag{2.25}$$

The normalised weights are combined a total weight per particle, $\bar{w}^i_{total}$. This total weight is normalised by its sum once more. This combination offers possibilities for adapting the contribution of each criterion in specific scenarios, for example by multiplying the different normalised weights in the numerator of the equation below. However, adapting the filter performance by these criterion is not discussed here, but will be detailed in Section 4.2.

$$\bar{w}^i_{total} = \frac{\bar{w}^i_r + \bar{w}^i_v + \bar{w}^i_\rho}{\sum_i \left(\bar{w}^i_r + \bar{w}^i_v + \bar{w}^i_\rho\right)} \tag{2.26}$$

Lastly, the new total weights can be combined with the total weights of the previous time step. This can create a memory effect to help particles survive that are representing detections, but for which no relevant measurements are received in the current time step. The effects of this addition will be evaluated in Section 4.1, as this intended effect is similar to that of the weight w^i_ρ. This combination is realised by using a decaying factor, $\alpha \in [0, 1]$, which reduces the contribution of the new weights and gives some confidence to the previous weights. For this step it is important to properly keep track of particles between steps and it is also the cause for the normalisation of the total weight in Eq. (2.26).

$$w^i_{total,k+1} = \alpha\, \bar{w}^i_{total,k+1} + (1 - \alpha)\ \bar{w}^i_{total,k} \tag{2.27}$$

2.3.4 Resampling

After each of the particles is updated and assigned a new weight, the resampling step is performed. A resampling algorithm is a random selection procedure to duplicate particles from the previous population to a new set, where the chance of being selected depends on the weight of each particle. Particles with higher weights are more likely to be selected and duplicated. The new, resampled population more accurately represents the posterior PDF of the occupancy. By resampling the problem of degeneracy of the algorithm is avoided. That is the situation that all but a handful of particles have weights close to zero.

Resampling in this case is done according to the Resample Wheel method [46]. The method can be visualised as in Fig. 2.14, where each particle is assigned a slice proportional to it's weight. The pseudo-code for the algorithm is shown below. Explained in short, a random particle index i and a value β, between 0 and twice the largest particle weight, are chosen. Index i is incremented until β is smaller than the weight of the current index, by subtracting the weight of the current index from β each increment. The particle at that index is duplicated to the new set, with its assigned weight. This selection procedure is repeated for the total amount of particles, N. An additional check is done so each cell contains at least one particle. If any cell remains empty after resampling, the algorithm is used to select a random particle, which is then replaced by a newly generated particle in the empty grid cell. Weights of these new particles are set equal to the smallest weight in the set. After resampling, all weights are normalised according to Eq. (2.25).

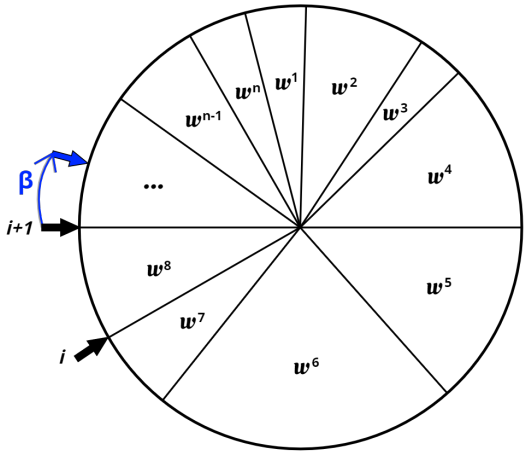


Figure 2.14: The resample wheel method visualised. In this wheel each particle is assigned a section proportional to its weight, w^i.

Algorithm 2.3 Resample Wheel algorithm in pseudo-code

$i = U[1, 2, \ldots, N]$, $\beta = 0$
for $particle = 1, 2, \ldots, N$ **do**
  $\beta = \beta + U[0, \ldots, 2 \cdot W_{max}]$
  while $W_i < \beta$ **do**
    $\beta = \beta - W_i$
    $i = i + 1$
  end while
  Select particle at index i
end for

2.3.5 Resulting occupancy

The result of the occupancy estimation by the PF can be visualised by plotting the particles by their position and weight. An example is given in Fig. 2.15, with the 1D+ setup from Fig. 2.6. In the top plot the objects and measurement beams are shown. The radar sensor model generates detections of the vehicles, which are displayed in the second plot as the measurements. The PF then approximates the occupancy of these objects by two separate particle clusters, as plotted in blue in the middle plot. Most particles have a negligible weight, however around the two detections distributions are formed. The difference in shape can be attributed to this specific scenario, in which the truck has been detected more often in the past. This 1D+ simulation shows that the filter approach is able to represent multiple objects simultaneously, which is an important conclusion before continuing on further analysis of this concept in the next chapters.

$$p_o(m^c) = \frac{N^c}{\max\limits_{1 \leq c \leq c_{max}} N^c} \tag{2.28}$$

The resulting, derived occupancy from the PF approximation is shown in the bottom plot of Fig. 2.15. Occupancy probabilities are derived from the amount of particles in a grid cell, N^c. That amount is counted and divided by the largest amount in one cell, according to Eq. (2.28). This results in a normalisation to occupancy probabilities for each cell. The cell with the highest amount of particles will be assigned a probability of $p_o(m^c) = 1$, while for the other cells $p_o(m^c) \in [0, 1]$.

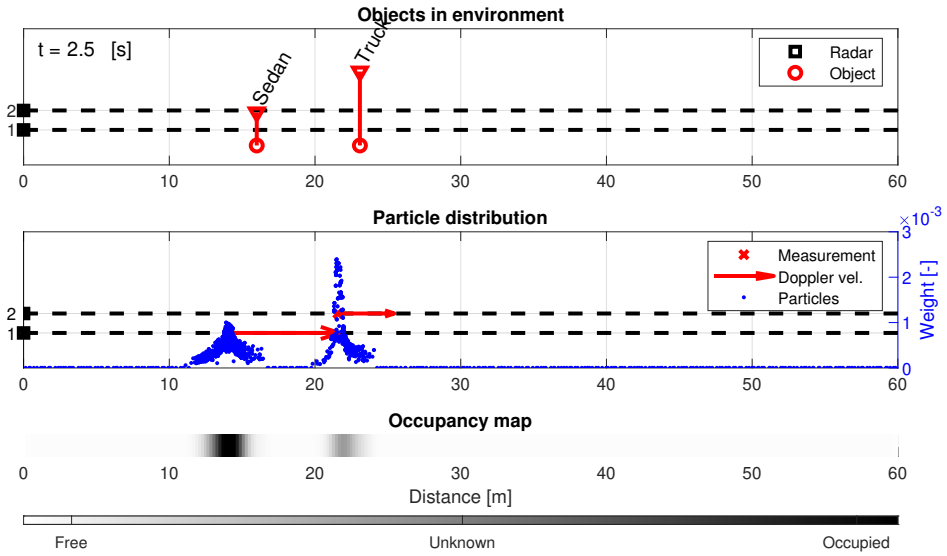


Figure 2.15: An example of occupancy estimation by a PF in the 1D+ setup. The top plot shows the actual objects, the middle the measurements with the particle population and the bottom shows the derived occupancy map.

2.4 Dynamic occupancy grid map

As outlined in Fig. 1.5, the outputs of the direct mapping method and PF are combined into the dynamic occupancy grid map. The direct mapping component uses measurements to update the map by a cumulative update equation. This method is susceptible to measurement uncertainties and dependent on the measurement frequency to update it with new information. In contrast, the PF can provide occupancy estimates at an independent frequency and can correct measurement uncertainties by assigning more confidence to predictions through the weighting criteria. However, the PF estimate does not indicate the unobservable area correctly. To that end the dynamic occupancy grid map combines both outputs.

Both outputs from the direct mapping and PF components are already in an occupancy grid format, so a simple logic rule is chosen to combine these maps. It is important to note that the direct mapping method treats unobservable area as unknown and assigns it as $p_o(m^c) = 0.50$. This is not done in the PF output as it treats unobservable area as free, unless previous detections are propagated there. Then occupancy is indicated in this area. For the combination the occupancy probability of a grid cell of the direct mapping output is used, unless the occupancy probability resulting from the PF exceeds $p_o(m^c) = 0.50$. In that case the value of the PF is used. This logic leads to the combination algorithm, as given here:

Algorithm 2.4 Dynamic occupancy grid map combination

if $m^c_{PF} \geq 0.50$ **then**
  $m^c_{DOGMa} = m^c_{PF}$ ▷ $_{DOGMa}$: indication of dynamic occ. grid map
else
  $m^c_{DOGMa} = m^c_{DM}$ ▷ $_{DM}$: indication of direct mapping
end if

To illustrate the use of the dynamic occupancy grid map, the scenario of Section 2.1.4 is revisited. Figure 2.16 shows a snapshot of the objects and a GT map. The occupancy grid map from the direct mapping only indicates occupancy at the closest object location and the cells behind it as unknown. The PF has estimates at both object locations, but with a larger spread. Combined in the dynamic occupancy grid map, the occupancy of both objects is indicated while also indicating unobservable area.

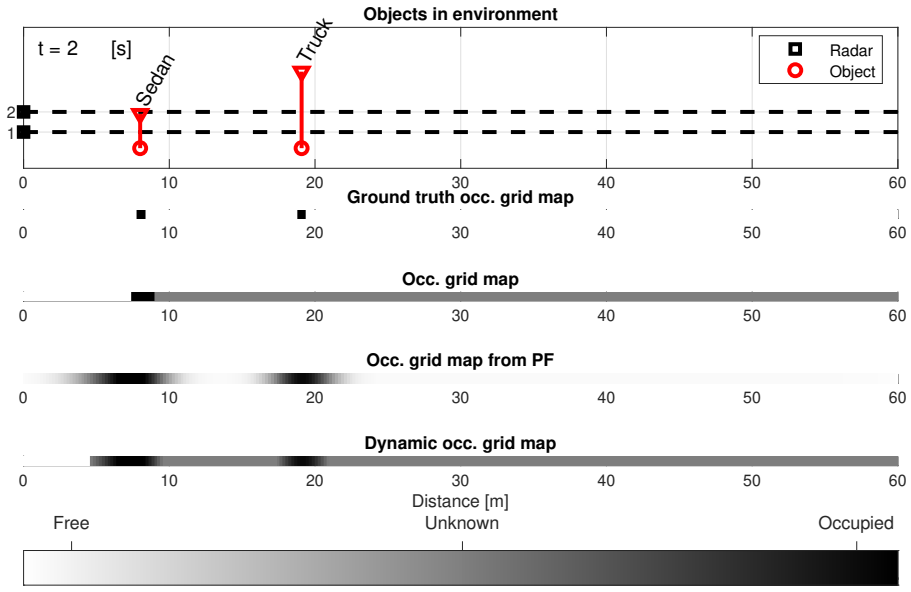


Figure 2.16: Comparison of occupancy maps generated from the 1D+ scenario of Fig. 2.6. From top to bottom the objects, the GT map and the occupancy grid maps from the direct mapping, PF and combination to the dynamic map are shown, respectively.

2.5 Summary

In this chapter the concept environment mapping algorithm is explained. It first provided an overview of the simulation environment in which the algorithm can be analysed. This detailed the scenario generation and how sensor models are applied to generate measurements. Then the components of the mapping algorithm are explained. In the direct mapping component measurements are translated to occupancy probabilities and cumulatively updated in an occupancy grid framework. Additionally, an approximation of the occupancy is provide by a PF. Therein a particle population is updated by a CVCO process model and corrected by weighting and resampling. In the last section the combination of both outputs to a dynamic occupancy grid map is explained.

3 Evaluation setup

To analyse the 1D algorithm more elaborately, several scenarios are defined in this chapter. These are all simulated in the extended 1D+ setup and each offers a specific challenge to the algorithm. Five scenarios are listed in Section 3.1, which also describes how these are setup. All scenarios are simulated once, to analyse the functionality of the algorithm and to detail the encountered shortcomings of the current implementation, both of which are described in Section 3.2. Then several evaluation metrics are defined in Section 3.3 so the performance of the algorithm can be evaluated by objective metrics. Results of this extensive evaluation are presented in Chapter 4.

3.1 Evaluation scenarios

The algorithm has demonstrated its ability to generate an occupancy map resembling the ground truth for a simulated scenario, by basic examples in Chapter 2. Additionally, the extended 1D+ setup proves that the PF is able to handle multiple targets, simultaneously. However, these basic scenarios do not allow for properly testing the algorithm. As the used scenarios only consist of one or two vehicles moving in the environment at constant velocity. Therefore, this chapter will several specific and more challenging scenarios. For example, it is important to analyse how the algorithm handles infrequent detections, but also acceleration in these detections. To that end five scenarios are defined as an elaboration on Section 2.1, which already introduced some scenario setups, the object properties and the 1D+ extension.

Below the five chosen scenarios are briefly introduced, before they are explained in more detail in Section 3.2. Additional snapshots of the scenarios are provided in Appendix A to better understand their progression in simulation. In the second column the motion profiles the objects adhere to are given by the equation to calculate their travelled distance, s, over the elapsed time, t. Parameters a and v refer to acceleration and velocity, respectively, while the scenario with sinusoidal velocities describes their motion according to a sinusoid with specified amplitude A, frequency f, and added phase Θ.

Scenario Description	Motion profile of objects *Equation*
Constant velocity scenario Simulated by a single object moving through the environment at a constant velocity.	Constant velocity $s = v \cdot t$
Scenario with loss of measurement Simulated by a single object moving through the environment at a constant velocity and stopping measurements after some time.	Constant velocity $s = v \cdot t$
Scenario with occlusion Simulated by an overtake manoeuvre or by having a small object disappear behind a larger one, all with constant velocity, causing the small object to be occluded from sensor view.	Constant velocity $s = v \cdot t$
Scenario with objects with changing velocity Simulated by a single object moving through the environment with a constant acceleration.	Constant acceleration $s = \frac{1}{2}a \cdot t^2 + v_0 \cdot t$
Combined scenario Simulated by initialising two objects in the environment with opposite sinusoidal velocity profiles, simulating occlusion and acceleration. One of the objects is given a larger height to simulate occlusion as well.	Sinusoidal velocity $s = \mathrm{A}\sin\left(2\pi \cdot f + \Theta\right) \cdot t$

Table 3.1: Overview of the five scenarios and the motion profiles their objects adhere to.

3.2 Functional analysis

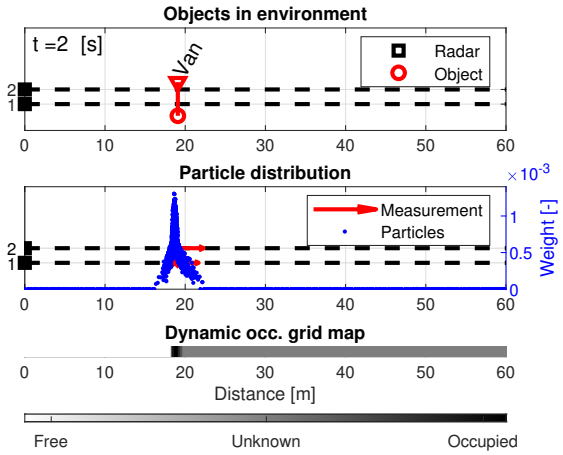
In this section each of the scenarios is described by snapshots of their simulated runs with the dynamic occupancy grid mapping algorithm. These first simulations will be analysed with a focus on the observed performance, but also the limitations and shortcomings of the algorithm, i.e. a functional analysis.

3.2.1 Scenarios

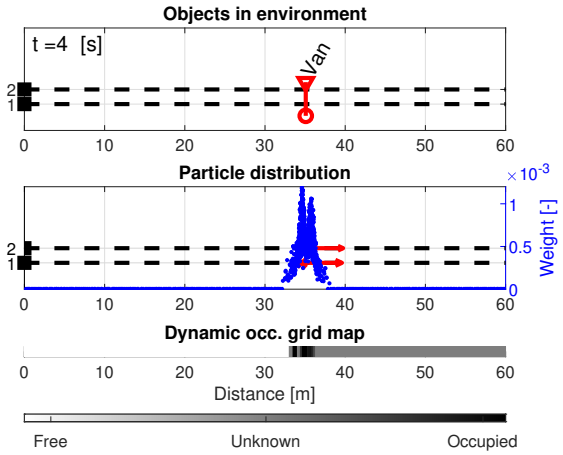
Constant velocity

The first scenario is used to establish a baseline as it is the most straight-forward scenario. A single object is simulated moving through the environment at a constant velocity. This scenario should not pose any challenges to the mapping algorithm, but it is a helpful scenario to compare other results to. The only expected difficulty is due to missed measurements as the object moves further away from the sensors, however the PF should take care of these instances by propagating the occupancy estimates.

To simulate this scenario an object is initialised close to the ego vehicle. In this case a van is chosen, which provides a substantial RCS, so it is unlikely to be missed by sensors compared to smaller vehicles. The van travels through the range at a constant velocity, such that in about 7 $[s]$ it has travelled through the 60 $[m]$ range. These values are chosen, as the sensor starts missing detections more frequently with objects further in the range. In Fig. 3.1 two snapshots of the simulation are shown at 2 and 4 $[s]$, respectively. At the first snapshot, the object is consistently detected as seen by the correct estimate of its occupancy. Both the bias and variance of the estimate are low. Around the time of the second snapshot, the sensors start to miss detections and noise on these measurements has grown. Therefore, the bias and variance of the occupancy estimate have grown larger, also visible by the two peaks in the particle distribution. But, visual inspection shows that the dynamic occupancy grid map still indicates occupied space at the correct position.



(a) Snapshot of the simulation at the start of the run. Bias and variance of the estimate by the PF are low due to availability of recent detections.



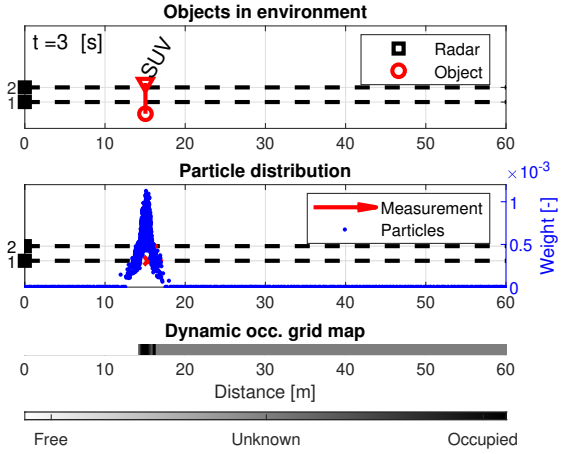
(b) Snapshot of the simulation later during the run. Bias and variance of the estimate have grown slightly due to missing some detections.

Figure 3.1: Simulation of the constant velocity scenario. Top plot shows the objects, the middle plot the detections and PF estimate, while the bottom plot shows the dynamic occupancy map.

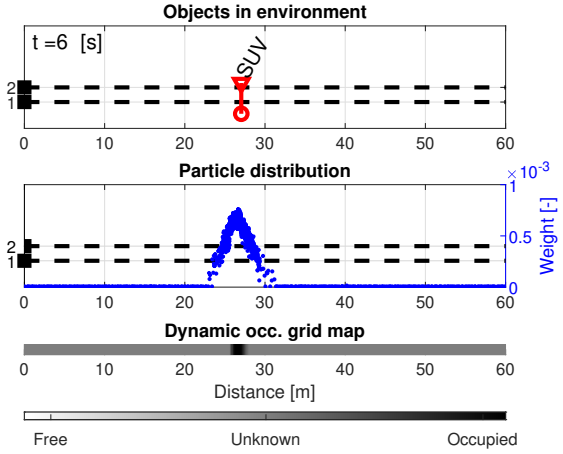
Loss of measurement

The second scenario simulates loss of measurements on an object that was detected earlier. This scenario enables to evaluate continuation of the PF estimate. As the object has been detected earlier on, the PF has an estimate of its occupancy. When the measurements are turned off, the estimate should survive due to propagation of the particles approximating its distribution. This scenario will test that hypothesis.

To simulate this scenario, an object is initialised travelling through the range with a constant velocity. To illustrate this, two snapshots of the simulation are provided in Figs. 3.2. After a certain time, 3 $[s]$ in this example, measurements are turned off and the object is not detected anymore. Therefore, only the propagated estimate by the PF remains. This estimate can be seen in Figs. 3.2a and 3.2b by the distribution in the particle population and the occupancy estimate in the middle and bottom plots, respectively. The propagated estimate in Fig. 3.2b remains close to the actual position of the object, so it has a low bias. The variance has grown, as can be seen by the wider distribution of the particles. Thus the algorithm is able to keep tracking the occupancy of an object after losing detections, but the estimate becomes less confident over time. As the direct mapping component receives no more measurements, the dynamic occupancy grid map indicates an unknown occupancy state for the whole range in the second snapshot, except for the estimated location of the SUV. Here the added value of the continued estimate by the PF shows.



(a) Snapshot of the simulation just before loss of measurement. Bias and variance of the estimate by the PF are low due to availability of recent measurements.



(b) Snapshot of the simulation after loss of measurement. The estimate by the PF shows a larger variance and some bias due to loss of measurements.

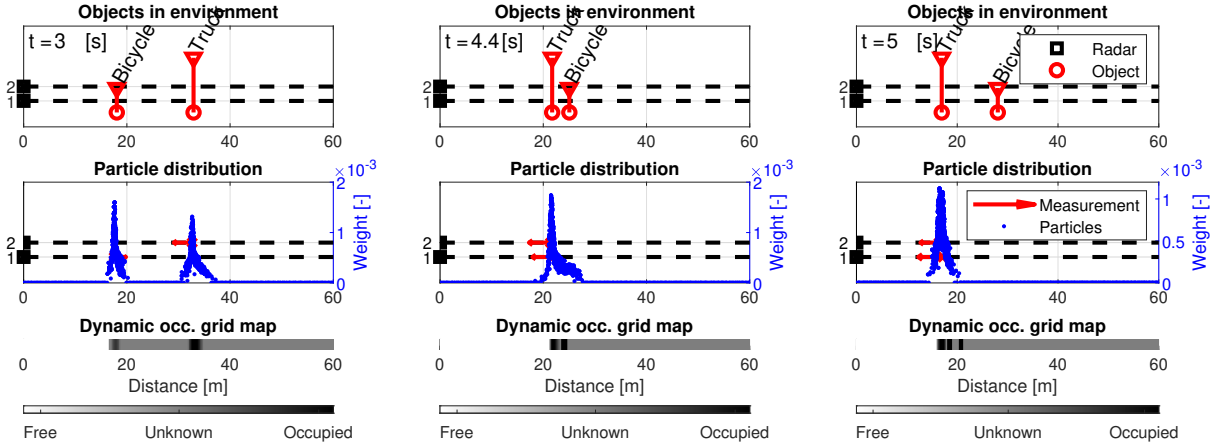
Figure 3.2: Simulation of loss of measurement scenario. Top plot shows the objects, the middle plot the detections and PF estimate, while the bottom plot shows the dynamic occupancy map.

Occlusion

The third scenario simulates loss of measurements on an object due to occlusion by another object. Similar to the loss of measurement case, one object has been detected earlier on and the continuation of its occupancy estimate is evaluated. However, now another objects is in the sensor's FOV and generates new detections. Therefore, detections of this second object can interfere with its occupancy estimate. So in this scenario the continuation of an occupancy estimate without new measurements, but with detections of other objects, is evaluated.

To simulate this, two objects, one significantly larger than the other, are initialised travelling in opposite directions. At the start of the simulation both objects are detected. The larger object, the truck as in the example of Figs. 3.3, is furthest away. It is not occluded, as it is significantly taller than the bicycle. When the objects pass each other, as in Fig. 3.3b, the smaller object is not detected anymore, due to occlusion by the larger one.

Looking at the occupancy estimates, Fig. 3.3a shows both objects clearly as both are currently detected. After passing each other, as in Fig. 3.3b, the estimate of the bicycle is still there but starts to fade, which shows by a large variance and low weights of particles at its position. This can be attributed to it not being detected anymore and the algorithm only receiving measurements of the larger object. Therefore the weights of the particles representing the occupancy of the truck to be significantly higher, giving those a higher chance of survival during resampling. Some time later, in the snapshot of Fig. 3.3c, the estimate of the bicycle's occupancy has disappeared completely.



(a) Snapshot of the simulation before occlusion. Occupancy estimates for both objects are present, with low bias and variance as both are detected.

(b) Snapshot of the simulation just after occlusion. The occupancy estimate of the bicycle is fading, as it is occluded by the truck and therefore is not detected.

(c) Snapshot some time after occlusion. There is no more estimate of the bicycle's occupancy, as it has not been detected anymore for some time.

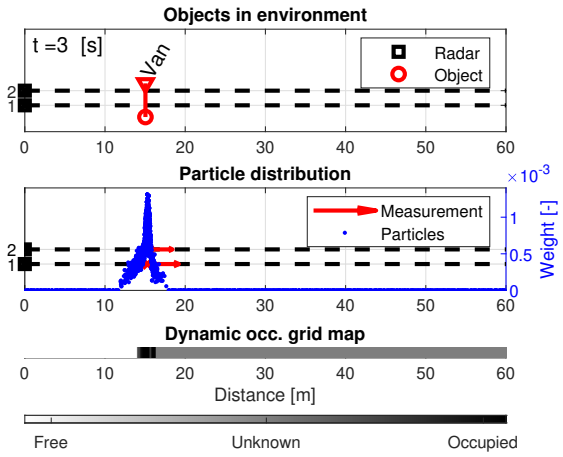
Figure 3.3: Simulation of the scenario with occlusion. Top plot shows the objects, the middle plot the detections and PF estimate, while the bottom plot shows the dynamic occupancy map.

This example shows that the algorithm is able to approximate the occupancy of multiple objects. However, when one of those is not detected anymore its estimate will disappear significantly faster than in the loss of measurement scenario, due to the availability of other measurements. The weighting criteria of the PF, as explained in Section 2.3.3, can be altered to change this behaviour. This will be investigated in Section 4.2 by changing the individual contributions of the weighting criteria to the total weight. Especially the weight based on the amount of particles in a grid cell, w_ρ, could positively influence this behaviour.

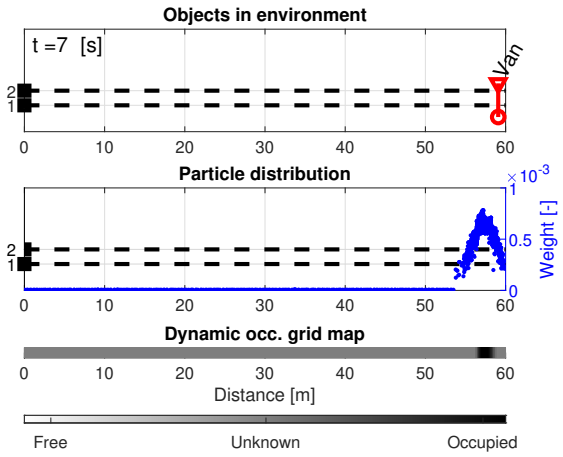
Changing velocity

In the fourth scenario an object with changing velocity is simulated. The process model in the PF operates on a constant velocity, i.e. CVCO, assumption to update the particles. As no acceleration is assumed in the propagation dynamics, the filter relies on the added noise, ν, to capture acceleration. In the weighting and resampling step, particles that more accurately resemble that are favoured by the selection procedure. Otherwise, the algorithm has to rely on newly generated particles with randomised velocities to capture the correct acceleration. This is not ideal however, and therefore it has to be evaluated how well the current process model performs for estimating occupancy from objects with accelerations.

To simulate this scenario a single object is initialised, travelling through the range with a constant acceleration. The object is easily detected when it is close to the radar, however as it travels further in the range the measurements become scarce. Therefore, the object is initialised close to the radar, with a low initial velocity. Figures 3.4a and 3.4b show the object at the start of the simulation and when it has almost left the range. In the latter case, several detections have been missed. Figure 3.4b shows that the both bias and variance of the estimate have grown, but it still approximates the correct object. The bias is explained by the misrepresentation of the object's velocity. Most likely due to missing several detections and not accounting for acceleration in the CVCO process model, which causes the velocity estimate to be off. Over time this causes an offset in the location of the approximated occupancy. This hypothesis of the misrepresented velocity is investigated in more detail in the quality analysis of Section 4.1. In conclusion, from visual inspection the algorithm behaves as expected in this scenario, where it shows the limitations of the chosen process model.



(a) Snapshot halfway through the simulation. The radar is able to detect the object frequently as it is close, so the estimation is accurate.



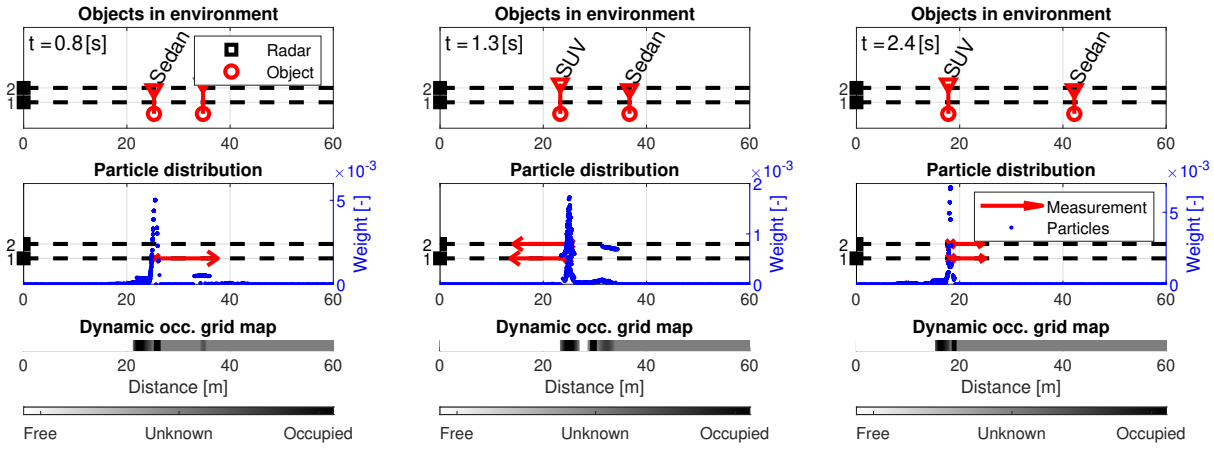
(b) Snapshot at the end of the simulation, when a some detections are missed. The estimate shows a larger variance and a significant bias of ≈ 2 $[m]$.

Figure 3.4: Simulation of the scenario with changing velocity by a constant acceleration. Top plot shows the objects, the middle plot the detections and PF estimate, while the bottom plot shows the dynamic occupancy map.

Combination of scenarios

Lastly, a challenging scenario is created, combining the specific challenges of the three other scenarios. It includes occurrences of loss of measurement, occlusion and changing velocity. To simulate this, two objects are initialised in the environment. One of the objects is smaller than the other and is not detected by the top radar beam. Both objects move according to a sinusoidal velocity profile with opposite periods, but with similar means and amplitudes. This creates a changing velocity profile for both, which is challenging as the accelerations are not constant and the estimation has to deal with velocities changing direction as well. Occlusion occurs whenever the tall object is obstructing the smaller from view and loss of measurements occurs when the larger object is far away, causing missed detections.

Figures 3.5 shows this scenario by three snapshots. The first frame in Fig. 3.5a shows both objects before they pass each other. The smaller sedan is detected consistently and shows a confident estimate with low bias. Further in the range the SUV is not detected as consistently and the estimate is less confident. It has a large bias, however in the middle plot new particles with high weights can be seen around the most recent measurement. The estimate of the sedan shows a trail, which can be attributed to the object's acceleration. The second plot in Fig. 3.5b shows the objects after passing each other. Estimates of both persist, with some bias. The bias and variance for the sedan are larger as detections on this object are lost. Lastly, in Fig. 3.5c a snapshot is taken just after both objects changed direction. The estimate for the SUV shows some bias, due to its change in direction. However, new particles with high weight are already situated around the recent measurements. So in a few time steps the estimate should be correct again. The estimate on the sedan is lost completely since it is occluded and other detections are available. This scenario proves especially challenging for the current implementation of the algorithm.



(a) Snapshot before objects cross each other. The estimate of the sedan is accurate, but shows a tail due to its acceleration. Due to scarce measurements the occupancy estimate of SUV is not accurate.

(b) Snapshot after objects cross each other. Occupancy is estimated for both objects, but the estimate of the sedan is biased due to lack of measurements and acceleration of the object.

(c) Snapshot after objects change velocity direction. There is no occupancy estimate of the sedan as it is occluded, while the estimate of the SUV is biased due to a change in direction of the object.

Figure 3.5: Simulation of the combined scenario, showing two objects with opposite sinusoidal velocity profiles. Top plot shows the objects, the middle plot the detections and PF estimate, while the bottom plot shows the dynamic occupancy map.

3.2.2 Results

The previous section shortly analysed each scenario and listed the shortcomings of the algorithm in each. Most of these are due to the chosen estimation approach, however they are important to be aware of. In this section these are listed. Below three limitations of the current algorithm implementation are identified:

1. The filter's process model, assuming CVCO of the particles, is not complex enough to estimate change in velocities of objects without consistent, subsequent measurements.
2. Estimates of recently occluded objects do not survive long when other measurements are available to update the particle population.
3. In case of occlusion, when objects pass each other closely, estimates can merge. This occurs when one of the objects does not receive measurements anymore and velocities are not significantly different.

The first limitation is due to the chosen process model. Here improvements are certainly possible and can be researched. However, the second and third limitation are inherent to the PF's non-parametric approach. As the filter estimates occupancy by updating an un-associated population of particles, estimates on occluded objects tend to disappear quickly. However, the longevity of these estimates could be improved by adapting and adding to the current weighting criteria, by focusing on preserving previously assigned weights. Some proposals are done to achieve this, by the introduction of the weighting criterion based on grouping of particles, w_ρ of Eq. (2.24), and the decay factor, α of Eq. (2.27), in computation of the total weight per particle.

3.3 Evaluation metrics

Now the limitations of the algorithm are identified a more extensive evaluation can be performed, which will be described in Chapter 4. In that chapter the contributions and strong points of components of the algorithm will be analysed in Section 4.1 and a parameter sweep is performed in Section 4.2 to analyse the contribution of several parameters to its performance. For this it is important to have objective, evaluation metrics in place. Therefore, several performance metrics are explained in this section. First, a set of metrics comparing generated maps to GT occupancy maps is given in Section 3.3.1. Then an additional, independent metric based on multi-target tracking applications is explained in Section 3.3.2.

3.3.1 Ground Truth map metrics

The first set of metrics compare the generated occupancy maps against a GT map. Since object movement is determined beforehand, their occupancy can be derived to generate a GT map. In this map each cell is assigned as either occupied or free, resulting in a binary map. Then several indicators can be computed from the comparison between this GT map and the one created by the mapping algorithm. These indicators, selected from work by V.R. Madhavan [47], will be explained in detail in this section.

First, the Map Score (MS) and Map Error (ME) metrics are introduced. These compare occupancy probabilities on a per cell basis and give an indication of similarity and discrepancy between the generated map and the GT map. The MS metric is based on the probability that two maps represent the same environment and scores the generated map between 0 and 1. A higher score indicates a closer similarity, with a score of 1 indicating both maps are equal. As the occupancy probabilities per cell can be small values, mathematical operations on these can lead to truncation errors in multiplication of these values. Therefore, a logarithmic approach is chosen to compute the score [48]. Additionally, the MS is normalised over the amount of cells, such that maps of different sizes can be compared.

$$MS = \sum_{c} \log_2 \left[1 + p_o\left(m^c\right) p_o\left(m^c_{GT}\right) + p_f\left(m^c\right) p_f\left(m^c_{GT}\right)\right] \tag{3.1}$$

$$\overline{MS} = \frac{MS}{c_{max}} \tag{3.2}$$

In Eq. (3.1) $p_o(\ldots)$ represents the probability for a cell, m^c, to be occupied and $p_f(\ldots)$ is its compliment, the probability for a cell to be free as according to Eq. (2.2). The subscript m_{GT} indicates a cell belongs to the GT map. The normalisation over the total amount of cells in the map, c_{max}, is displayed in Eq. (3.2). Instead, the ME metric evaluates the differences between the generated and the GT map and will score opposite to the MS metric. It represents the sum of all differences on a cell by cell basis [47]. Similar to the MS metric, it is normalised over the amount of cells and indicates a value between 0 and 1, for binarised GT maps. For this metric a lower score is better, with a score of 0 indicating that both maps are equal.

$$ME = \sum_{c} \left| p_o\left(m^c\right) - p_o\left(m^c_{GT}\right) \right| \tag{3.3}$$

$$\overline{ME} = \frac{ME}{c_{max}} \tag{3.4}$$

To better interpret the difference between the generated map and the GT map, a binary classification is used to compare them. This classification uses a confusion matrix to charactarise the differences, following the matrix proposed by V.R. Madhavan [47]. In this matrix individual cells are classified as either correctly identified, as occupied or free, or as misidentified. Occupancy probabilities in the generated maps are therefore approximated to either occupied or free, i.e. probabilities of $p_o(m^c) > 0.50$ are indicated as occupied. Within the confusion matrix framework, free cells are referred to as negative and occupied ones as positive. A cell correctly identified as free is then referred to as True Negative (TN), while for occupied this is a True Positive (TP). Similarly, a misidentified cell is referred to as a False Negative (FN), for incorrect free cells in the generated map, or a False Positive (FP), for occupied ones. These distinctions aid in identifying over- or underestimation of either occupied or free space.

Confusion Matrix		**Generated Map**	
		Occupied	Free
GT Map	Occupied	True Positive (TP)	False Negative (FN)
	Free	False Positive (FP)	True Negative (TN)

Table 3.2: Confusion matrix for comparing binarised occupancy maps

From the identified states of the confusion matrix in Tab. 3.2 several rates can be computed. These help to give insight in what happens during simulation. One of these is the False Positive Rate (FPR), which is the ratio of cells that are falsely identified as occupied, FP, out of all free cells in the GT map. This gives and indication of the overestimation of occupancy in the generated map, which can limit the perceived, available free space. It gives a value between 0 and 1 and thus can be compared against maps of varying

sizes and across different scenarios. The compliment of this metric is the True Negative Rate (TNR), which indicates the cells correctly identified as free, TN, out of all free cells in the GT map.

$$FPR = \frac{C_{FP}}{C_{FP} + C_{TN}} \tag{3.5}$$

On the contrary, the False Negative Rate (FNR) indicates the amount of cells falsely identified as free, FN, out of all occupied cells. This ratio is again between 0 and 1, but it can show large variance if the amount of occupied cells in the GT map is low. As this metric indicates missing objects in the environment, it should be as low as possible. A high FNR can result in severe consequences, e.g. a collision due to incorrect path planning as a consequence of incorrectly perceived free space. The compliment of this metric is the True Positive Rate (TPR), which indicates the cells correctly identified as occupied, TP, out of all occupied cells in the GT map.

$$FNR = \frac{C_{FN}}{C_{TP} + C_{FN}} \tag{3.6}$$

3.3.2 GOSPA metric

An additional metric, the Generalized Optimal Sub-Pattern Assignment (GOSPA) metric, is used to evaluate the algorithm performance. It provides an alternative metric, which has been developed for multi-target tracking applications. It assesses similarity between a set of ground truths and estimated tracks. This metric is based off the OSPA metric, which compares the sets on the optimal, least distance between them. It penalises the localisation errors and normalises this by the amount of truths. By removing this normalisation from the OSPA metric, over- and underestimation of the amount of targets is penalised so both localisation and cardinality errors can be analysed. Additionally, it provides a more common metric that allows comparing this algorithm to alternative environment mapping approaches. Following the work of Rahmathullah et. al. [49], the computation of the GOSPA metric is outlined here.

At any time-step, t, a list of m truths is defined, by position and velocity states of the actual objects.

$$Q = [q_1, q_2, \dots q_m] \tag{3.7}$$

At the same time a list of n tracks is derived from the PF estimate, which is detailed later in this section.

$$Y = [y_1, y_2, \dots y_n] \tag{3.8}$$

Assuming $m \leq n$, the GOSPA metric can be computed according to:

$$\text{GOSPA} = \left[\sum_{i=0}^{m} d_c \left(q_i, y_{\pi(i)} \right)^p + \frac{c^p}{\alpha} (n - m) \right]^{1/p} \tag{3.9}$$

Here d_c is the cutoff-based distance, which is the minimum between a defined cutoff distance, c, and the base distance $d_b(x, y)$. The latter is calculated by an Euclidean metric between a truth, q_i, and its assigned track, $y_{\pi(i)}$, which is based on the optimal, least distance between the two sets, by the average Euclidean distance over all states. The cutoff value c then defines the maximum, allowable localisation error.

$$d_c(q, y) = \min \left(d_b(q, y), c \right) \tag{3.10}$$

The exponent p in Eq. (3.9) determines how severely the localisation and cardinality errors are penalised. The larger this value, the higher the GOSPA metric can become. Lastly, parameter α is a normalisation parameter in the cardinality error component, which is set to $\alpha = 2$ to accomplish the GOSPA as proposed by Rahmathullah et. al. [49], who argue that: "*The basic idea behind selecting* $\alpha = 2$ *is that the cost for a single unassigned (missed or false) target should be the same, whether or not it is associated to another target in the permutation in Eq. (3.9). Therefore, given that a pair of unassigned targets costs* c^p *and an unassigned target costs* $\frac{c^p}{\alpha}$*, we argue that* $\alpha = 2$ *is the most appropriate choice.*".

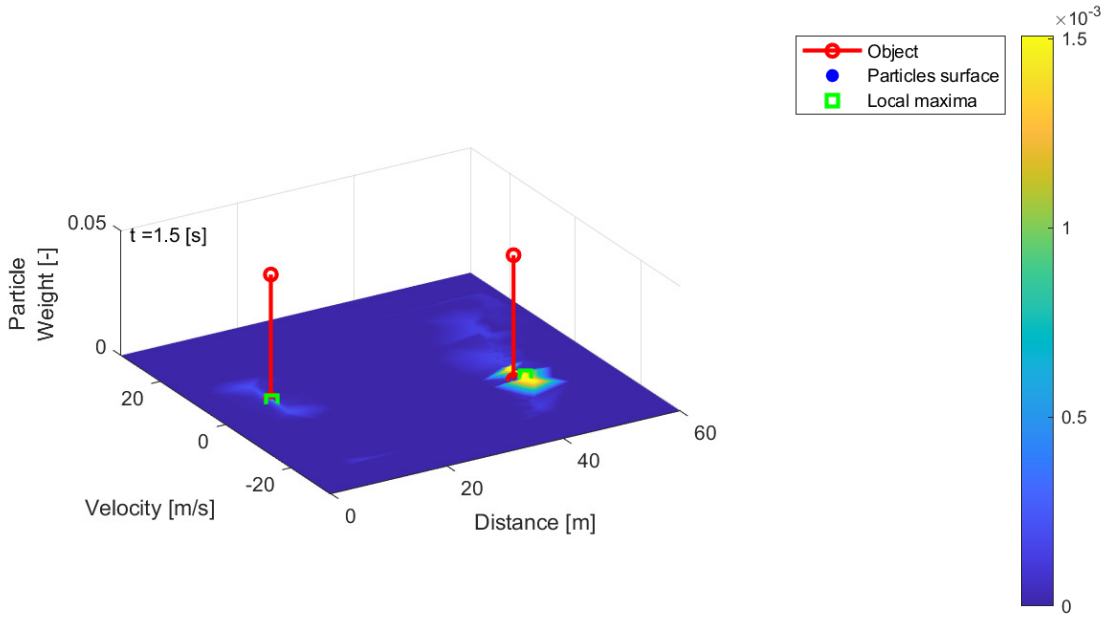


Figure 3.6: Interpolated surface through the particle population, by position, velocity and weight of each particle. The plot indicates the derived local maxima and actual objects.

The set of truths can simply be derived from locations and velocities of objects in the scenarios. In contrast, the set of tracks are derived from the PF estimate, as particles carry both position and velocity information. First a surface is interpolated through the particle population, which is treated as 3D scattered data points by position, velocity and weight. Then the local maxima are determined by 2D order-statistic filtering, which looks for the highest weight in a specified domain, along the 2D plane formed by position and velocity data. Figure 3.6 shows an example of the interpolated surface and derived local maxima.

As the PF represents the dynamic environment by estimating occupancy, instead of detecting and tracking targets, it is not directly compatible with the GOSPA metric. The resulting scores will therefore be highly dependent on correctly identifying the local maxima. However, computing this metric gives an additional metric, that also indirectly incorporates the velocity estimates in its scoring. It also offers a metric which can be compared to other environment mapping approaches.

3.4 Summary

The presented environment mapping has been worked out in Chapter 2 in the 1D+ setup. In this chapter five simulation scenarios were introduced, which are used to analyse and evaluate the concept algorithm. These scenarios each offer a specific challenge to the mapping algorithm and are used to evaluate the functionality, quality and performance of the concept. Due to the predefined motion of the objects in the scenarios, ground truths are easily obtained. Then a first, functional analysis is done to see if the algorithm functions as intended, but also to list its limitations. This simultaneously shows the progression of the scenarios. Lastly, this chapter introduces evaluation metrics, based on comparing output maps to ground truths and multi-target tracking evaluation metrics. These will be used in the next chapter to objectively evaluate the accuracy and performance of the algorithm.

4 Performance evaluation results

In the previous chapters both the environment mapping algorithm and the evaluation setup are explained. To evaluate the accuracy and performance of the algorithm, objective evaluation metrics were introduced in Section 3.3. These are used to score and review the resulting environment map in this chapter. First the direct mapping and PF components are analysed in Section 4.1, to determine the contribution made by each component. Then a baseline indication of the performance and accuracy of the complete 1D+ algorithm is given, for each of the scenarios described in Section 3.1. In a parameter study several parameters are isolated and varied to study their effect on the performance. This chapter will conclude by giving recommendations on how to change parameters in the algorithm to improve on the established baseline.

4.1 Quality analysis

In the quality analysis the performance of the algorithm is evaluated step by step. Both outputs of the direct mapping and PF components are separately evaluated to analyse their contributions in the dynamic occupancy grid map. First, direct mapping component is compared against the PF estimate. Then the PF extensions are evaluated, i.e. the addition of weighting criteria and the decay of the particle weights.

4.1.1 Simulation setup

To reduce the deviation in the resulting maps, the measurements are generated once for each of the scenarios, fixed and saved as a data set. Then the algorithm is simulated twenty times on each of these sets. This amount of runs is determined by comparing the mean scores of two sets of runs on the same scenario, but with a variation in the algorithm. A student t test was applied, according to the theory by C.H. Goulden [50], to determine if there is statistical evidence that the means are significantly different. This was the case for twenty samples or more. The previously described metrics of Section 3.3 are used to score the algorithm and are computed as the average over the runs, computed first as the average at each time-step and then averaged over the duration of the simulated scenarios, i.e. 7 [s].

For the quality analysis only three scenarios are used. First is the constant velocity scenario, which is used as the basic scenario for referencing the other scores to and evaluating the additions to the PF. Secondly, the scenario with loss of measurements is used to examine the PF performance and is used to evaluate the benefit of adding the weighting criterion, w_ρ, based on the number of particles per grid cell. Lastly, the scenario with objects with changing velocity is used to evaluate the velocity estimation in the PF. Six variations of the algorithm will be simulated, each one isolating performance by the additions to the PF:

1. The direct mapping component.
2. Occupancy estimate from the PF with only weighting criterion w_r.
3. Occupancy estimate from the PF with weighting criteria w_r and w_v.
4. Occupancy estimate from the PF with weighting criteria w_r and w_ρ.
5. Occupancy estimate from the PF with all three weighting criteria.
6. Occupancy estimate from the PF with all three weighting criteria and the decay factor α as in Eq. (2.27).

4.1.2 Results

First, the occupancy grid map from the direct mapping component is compared to the occupancy estimate by the PF to determine the performance by the individual components. The PF variation with all weighting criteria and the decay factor α is used here. Then the evaluation metrics are computed for each of the scenarios and the average scores over the twenty runs are listed in Tab. 4.1, together with standard deviations. As the GOSPA metric is only computed for the PF estimate, it is not shown in the table for the direct method. Therefore, this metric will be left out of this first comparison.

Scenario	GOSPA	MS	ME	FPR	FNR
Constant velocity - PF	$1.795_{\pm 1.049}$	$0.966_{\pm 0.004}$	$0.039_{\pm 0.004}$	$0.028_{\pm 0.004}$	$0.271_{\pm 0.120}$
Constant velocity - Direct	–	$0.659_{\pm 0.000}$	$0.409_{\pm 0.000}$	$0.010_{\pm 0.000}$	$0.540_{\pm 0.000}$
Loss of measurement - PF	$1.328_{\pm 1.524}$	$0.963_{\pm 0.005}$	$0.042_{\pm 0.005}$	$0.031_{\pm 0.004}$	$0.142_{\pm 0.143}$
Loss of measurement - Direct	–	$0.618_{\pm 0.000}$	$0.459_{\pm 0.000}$	$0.009_{\pm 0.000}$	$0.535_{\pm 0.000}$
Changing velocity - PF	$1.530_{\pm 1.104}$	$0.969_{\pm 0.003}$	$0.036_{\pm 0.003}$	$0.025_{\pm 0.003}$	$0.158_{\pm 0.022}$
Changing velocity - Direct	–	$0.664_{\pm 0.000}$	$0.402_{\pm 0.000}$	$0.013_{\pm 0.000}$	$0.347_{\pm 0.000}$

Table 4.1: Evaluation results comparing the PF estimate to the direct mapping output, with baseline settings. Mean values with the standard deviation are given for scores of each metric.

The results shows that the PF estimate is performing better in all scenarios, as indicated by the scores highlighted in green. Only two metrics are indicated as both the MS and ME give similar information, while the FNR is deemed more critical to safety than the FPR. The better score of the MS can be attributed to how unobservable area is handled. The PF shows this as free and the direct mapping method as unknown, while it is occupied or free in the binary GT map. Results from the FNR indicate better performance for the PF estimate as well. It has a much higher standard deviation than the other GT metrics. This is attributed to the way the metric is computed. As explained in Section 3.3, the FNR is computed as a ratio between the amount of cells being falsely identified as free to the actual amount of occupied cells. As the latter amount is often relatively low, the standard deviation is high.

An example of the average scores during a simulation of the changing velocity scenario is shown in Fig. 4.1. Here the scores of the direct mapping are shown in green and the scores by the PF in blue. One thing that can be observed immediately is the spike around 6 $[s]$ of the GOSPA and FNR scores. This is when detections are missed more often, as the object is further away from the sensor. Due to this and the misrepresentation of the acceleration of the object in the PF process model, the scores worsen. From the FNR, in the bottom-right plot, the missed detections are apparent by the spikes in the score of the direct mapping component. However, these spikes are not present in the PF score as its estimates continue to approximate the occupancy of the object correctly. The same explanation applies to the peak in the GOSPA metric. For the MS its dip and for the FPR its peak are less significant as these metrics are normalised over all cells and all occupied cells, respectively. Lastly, the standard deviation indicated for the GOSPA score is quite large. This can be attributed to how the metric is computed, as it relies on correctly identifying local maxima from the particle population.

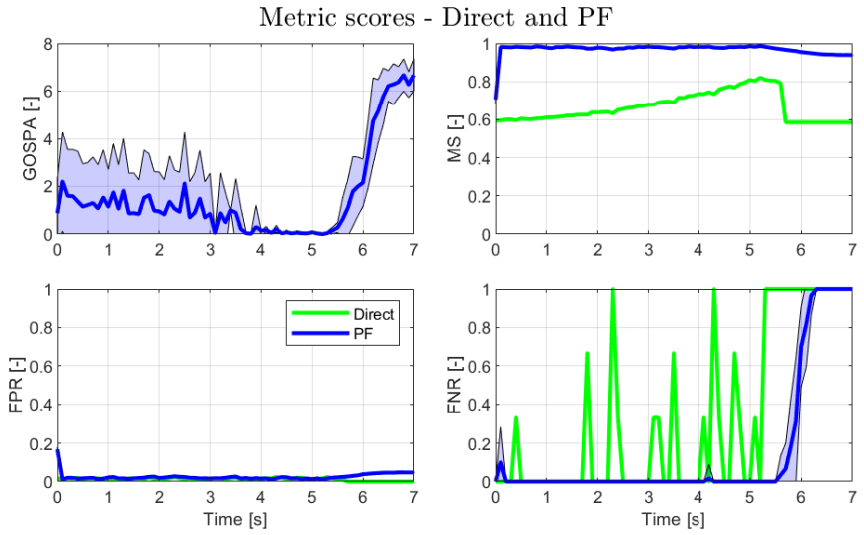
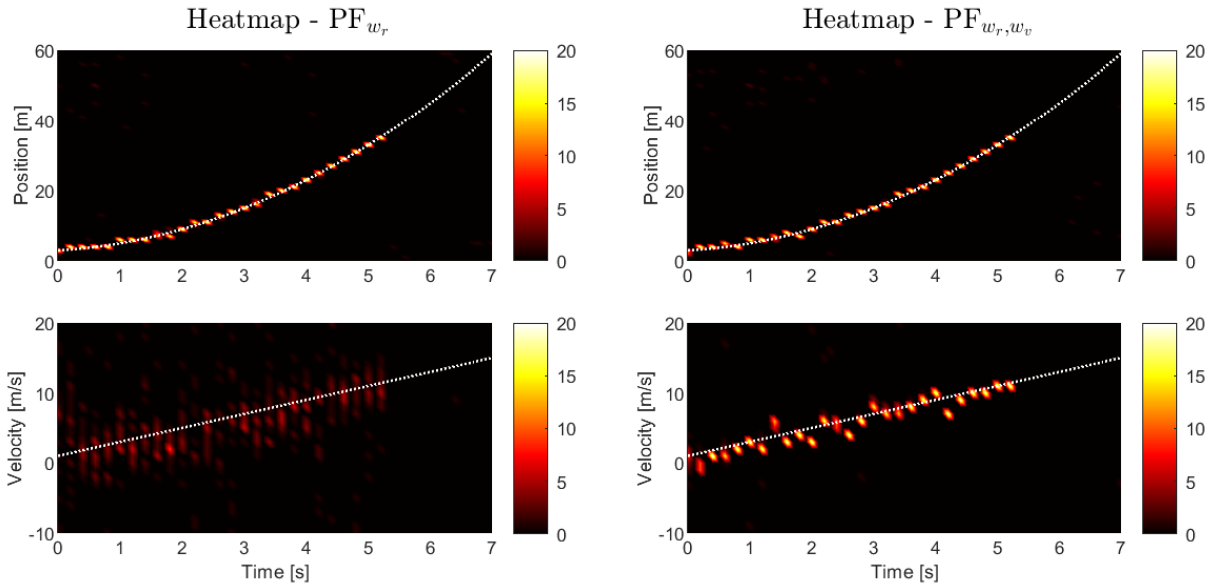


Figure 4.1: Average scores computed for the changing velocity scenario, comparing the occupancy grid map by the direct mapping method (green) to the estimate by the PF (blue). Each metric shows the mean value as a solid line and the standard deviations by shaded boundaries.

This first comparison shows that the occupancy estimate by the PF is better able to represent the actual occupancy than the direct mapping component. Therefore, the quality analysis continues by comparing the additions to the PF.



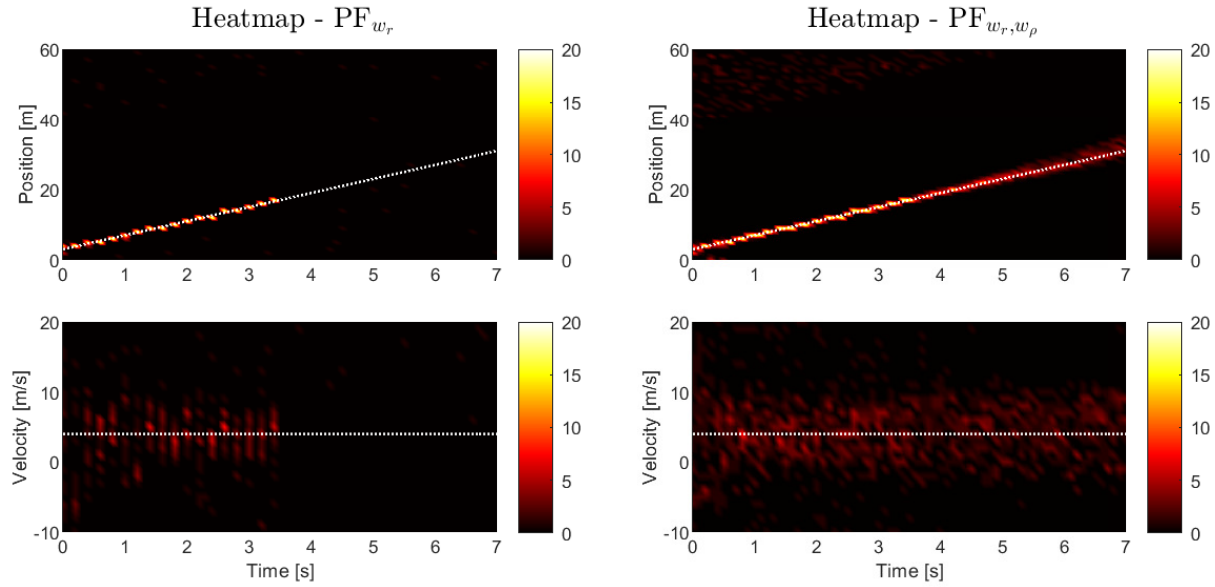
(a) PF with weighting criterion w_r

(b) PF with weighting criteria w_r and w_v

Figure 4.2: Heatmaps of the PF estimates, averaged over 20 runs, for the scenario with objects with changing velocity. The top plot shows the position estimates and the bottom plot the velocity estimates. In both plots the GT is indicated in white.

First, the PF with just w_r, is compared to the one with both w_r and w_v in the changing velocity scenario. Figure 4.2 shows the results, with the two variations in Figs. 4.2a and 4.2b, respectively. Comparing both velocity plots, the velocity estimate improves in accuracy with the addition of w_v, as shown by the decreased variance in the bottom plot of 4.2b. This improved accuracy will increase the position accuracy of the occupancy estimate.

Secondly, the criterion based on the amount of particles per grid cell, w_ρ, is evaluated. This criterion aims to retain estimates of targets by assigning more weight to particles that are closely grouped together. This is evaluated in the loss of measurement scenario, with the results shown in Figure 4.3. Comparing Figs. 4.3a and 4.3b, both the position and velocity estimates are kept for a longer time by the addition of this criterion. At 3.5 $[s]$ the measurements are turned off and the occupancy estimates by the PF with only w_r disappear. In contrast with the addition of w_ρ the estimate is kept for the duration of the simulation.



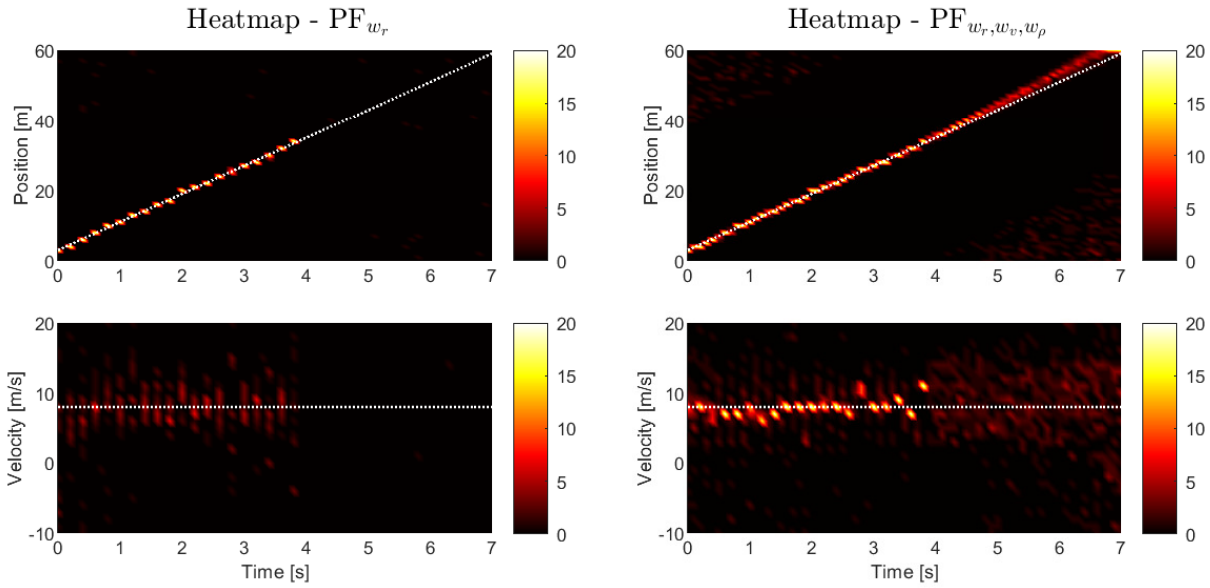
(a) PF with weighting criterion w_r

(b) PF with weighting criteria w_r and w_ρ

Figure 4.3: Heatmaps of the PF estimates over 20 runs for the scenario with loss of measurement. The top plot shows the position estimates and the bottom plot the velocity estimates. In both plots the GT is indicated in white.

Lastly, the addition of all three weighting criteria is compared to the PF with only w_r, in the constant velocity scenario. In Fig. 4.4 the improvements by both criteria, w_v and w_ρ, are visible. The accuracy of the velocity estimate has improved, up until measurements become scarce. Similarly, the longevity of both estimates has improved as is clear at the moment measurements become scare, i.e. at around 4 $[s]$.

The extension of the PF with the addition of weighting criteria has shown to improve both estimates, in accuracy and longevity. An additional decay factor, α, is added to try and improve the estimates in case of occlusion or loss of measurement. Therefore, the particle weights of the previous time step are included in the calculation of total weight per particle, as in Eq. (2.27). Afterwards, the total weights are normalised once more. The addition of $\alpha = 0.50$ is evaluated for all scenarios and the results are shown in Tab. 4.2.



(a) PF with weighting criterion w_r

(b) PF with weighting criteria w_r, w_v and w_ρ

Figure 4.4: Heatmaps of the PF estimates over 20 runs for the constant velocity scenario. The top plot shows the position estimates and the bottom plot the velocity estimates. In both plots the GT is indicated in white.

Scenario	GOSPA	MS	ME	FPR	FNR
Constant velocity **No decay - $\alpha = 0$**	$1.902_{\pm 1.274}$	$0.976_{\pm 0.007}$	$0.027_{\pm 0.007}$	$0.018_{\pm 0.006}$	$0.331_{\pm 0.155}$
Constant velocity 50% **decay - $\alpha = 0.50$**	$1.795_{\pm 1.049}$	$0.966_{\pm 0.004}$	$0.039_{\pm 0.004}$	$0.028_{\pm 0.004}$	$0.271_{\pm 0.120}$
Loss of measurement **No decay - $\alpha = 0$**	$1.053_{\pm 1.396}$	$0.978_{\pm 0.004}$	$0.047_{\pm 0.004}$	$0.034_{\pm 0.003}$	$0.147_{\pm 0.172}$
Loss of measurement 50% **decay - $\alpha = 0.50$**	$1.328_{\pm 1.524}$	$0.963_{\pm 0.005}$	$0.042_{\pm 0.005}$	$0.031_{\pm 0.004}$	$0.142_{\pm 0.143}$
changing velocity **No decay - $\alpha = 0$**	$1.749_{\pm 1.289}$	$0.981_{\pm 0.002}$	$0.022_{\pm 0.002}$	$0.015_{\pm 0.002}$	$0.193_{\pm 0.035}$
changing velocity 50% **decay - $\alpha = 0.50$**	$1.530_{\pm 1.104}$	$0.969_{\pm 0.003}$	$0.036_{\pm 0.003}$	$0.025_{\pm 0.003}$	$0.158_{\pm 0.022}$

Table 4.2: Evaluation results with the baseline settings, comparing the PF output with and without the decay factor. Mean values with the standard deviation are given for each metric.

Similar to Tab. 4.1 the better scores are highlighted in green, however now GOSPA scores can be compared as well. The addition of the decay term show no clear improvement overall. Especially, the GOSPA terms are quite close together with their large standard deviations. Instead, the MS seems to favour the PF without the decay term. Only the FNR has improved in all cases, but again the difference is smaller than the standard deviation on this metric. Therefore, this addition should be further investigated as it shows potential for improvement, but it is not distinctly better from this evaluation.

4.2 Parameter variations

To get an idea of how the algorithm can be adapted, a parameter study is performed. Similarly to the quality analysis in Section 4.1, the previously described metrics are used to quantify the results. First, a baseline score is set for all scenarios, with the base parameters. Therefore, an overview of available parameters is outlined. A selection of these parameters is varied and simulated. The results from these variations will give an indication on how to change the parameters to improve the algorithm performance.

4.2.1 Variations

Below a table is given with the parameters that are deemed most influential and are varied in these simulations. Appendix B lists the remaining parameters that are left out in Tab. B.1. In the table here, the baseline values are stated for each of the candidates and a short description is given. These parameters are selected as they are expected to have the most effect on the performance. Half are related to the PF component, however the other ones are of influence to the full algorithm.

Parameter	**Baseline values**	**Explanation**
Measurement frequency	5 [Hz]	Frequency at which measurements are generated by the sensor model in simulation.
Algorithm frequency	10 [Hz]	Frequency at which the mapping algorithm is executed.
Grid cell size	$20 \cdot 10^{-2}$ [m]	Grid cell length in 1D. Taken as equal to radar bin size.
Max. particles	$2 \cdot 10^{3}$ [-]	Maximum amount of particles in the population, which is a fixed amount during any simulation run.
Weight contributions	$\begin{pmatrix}1\\1\\1\end{pmatrix} \cdot \begin{bmatrix}w_r\\w_v\\w_\rho\end{bmatrix}$ [-]	Contributions of the individual weighting criteria to the total weight per particle.
Decay factor, α	0.50 [-]	See Eq. (2.27).

Table 4.3: Overview of the parameters which are varied, with their baseline values.

Each of the parameters is varied compared to the baseline and is simulated for all scenarios as listed in Section 3.1. For each parameter three values are simulated, both higher and lower values to get an indication of which direction to improve in. The table below gives an overview.

Parameter	**Values to be varied**
Variation 1 - Measurement frequency	$\{1, 2, 10\}$ [Hz]
Variation 2 - Algorithm frequency	$\{5, 20, 40\}$ [Hz]
Variation 3 - Grid cell size	$\{10, 40, 100\} \cdot 10^{-2}$ [m]
Variation 4 - Max. particles	$\{1, 4, 8\} \cdot 10^{3}$ [-]
Variation 5 - Weight contributions	$\begin{pmatrix}2\\1\\1\end{pmatrix}, \begin{pmatrix}1\\2\\1\end{pmatrix}, \begin{pmatrix}1\\1\\2\end{pmatrix} \cdot \begin{bmatrix}w_r\\w_v\\w_\rho\end{bmatrix}$ [-]
Variation 6 - Decay factor weights	$\{0.25, 0.75, 1.0\}$ [-]

Table 4.4: Overview of the values in the parameter study simulations.

4.2.2 Results

First the evaluation metrics are computed for all of the scenarios of Section 3.1, with the baseline settings as listed in Tab. 4.3. Each run is again simulated twenty times for 7 [s], the metrics are then computed as the average over the duration, computed from the average score of the twenty runs at each time interval.

Scenario	GOSPA	MS	ME	FPR	FNR
Constant velocity	$1.614_{\pm 1.227}$	$0.966_{\pm 0.004}$	$0.039_{\pm 0.005}$	$0.027_{\pm 0.005}$	$0.226_{\pm 0.147}$
Loss of measurement	$1.139_{\pm 1.442}$	$0.964_{\pm 0.005}$	$0.042_{\pm 0.005}$	$0.030_{\pm 0.005}$	$0.113_{\pm 0.138}$
Changing velocity	$1.530_{\pm 1.132}$	$0.969_{\pm 0.003}$	$0.036_{\pm 0.003}$	$0.025_{\pm 0.003}$	$0.158_{\pm 0.023}$
Occlusion	$4.477_{\pm 0.601}$	$0.956_{\pm 0.014}$	$0.050_{\pm 0.015}$	$0.030_{\pm 0.015}$	$0.433_{\pm 0.048}$
Combined	$4.525_{\pm 0.880}$	$0.934_{\pm 0.027}$	$0.074_{\pm 0.030}$	$0.048_{\pm 0.030}$	$0.672_{\pm 0.180}$

Table 4.5: Evaluation results of the algorithm with the baseline settings of Tab. 4.3 for all scenarios. Mean values with the standard deviation are given for each metric.

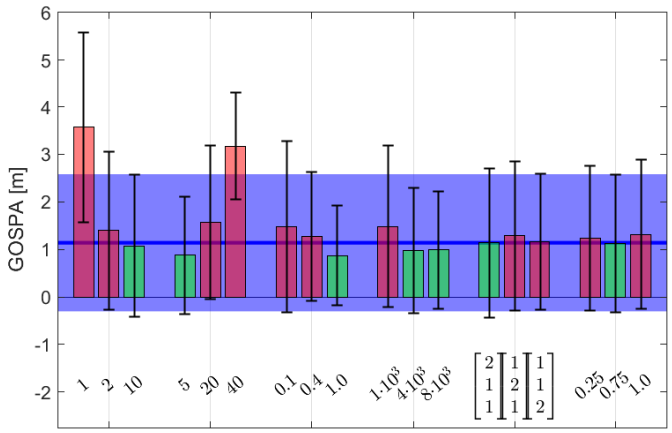
The evaluation results for the baseline parameters are given in Tab. 4.5. Similar to the highlighting in Tab. 4.1, if any of the GOSPA, MS or FNR metrics of the variations listed in Tab. 4.4 is better than the baseline, its cell is highlighted in green. To give an example this is done in the table. Only the GOSPA, MS and FNR metric are indicated as they are considered the most important. The GOSPA norm indicates error w.r.t. tracking occupancy estimates of objects, with the lower the average score, the better. Together with the FNR metric, tracking occupancy of objects can be evaluated. Again a lower FNR score indicates better performance. Both the MS and ME show similarity between the GT map and the generated environment map. As both are related, looking at only either of the metric is sufficient. Here the MS is highlighted, for which higher scores are better. The FPR is left out of the analysis as falsely identifying cells as occupied is less critical to safety than falsely identifying cells as free.

Highlighting the baseline scenarios as described, shows the best GOSPA and FNR scores for the loss of measurement scenario. It has to be noted that the GOSPA cannot be compared, if the amount of objects in the scenario is different. Due to the metric accounting for cardinality errors, average scores will in general be higher with more targets. As such the scenario with occlusion and combined scenario likely get higher average GOSPA scores. The other four metrics can be compared between scenarios. As stated in Section 3.3, these are normalised over all grid cells or into ratios such that they can be compared for different grid sizes or over different scenarios.

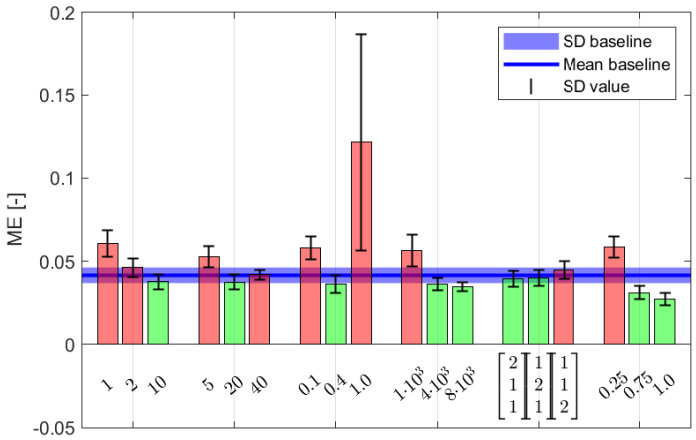
That the best GOSPA score in Tab. 4.5 belongs to the scenario with loss of measurement can be attributed to the accuracy of the PF estimate. Due to the implementation of the radar sensor model, measurements at long range are less accurate, failing detections more often and increasing measurement uncertainty. This probably causes the loss of measurement scenario to score better than the constant velocity scenario, as measurements are turned off when the target is further in range. This also holds for the FNR, which is best for the loss of measurement scenario as well. Additionally, the changing velocity scenario shows better performance in the MS norm. A possible explanation is again in the setup of this scenario. Due to the acceleration the target spends relatively more time closer to the sensor and less further in the range. Therefore, it will receive more accurate measurements. However, the differences are as small as they are only one standard deviation apart.

Only three scenarios will be simulated with the parameter variations, to limit the data to be analysed. The scenarios with loss of measurement, objects with changing velocity and with occlusion are selected to be varied. Then the constant velocity and combined scenarios serve as control groups and will only be checked with the baseline and improved parameters. The loss of measurement scenario is evaluated first.

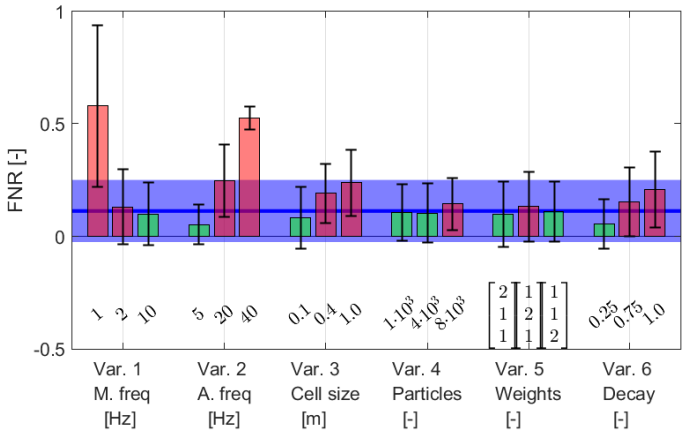
Instead of listing the results in one huge table, the results are summarised in a figure. However, in Appendix B.2 tables similar to Tab. 4.5 are given, so the exact values can be looked up. Below in Fig. 4.5 the GOSPA, ME and FNR metrics are displayed against their baseline values for the variations as listed in Tab. 4.4. Earlier the MS was compared, but here the ME metric is displayed as it gives a cleaner plot. However, as stated before, both metrics result in similar conclusions. Variations that have improved compared to the baseline, have their bar plotted in green, if not the bar is plotted in red. Behind all of the bars, the mean baseline value is plotted as the blue line, with its standard deviation indicated as the shaded blue area. Similarly, the bar plots show the mean value of each of the variations and they have their standard deviation indicated by the black error bars.



(a) Results GOSPA.



(b) Results ME.



(c) Results FNR.

Figure 4.5: Results of the evaluation metrics for the loss of measurement scenario. Mean values with standard deviation per variation are plotted in front of the baseline.

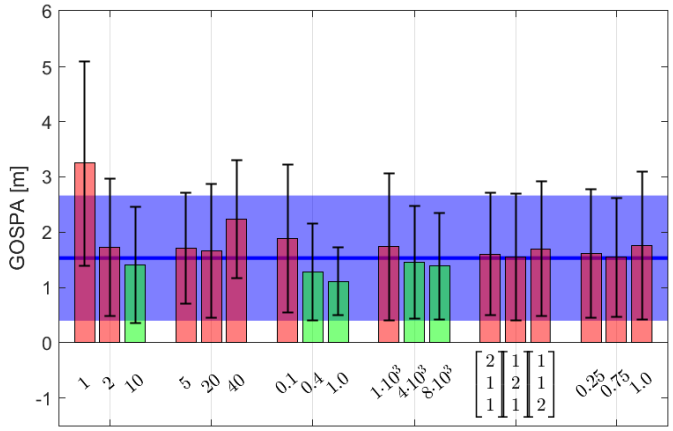
By looking at the outcomes in Fig. 4.5, the following conclusions can be drawn on the evaluation of the scenario with loss of measurement:

1. A higher measurement frequency leads to better scores overall. As more frequent measurements are received the occupancy grid map from the direct mapping will be more accurate. Similarly, the PF estimate will also benefit from more measurements.
2. In this scenario a lower algorithm frequency of 5 [Hz] seems best, which is half that of the baseline algorithm frequency and equal to the baseline measurement frequency. It has the best GOSPA and FNR scores, except for the ME which is best for a frequency of 20 [Hz]. A lower algorithm frequency decreases the resampling frequency, thus increasing longevity of clusters of particles which could explain the better GOSPA and FNR scores. Contrarily, it decreases the update frequency of the occupancy grid map, causing the higher ME score.
3. The results of the grid cell size are inconclusive, as each score indicates another outcome. The baseline results seem the best choice overall.
4. More particles in the population results in better performance for the GOSPA and ME metrics, with both scores closely matched. However, the FNR indicates an amount of $4 \cdot 10^3$ as the best option. This can be explained by a better approximation of the occupancy PDF, when more particles are available.
5. The variations in the weight contributions are all close to the baseline results. However increasing the contribution of w_r improves all scores slightly. As the object in this scenario has a constant velocity, the velocity estimate will be quite accurate after several iterations by the PF. Therefore, the increasing the contribution of the position weight probably has a larger influence in correcting the location of the occupancy estimate.
6. The results of changing the decay factor are inconclusive. But the baseline setting seems to give the overall best results. However, setting it to 0.75 gives better results in both the GOSPA and ME scores.

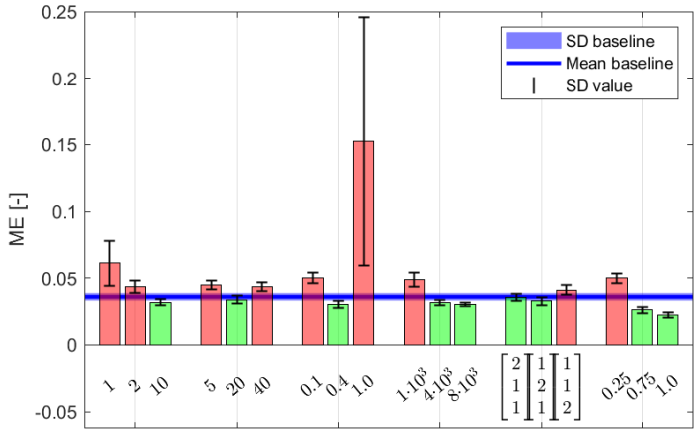
Concluding, from the loss of measurement scenario the following recommendations are made for adapting the parameters:

- Double the measurement frequency to 10 [Hz].
- Decrease the algorithm frequency to 5 [Hz].
- Increase the amount of particles in the particle population, either to $4 \cdot 10^3$ or $8 \cdot 10^3$.
- Increase the contribution of w_r to the total weight.
- Increase the decay factor of the weights to $\alpha = 0.75$.

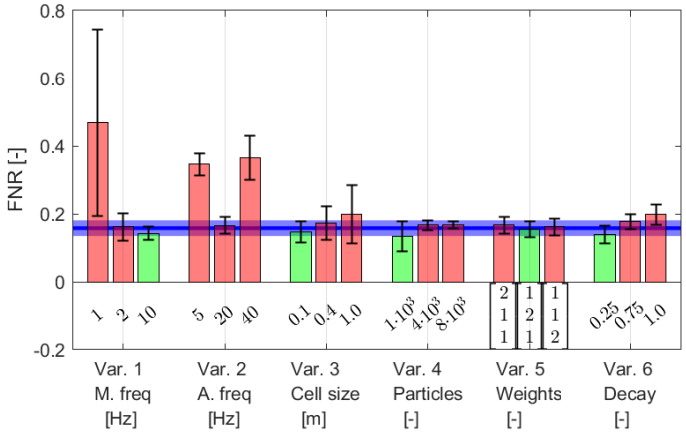
Secondly, the changing velocity scenario is evaluated and the results given in a similar figure. Again all variations are as listed in Tab. 4.4 and conclusions of the evaluation are summarised.



(a) Results GOSPA.



(b) Results ME.



(c) Results FNR.

Figure 4.6: Results of the evaluation metrics for the scenario with objects with changing velocity. Mean values with standard deviation per variation are plotted in front of the baseline.

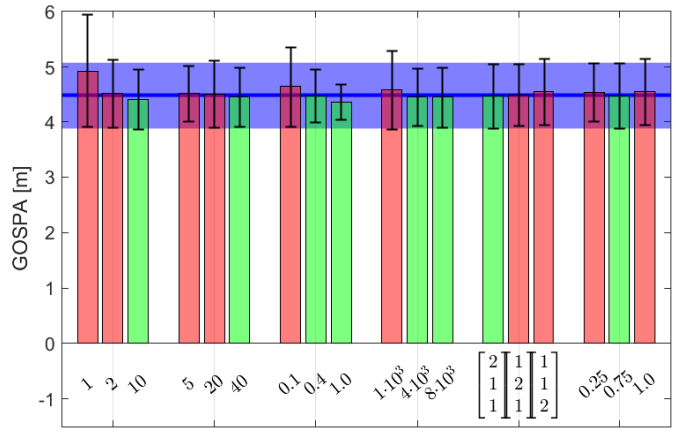
The conclusions from the evaluation on the changing velocity scenario of Fig. 4.6 are:

1. A higher measurement frequency of 10 [Hz] leads to better scores overall. This is likely due the more frequent updates to both the occupancy grid map and the PF estimate, resulting in more corrections by measurements to the occupancy estimate of the accelerating object.
2. The baseline algorithm frequency seems to perform best here. Likely, as the mapping relies on measurements to correct for the acceleration in the derived occupancy. Both the GOSPA and FNR score worse than the baseline for all variations. However, for a frequency of 20 [Hz] the ME does improve slightly. But still the baseline value is the best option out of the tried variations.
3. A larger grid cell size results in better scores for the GOSPA and ME metrics, but a higher FNR than the baseline. The FNR is best for the lowest grid cell size. So again this gives inconclusive results, however a choice of $40 \cdot 10^{-2}$ seems the overall best. For this option both the GOSPA and ME are better than the baseline, but the FNR is somewhat worse.
4. More particles in the population results in better performance in the GOSPA and MS metrics, with the best results for $8 \cdot 10^3$ particles. The FNR at this variation is just slightly worse than the baseline while it is best for the lowest amount of particles. However, the standard deviation is relatively high for this result.
5. Increasing the contribution of w_v scores best by all metrics, with the GOSPA being just higher than the baseline. Increasing this contribution likely captures the acceleration better in the PF estimate.
6. For the decay in the weights there is slight improvements for different variations. In general, however, the baseline seems to give good overall results.

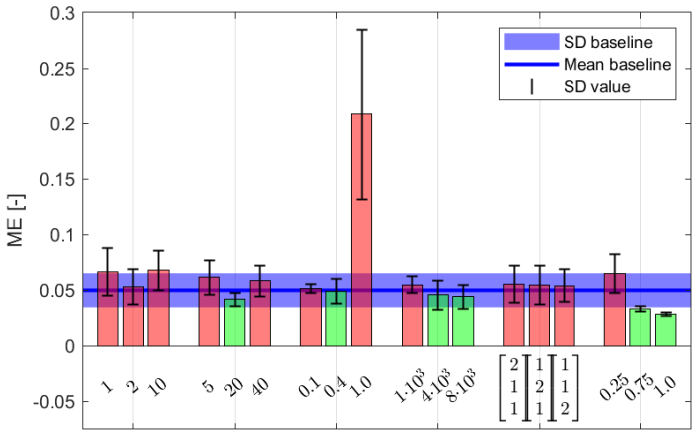
Concluding, from the scenario with objects with changing velocity the following recommendations are made for adapting the algorithm:

- Double the measurement frequency to 10 [Hz].
- Increase the grid cell size to $40 \cdot 10^{-2}$ [m].
- Increase the amount of particles in the population to $8 \cdot 10^3$.
- Increase the contribution of w_v to the total weight.

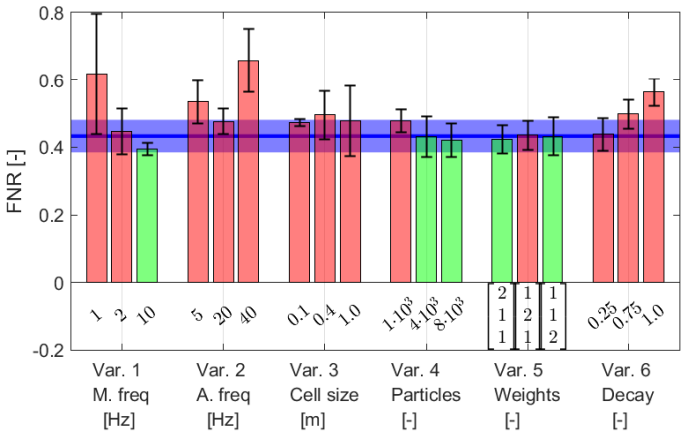
Lastly, the scenario with occlusion is evaluated. The results of this evaluation are given in a similar figure. Conclusions of this simulation are summarised below the figure.



(a) Results GOSPA.



(b) Results ME.



(c) Results FNR.

Figure 4.7: Results of the evaluation metrics for the scenario with occlusion. Mean values with standard deviation per variation are plotted in front of the baseline.

The results of the evaluation on the occlusion scenario in Fig. 4.7 are:

1. A higher measurement frequency of 10 [Hz] improves the metric scores, except for the ME, which is higher than the baseline. Similarly to the evaluation of the loss of measurement scenario, this can be attributed to more frequent measurements, which create more accurate mapping results.
2. The baseline algorithm frequency seems the best overal. A higher frequency of 20 [Hz] improves the ME, while the GOSPA barely improves for 40 [Hz]. The FNR indicates the baseline is best. So the results are inconclusive for this scenario and the baseline will be recommended.
3. A larger grid cell size results in a better scores for the GOSPA metric for both larger options, however only the $40 \cdot 10^{-2}$ [m] has a better ME than the baseline. The FNR is best for the baseline results. Therefore, the best results seem to be for $40 \cdot 10^{-2}$ [m].
4. More particles in the population results in better performance by all metrics. However, the best results are achieved for $8 \cdot 10^3$ particles. Once again, this can be attributed to a better PF estimate.
5. The variations in the weight contributions are all close to the baseline. Increasing the contribution of w_r, scores best by the GOSPA and FNR metrics and has an almost similar ME to the baseline.
6. Increasing the decay factor to 0.75 results in the best GOSPA score and gives an improvement of the ME. However, the FNR is best for the baseline.

Concluding from the scenario with occlusion the following recommendations are made for adapting the algorithm:

- Double the measurement frequency to 10 [Hz].
- Increase the grid cell size to $40 \cdot 10^{-2}$ [m].
- Increase the amount of particles in the particle population to $8 \cdot 10^3$.
- Increase the contribution of w_r to the total weight.
- Increase the decay factor of the weights to 0.75.

4.3 Algorithm improvements

This section summarises the recommended adaptations to the parameters of the algorithm, resulting from the parameter study. In the next section a recommendation is given for a set of improved parameters, which is then simulated and compared against the earlier established baseline results. In the second section the benefits of including the velocity measurement information in the algorithm are detailed.

4.3.1 Adapted parameters

By combining the conclusions on the three scenarios of the parameter study of Section 4.2, several improvements on adapting the algorithm are identified. The recommended adaptations to the parameter set are summarised in Tab. 4.6, below.

Parameter	Recommended value
Measurement frequency	10 [Hz]
Algorithm frequency	10 [Hz]
Grid cell size	$40 \cdot 10^{-2}$ [m]
Max. particles	$8 \cdot 10^3$ [-]
Weight contributions	$\begin{bmatrix}3\\2\\1\end{bmatrix} \cdot \begin{bmatrix}w_r\\w_v\\w_\rho\end{bmatrix}$ [-]
Decay factor weights	0.75 [-]

Table 4.6: The adapted parameter set as a result of the parameter study.

Each scenario recommends to increase the measurement frequency to 10 [Hz], since that increases the overall mapping accuracy. The results on the algorithm frequency are conflicted, therefore that will remain at 10 [Hz], which is equal to the recommended value for the measurement frequency. An increased grid cell size gave better results in two out of the three scenarios, therefore it is increased to $40 \cdot 10^{-2}$ [m]. Similarly, an increase in the amount of particles gave better results in two out of the three cases, with $8 \cdot 10^3$ particles as the recommended amount. This results stems from the increased accuracy of the occupancy estimate by the PF due to the higher amount of available particles. Therefore the occupancy PDF can be better approximated.

For the particle weights, the contributions of both w_r and w_v will be increased. In the scenario with objects with changing velocity, increasing the weight on velocity improved the metrics scores most. In the other scenarios increasing the weight on the position improved the results. Therefore, the contribution of w_r will be increased more than that of w_v. Lastly, the results for the decay factor leads us to increase the decay factor to $\alpha = 0.75$. According to Eq. (2.27), this increases the contribution of the newly assigned weights in the total weight equation. Thus giving relatively more confidence to measurement information, analogous to a higher Kalman gain in Kalman filters.

The algorithm is simulated to confirm if the performance improves with the recommended parameters set as in Tab. 4.6. This is done for all of the scenarios and the results are displayed below in Tab. 4.7. These values are compared to the baseline simulation results of Tab. 4.5 and highlighted green if they have improved or orange if it has scored worse. Overall, it is clear that the performance has improved as from the GOSPA and MS metrics. Only the FNR is higher in several scenarios. But overall it can be concluded that the recommended improvements have resulted in better scores and a better performing algorithm.

Scenario	GOSPA	MS	ME	FPR	FNR
Constant velocity	$0.407_{\pm 0.550}$	$0.986_{\pm 0.002}$	$0.017_{\pm 0.002}$	$0.010_{\pm 0.002}$	$0.131_{\pm 0.039}$
Loss of measurement	$0.543_{\pm 0.685}$	$0.982_{\pm 0.003}$	$0.021_{\pm 0.003}$	$0.012_{\pm 0.003}$	$0.184_{\pm 0.107}$
changing velocity	$1.047_{\pm 0.520}$	$0.981_{\pm 0.001}$	$0.022_{\pm 0.001}$	$0.012_{\pm 0.001}$	$0.326_{\pm 0.012}$
Occlusion	$4.311_{\pm 0.317}$	$0.969_{\pm 0.005}$	$0.035_{\pm 0.005}$	$0.010_{\pm 0.005}$	$0.474_{\pm 0.023}$
Combined	$4.282_{\pm 0.398}$	$0.967_{\pm 0.008}$	$0.037_{\pm 0.009}$	$0.009_{\pm 0.008}$	$0.586_{\pm 0.082}$

Table 4.7: Results of the evaluation metrics for the given scenarios, with the tuned algorithm. Average values and standard deviations are given and improvements compared to the baseline are highlighted in green and deteriorated scores in orange.

4.3.2 Velocity information

Additionally, the algorithm is tested with and without using velocity measurement information, by removing the weighting criterion based on velocity in the PF. This is done to test the hypothesis that the algorithm improves by including velocity information. The results are displayed in Tab. 4.8 and again compared to the baseline results of Tab. 4.5, by highlighting scores. Most scores have gotten worse, thus it can be concluded that including velocity information in the PF estimate improves the occupancy estimation.

Scenario	GOSPA	MS	ME	FPR	FNR
Constant velocity	$1.107_{\pm 1.186}$	$0.958_{\pm 0.006}$	$0.048_{\pm 0.006}$	$0.036_{\pm 0.008}$	$0.067_{\pm 0.107}$
Loss of measurement	$1.274_{\pm 1.694}$	$0.957_{\pm 0.006}$	$0.049_{\pm 0.006}$	$0.038_{\pm 0.008}$	$0.098_{\pm 0.147}$
changing velocity	$1.803_{\pm 1.234}$	$0.956_{\pm 0.006}$	$0.050_{\pm 0.006}$	$0.040_{\pm 0.009}$	$0.157_{\pm 0.044}$
Occlusion	$4.558_{\pm 0.625}$	$0.948_{\pm 0.009}$	$0.059_{\pm 0.010}$	$0.038_{\pm 0.013}$	$0.455_{\pm 0.057}$
Combined	$4.745_{\pm 0.880}$	$0.893_{\pm 0.036}$	$0.120_{\pm 0.040}$	$0.089_{\pm 0.044}$	$0.686_{\pm 0.207}$

Table 4.8: Results of the evaluation metrics for the given scenarios, with the baseline algorithm but without including velocity measurement information. Scores are compared to the baseline algorithm by highlighting improvements in green and deteriorated scores in orange.

Similar to the quality analysis of Section 4.1, a comparison can be made of the combined position and velocity estimates with and without including velocity measurement information. To that end Fig. 4.8 shows the combined estimates derived from the local maxima of the occupancy estimate by the PF. This is the same data that is used to compute the GOSPA metric score. The top plot shows the situation without w_v in the PF and the bottom plot with w_v. The latter clearly shows less spread on the velocity estimates, similar to what the heat maps in Fig. 4.2 showed. Thus inclusion of the velocity measurement information improves the velocity estimates on the target. This will lead to a more accurate occupancy mapping.

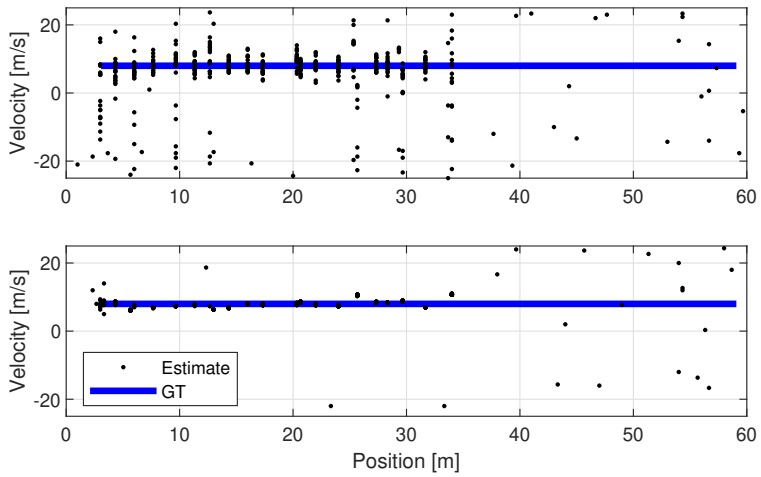


Figure 4.8: A density plot of the combined position and velocity estimates as derived from the PF during 20 runs of the constant velocity scenario. The top plot shows the results without w_v, while in the bottom plot w_v is included.

4.4 Summary

First a quality analysis was performed to determine if the additions done to the PF, as the added weighting criteria, provide improved mapping results. This was evaluated by looking at position and velocity estimates derived from the approximated occupancy and comparing this with and without these additions. It showed that the added weighting criteria improve both accuracy and longevity of the estimates and thus the occupancy estimation by the PF. Additionally, a parameter variation was done to give recommendations on how to adapt the algorithm in specific scenarios. The resulting recommended parameter set is shown in Section 4.3.1. This set was simulated once more to show the improved performance compared to the baseline algorithm. The chapter concluded by highlighting the improvements made to the concept dynamic environment mapping algorithm, by incorporating velocity measurement information.

5 Conclusion and Recommendations

The main goal of this study was to develop and research a dynamic environment mapping approach to map the local environment around a moving vehicle. The framework should support multi-view and multi-sensor data from several sensor types, while also being able to incorporate measurement and pose uncertainties. In this approach, the occupancy resulting from the dynamic environment should be estimated and properly represented, without using target detection and tracking methods. Lastly, the research should focus on using available velocity measurement information to improve the mapping. This chapter provides the final remarks on the results of this study and gives recommendations on future research.

5.1 Conclusions

This report provided background on environment perception and mapping approaches in Chapter 1. In this literature study the occupancy grid map was put forward as a suitable candidate framework for the mapping algorithm. It creates a map by tessellation of the area around the ego vehicle in grid cells. These grid cells are assigned an occupancy probability based on measurements of the environment. This probabilistic approach enables combining data from different sensors and incorporation of uncertainties. Based on this framework the dynamic occupancy grid mapping algorithm was developed, in which an occupancy grid map generated from measurements is combined with occupancy estimates by a filter method.

In Chapter 2 the concept mapping algorithm is explained. In the direct mapping component a shaped function, based on an inverse sensor model, is applied to translate the measurements to occupancy probabilities. These probabilities are then updated in a log-likelihood update equation to receive posterior occupancy probabilities for the grid cells. In parallel a particle filter is used to provide a prediction of the occupancy, which is corrected by both position and velocity measurement information. The particles are assigned position and velocity states and are updated by a CVCO process model. Three weighting criteria are used to assign weights to each of the predicted particles. Both a position and a velocity weighting criterion are applied to assign higher weights to particles correctly approximating the occupancy. A third criterion assigns more weight to dense clusters of particles to preserve estimates on occupancy of which measurements are missed. In the resampling step the most relevant particles are duplicated to a new set to create the posterior occupancy PDF. The PF estimate is then translated to occupancy probabilities and combined with the direct mapping output to the dynamic occupancy grid map.

The presented concept has been worked out in a 1D setup, with an extension such that it can be tested and evaluated with multiple, simultaneous measurements. In Section 2.1 the simulation environment is described, together with the sensor model that is applied to generate measurements. Chapter 3 proceeds to detail several simulation scenarios that are used to evaluate the functionality, quality and performance of the algorithm under specific conditions. The functional analysis is performed to determine if the algorithm functions as intended. It concludes that the mapping algorithm works as intended but has two limitations. First, the PF has trouble with estimating occupancy from accelerating objects due to its CVCO process model. Secondly, estimates of occupancy for objects of which detections are lost, do not survive long. This is mainly the case when detections on other objects are present. For the latter case the third weighting criterion, based on the amount of particles per grid cell, and the decay factor, α, were added to the PF. However, according to this first functional analysis their effect is only limited.

In Chapter 4 a quality analysis is performed to determine if the additions done to the PF, as the added weighting criteria, provide improved mapping results. This is evaluated by comparing position and velocity estimates of the occupancy with and without these additions. It shows that the added weighting criteria improve both accuracy and longevity of the estimates and thus the occupancy estimation by the PF. Additionally, a parameter study is done to give recommendations on how to adapt the algorithm in spe-

cific scenarios. The resulting recommended parameter set is shown in Section 4.3 and simulated to show the improved performance compared to the baseline algorithm. This chapter concludes by highlighting the improvements made to the concept dynamic environment mapping algorithm, by incorporating velocity measurement information.

In conclusion, the main research question is answered by providing a concept dynamic environment mapping algorithm that has been shown to have improved the mapping performance by incorporation of velocity measurement information. The concept has been worked out in a 1D setup in simulation and is proven to function correctly in that context. Various evaluations have been performed to asses functionality, quality and get an indication on how to adapt parameters to improve the algorithm. The performance is mainly limited by the process model of the PF and by the longevity of estimates without new detections. Therefore, some recommendations will be listed in the next section to improve on this. However, this concept is ready to be developed to an implementable algorithm.

5.2 Recommendations

As concluded, the concept is tested well in the 1D setup and is ready to be developed to an implementable algorithm. However, some limitations were identified during the evaluations. These can mainly be attributed to the non-parametric nature of the filter and can be improved by additions to the PF. Section 3.2.2 recommends to change or add to the weighting criteria, further investigation of the decay term α in Eq. (2.27) and improvements to the process model.

For one, the current CVCO process model limits estimation of targets with acceleration. The added process noise can only capture a limited range of accelerations through stochastic diffusion of the velocity state. However, more elegant solutions might be available. Secondly, the weighting step can be adapted to try and improve longevity of estimates. For example, the addition of specific weighting criterion or changing contributions of criteria to the total weight can be investigated. The current three weighting criteria have already been proven to improve the occupancy estimate in the quality analysis. Also, the parameter study showed that performance can improve in specific scenarios by changing the contributions.

To evaluate the longevity of estimates, it is beneficial to add an evaluation criterion that can judge retention of occupancy estimates. In Section 3.3 the GOSPA has already been proposed, which can achieve this. However, the determination of local maxima from the particle population is not ideal. A decay factor, α, has also been tried to improve longevity. However, its effect on the performance was limited at best, as shown in the quality analysis. Therefore, it is recommended to remove this parameter from the PF as it might obscure the influence of the weighting criterion, w_ρ, on the longevity of the estimates.

The main research question stated to realise the mapping approach on a moving vehicle. In this research the concept has been worked out, assuming a static ego vehicle in a dynamic environment. So pose uncertainty and merging of subsequent maps has not been investigated yet. However, due to the probabilistic nature of the grid map framework, incorporating uncertainties and merging of maps can be performed in an intuitive way.

Lastly, the concept has to be extended to at least a 2D setup, before implementation. The current 1D setup has been used to work out the concept, however it is not suited for use on the test vehicles. In a 2D or higher dimension, care has to be taken to correctly extend the 1D setup to additional dimensions. However, the current algorithm is not bound to specific 1D dimensions and an extension should be straightforward by adaptation of the grid. In conclusion, the presented concept shows potential to realise all of the listed recommendations in future research.

Appendices

A Evaluation scenarios

This appendix shows the progression of each of the scenarios listed in Tab. 3.1, by 10 snapshots of the objects in motion for the duration of the simulation, i.e. 7 $[s]$. Each plot will show the objects by their position and velocity and is marked with a time-stamp in the top left corner.

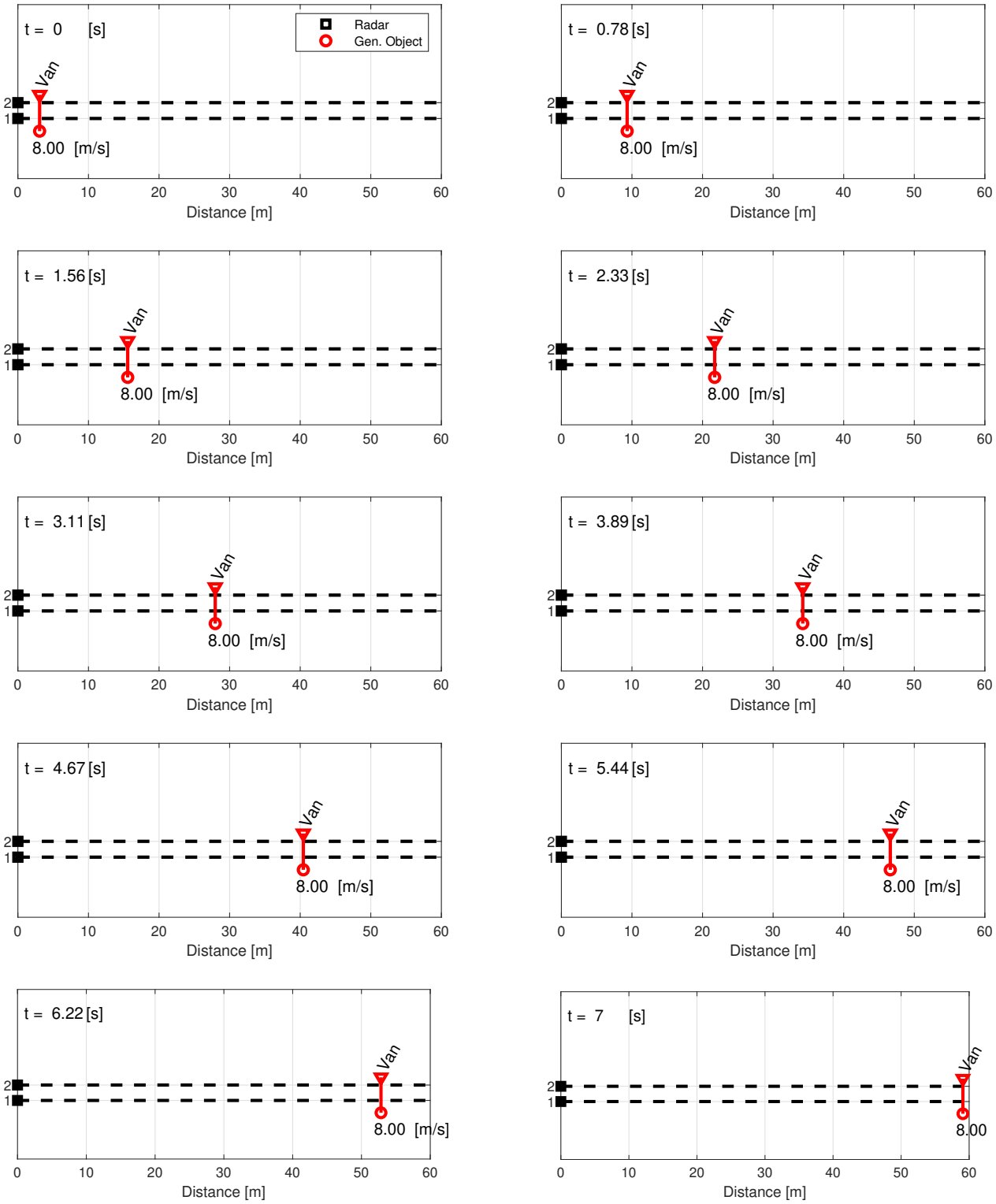


Figure A.1: Progression of the constant velocity scenario

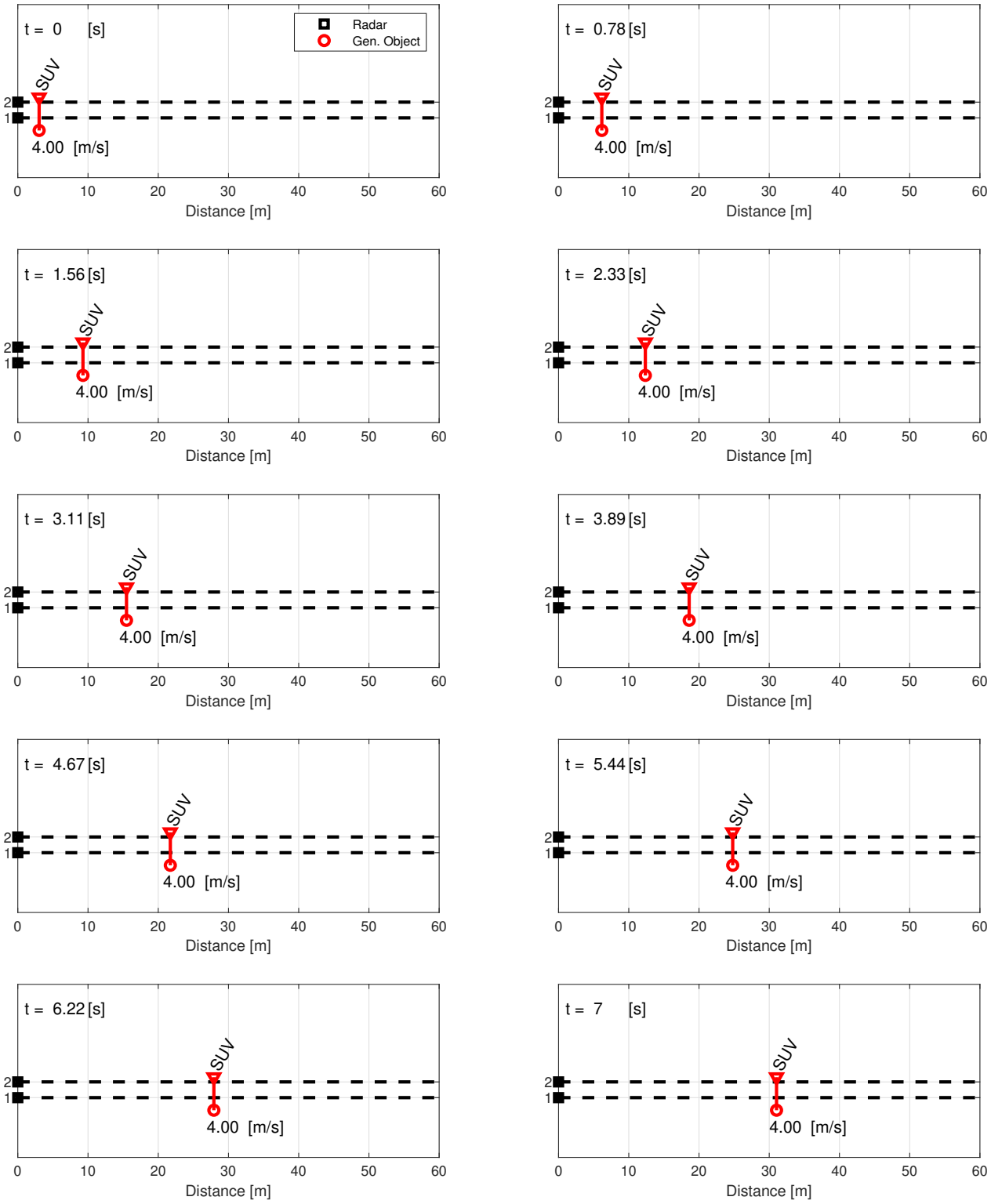


Figure A.2: Progression of the scenario with loss of measurement

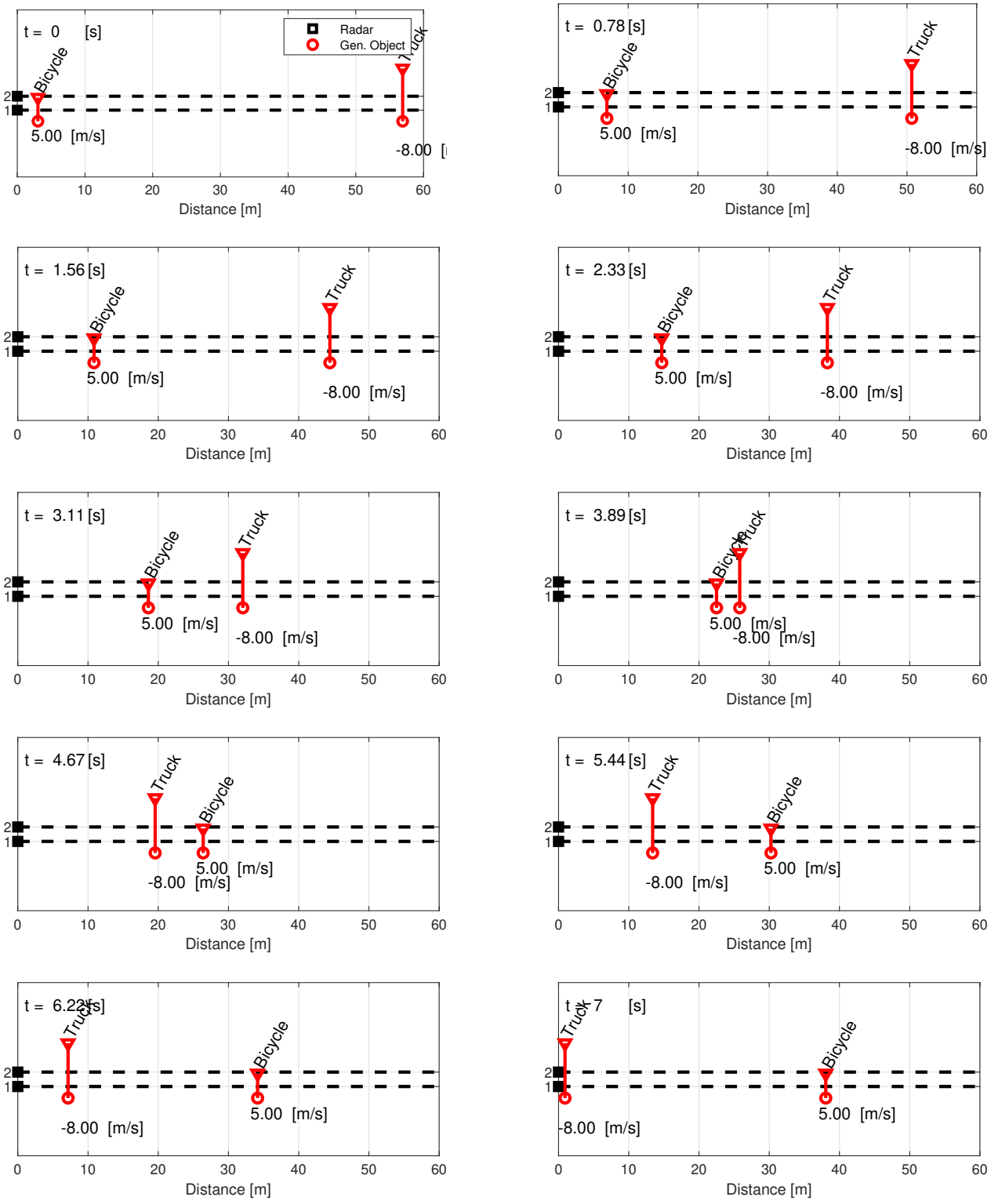


Figure A.3: Progression of the scenario with occlusion

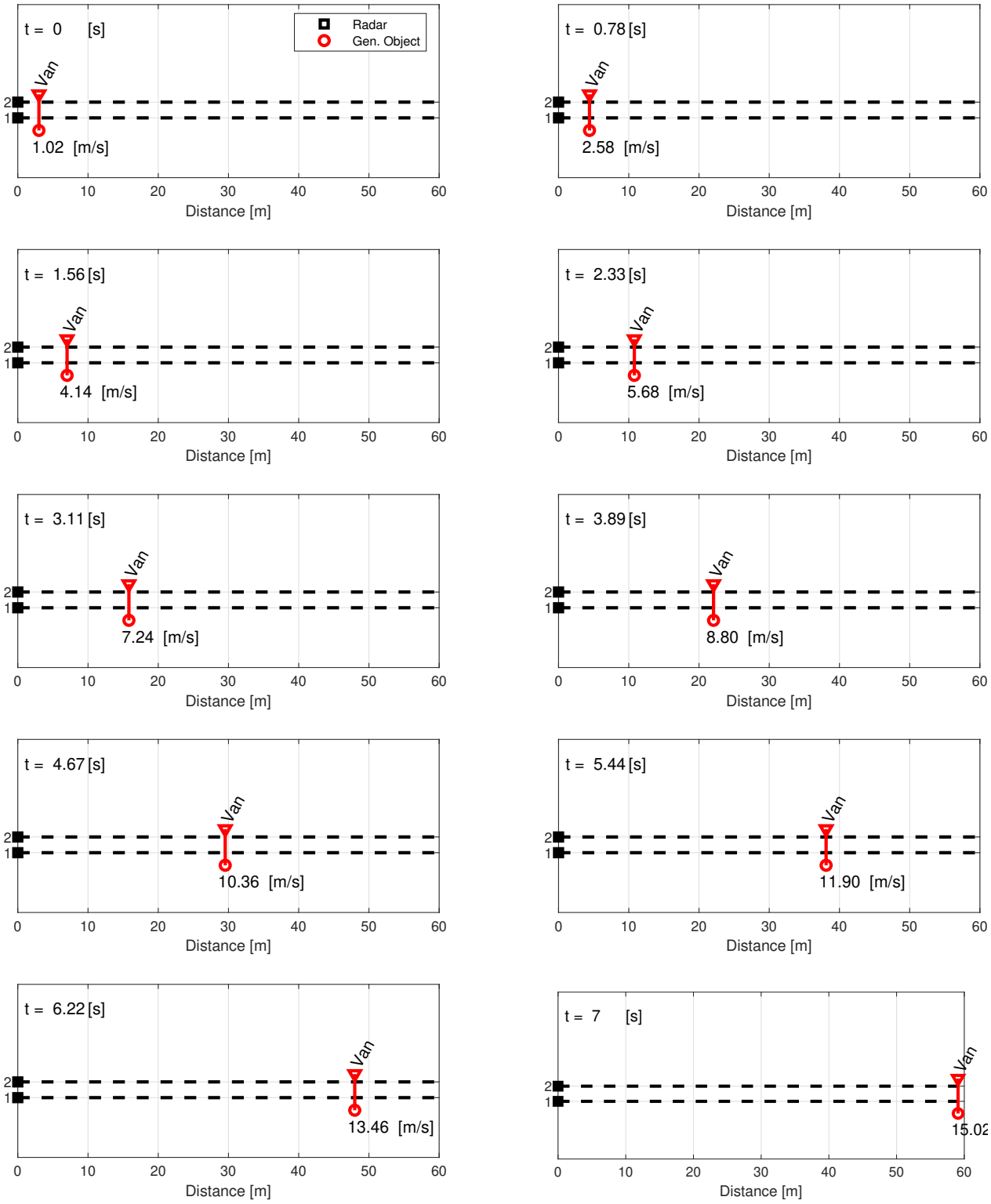


Figure A.4: Progression of the scenario with objects with changing velocity

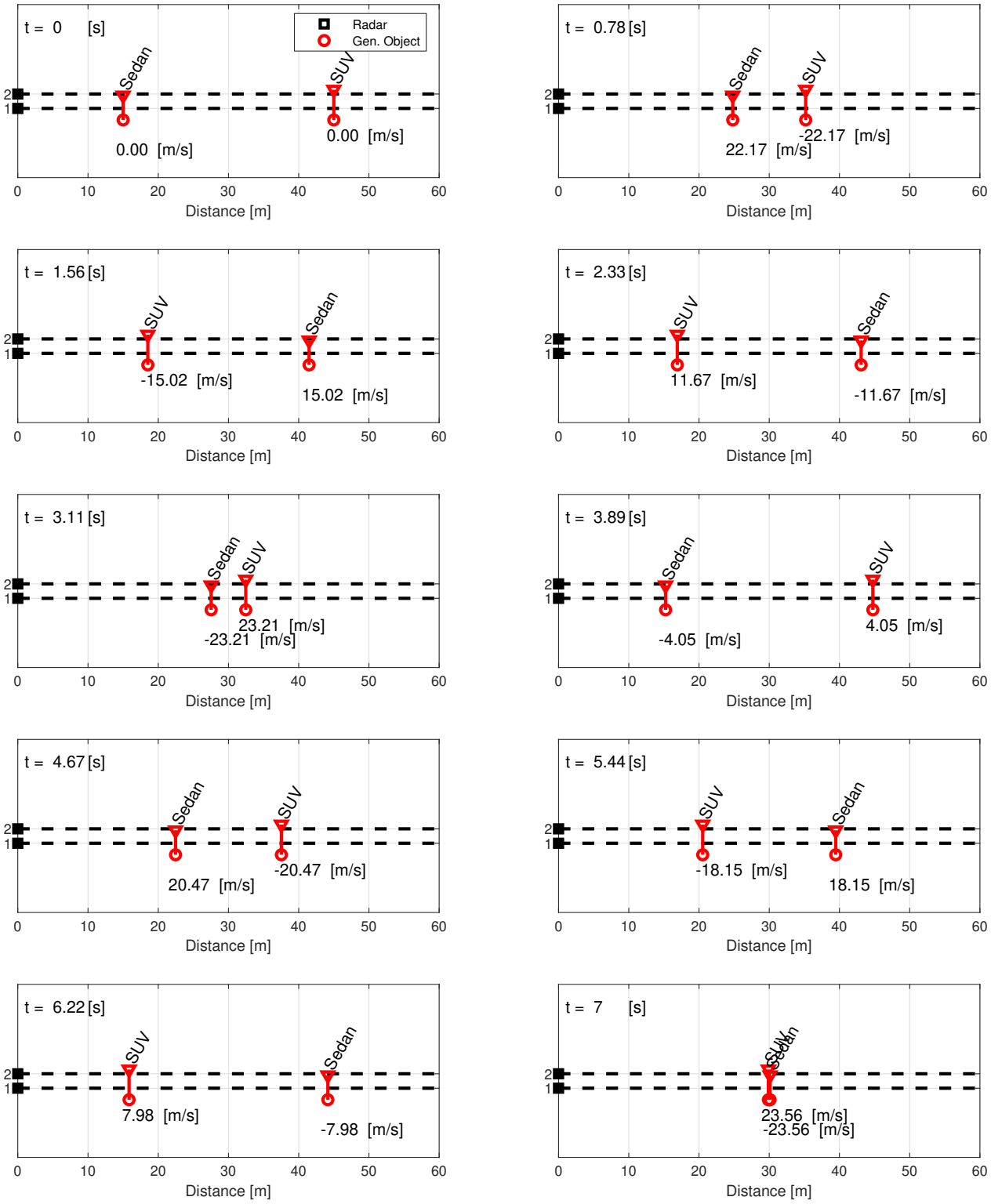


Figure A.5: Progression of the combined scenario

B Evaluation simulations

This appendix shows the remaining parameters and data of the evaluation results for Chapter 4.

B.1 Parameter overview

The table below shows parameters that are in the algorithm, but are not chosen for the parameter study simulations of Section 4.2. Similarly to Tab. 4.3 the table lists baseline values and a short description of each parameter. An additional column is present which lists if the variable can be varied or if it is fixed.

Parameter	**Baseline values**	**Fixed**	**Explanation**
Decay in log odds ratio	0.50	Candidate	The ζ term in Eq. (2.13).
Association probability	Grid cell size [m]	Candidate	Area around particle to consider measurements relevant.
Sensor variances	-	Per sensor type	σ_r, σ_{v_r}
Reference sensor values	-	Per sensor type	E.g. $p_{d,ref}$, p_{fa}, RCS_{ref}, R_{ref} for radar.
Other sensor properties	-	Per sensor type	E.g. amount, placement, etc.
Map creation parameters	-	Mostly	E.g. map size, dimension, etc.
Scenario definitions	-	Mostly	Sine and random scenarios can be varied, others are fixed.
Object specifications	-	Yes	E.g. dimensions, velocities, etc.
Prior shaping function	-	Yes	Parameters for function shaping.
Log odds ratio updating	-	Yes	Parameters for update equation.
Finding local maxima	-	Yes	Parameters in search function.
Evaluation metrics	-	Yes	Parameters to compute metrics.

Table B.1: Remaining parameters that are candidate to be varied in simulation.

Additionally, the object properties are listed in the table below. These were first discussed in Section 2.1.3 to explain the use of object properties in the sensor model. Values are chosen to loosely represent actual values for the object types.

Object type	**Dimensions** [m] Height Width Length	**Velocities** $[\frac{m}{s}]$ Min. velocity Max. velocity	**Accelerations** $[\frac{m}{s^2}]$ Min. acceleration Max. acceleration
Truck	3.80 2.60 16.00	4.00 16.00	0.00 1.50
Van	2.30 2.00 5.00	4.00 16.00	0.00 2.00
SUV	2.10 1.90 4.80	4.00 16.00	0.00 2.50
Sedan	1.70 1.80 4.50	4.00 16.00	0.00 3.00
Hatchback	1.60 1.80 4.20	4.00 16.00	0.00 3.50
Bicycle	1.70 0.70 2.00	2.00 10.00	0.00 2.00
Hatchback	1.90 0.70 0.70	0.50 2.50	0.00 1.50

Table B.2: Object types and their properties.

B.2 Parameter study results

In the three tables below the evaluation results are given for the parameter study simulations of Section 4.2. These are displayed in the figures of that chapter, however for reference they are given here as well. Each table lists the average metric scores and their standard deviations for the variations of Tab. 4.4. The baseline scores are listed at the tob of each table. Per variation the best score per metric is highlighted in green. If the best score of any metric for the given variation is not better than the baseline that score will be highlighted in orange instead.

Variation	**GOSPA**	**MS**	**ME**	**FPR**	**FNR**
Baseline	$1.139_{\pm 1.442}$	$0.964_{\pm 0.005}$	$0.042_{\pm 0.005}$	$0.030_{\pm 0.005}$	$0.113_{\pm 0.138}$
Variation 1 - 1 **[Hz]**	$3.581_{\pm 2.003}$	$0.945_{\pm 0.008}$	$0.061_{\pm 0.008}$	$0.044_{\pm 0.007}$	$0.580_{\pm 0.361}$
Variation 1 - 2 **[Hz]**	$1.399_{\pm 1.664}$	$0.960_{\pm 0.006}$	$0.046_{\pm 0.006}$	$0.035_{\pm 0.006}$	$0.130_{\pm 0.166}$
Variation 1 - 10 **[Hz]**	$1.071_{\pm 1.492}$	$0.967_{\pm 0.004}$	$0.038_{\pm 0.004}$	$0.027_{\pm 0.004}$	$0.099_{\pm 0.139}$
Variation 2 - 5 **[Hz]**	$0.879_{\pm 1.236}$	$0.954_{\pm 0.006}$	$0.053_{\pm 0.006}$	$0.038_{\pm 0.007}$	$0.052_{\pm 0.088}$
Variation 2 - 20 **[Hz]**	$1.574_{\pm 1.612}$	$0.967_{\pm 0.005}$	$0.038_{\pm 0.005}$	$0.027_{\pm 0.004}$	$0.246_{\pm 0.162}$
Variation 2 - 40 **[Hz]**	$3.169_{\pm 1.123}$	$0.962_{\pm 0.003}$	$0.042_{\pm 0.003}$	$0.028_{\pm 0.003}$	$0.525_{\pm 0.052}$
Variation 3 - $10 \cdot 10^{-2}$ **[m]**	$1.479_{\pm 1.800}$	$0.949_{\pm 0.007}$	$0.058_{\pm 0.007}$	$0.041_{\pm 0.006}$	$0.081_{\pm 0.137}$
Variation 3 - $40 \cdot 10^{-2}$ **[m]**	$1.275_{\pm 1.354}$	$0.969_{\pm 0.005}$	$0.036_{\pm 0.005}$	$0.023_{\pm 0.005}$	$0.191_{\pm 0.131}$
Variation 3 - $100 \cdot 10^{-2}$ **[m]**	$0.870_{\pm 1.052}$	$0.891_{\pm 0.062}$	$0.122_{\pm 0.065}$	$0.102_{\pm 0.065}$	$0.238_{\pm 0.147}$
Variation 4 - $1 \cdot 10^{3}$ **[-]**	$1.483_{\pm 1.704}$	$0.951_{\pm 0.009}$	$0.056_{\pm 0.009}$	$0.039_{\pm 0.009}$	$0.105_{\pm 0.125}$
Variation 4 - $4 \cdot 10^{3}$ **[-]**	$0.977_{\pm 1.315}$	$0.968_{\pm 0.004}$	$0.036_{\pm 0.004}$	$0.027_{\pm 0.004}$	$0.104_{\pm 0.130}$
Variation 4 - $8 \cdot 10^{3}$ **[-]**	$0.989_{\pm 1.239}$	$0.969_{\pm 0.003}$	$0.035_{\pm 0.003}$	$0.026_{\pm 0.003}$	$0.144_{\pm 0.115}$
Variation 5 - $[2;\ 1;\ 1]$ **[-]**	$1.137_{\pm 1.565}$	$0.966_{\pm 0.005}$	$0.040_{\pm 0.005}$	$0.029_{\pm 0.005}$	$0.099_{\pm 0.145}$
Variation 5 - $[1;\ 2;\ 1]$ **[-]**	$1.282_{\pm 1.562}$	$0.965_{\pm 0.005}$	$0.040_{\pm 0.005}$	$0.029_{\pm 0.005}$	$0.132_{\pm 0.155}$
Variation 5 - $[1;\ 1;\ 2]$ **[-]**	$1.166_{\pm 1.431}$	$0.961_{\pm 0.005}$	$0.045_{\pm 0.005}$	$0.034_{\pm 0.007}$	$0.108_{\pm 0.133}$
Variation 6 - 0.25 **[-]**	$1.239_{\pm 1.527}$	$0.949_{\pm 0.006}$	$0.059_{\pm 0.006}$	$0.050_{\pm 0.008}$	$0.055_{\pm 0.109}$
Variation 6 - 0.75 **[-]**	$1.123_{\pm 1.448}$	$0.973_{\pm 0.004}$	$0.031_{\pm 0.004}$	$0.023_{\pm 0.004}$	$0.152_{\pm 0.153}$
Variation 6 - 1.0 **[-]**	$1.313_{\pm 1.572}$	$0.976_{\pm 0.004}$	$0.027_{\pm 0.004}$	$0.019_{\pm 0.003}$	$0.208_{\pm 0.170}$

Table B.3: Results of the evaluation metrics for the scenario with loss of measurement. Average values and standard deviations are given for all metrics and improvements compared to the baseline are highlighted per variation.

Variation	GOSPA	MS	ME	FPR	FNR
Baseline	$1.530_{\pm 1.132}$	$0.969_{\pm 0.003}$	$0.036_{\pm 0.003}$	$0.025_{\pm 0.003}$	$0.158_{\pm 0.023}$
Variation 1 - 1 **[Hz]**	$3.248_{\pm 1.852}$	$0.945_{\pm 0.016}$	$0.061_{\pm 0.017}$	$0.044_{\pm 0.015}$	$0.469_{\pm 0.274}$
Variation 1 - 2 **[Hz]**	$1.727_{\pm 1.237}$	$0.962_{\pm 0.005}$	$0.044_{\pm 0.005}$	$0.032_{\pm 0.005}$	$0.161_{\pm 0.041}$
Variation 1 - 10 **[Hz]**	$1.403_{\pm 1.048}$	$0.973_{\pm 0.002}$	$0.032_{\pm 0.002}$	$0.022_{\pm 0.003}$	$0.142_{\pm 0.020}$
Variation 2 - 5 **[Hz]**	$1.717_{\pm 1.002}$	$0.960_{\pm 0.003}$	$0.045_{\pm 0.003}$	$0.029_{\pm 0.003}$	$0.346_{\pm 0.032}$
Variation 2 - 20 **[Hz]**	$1.656_{\pm 1.211}$	$0.971_{\pm 0.003}$	$0.034_{\pm 0.003}$	$0.023_{\pm 0.003}$	$0.165_{\pm 0.025}$
Variation 2 - 40 **[Hz]**	$2.239_{\pm 1.066}$	$0.961_{\pm 0.003}$	$0.043_{\pm 0.003}$	$0.031_{\pm 0.004}$	$0.365_{\pm 0.066}$
Variation 3 - $10 \cdot 10^{-2}$ **[m]**	$1.877_{\pm 1.337}$	$0.956_{\pm 0.004}$	$0.050_{\pm 0.004}$	$0.035_{\pm 0.005}$	$0.146_{\pm 0.031}$
Variation 3 - $40 \cdot 10^{-2}$ **[m]**	$1.282_{\pm 0.880}$	$0.975_{\pm 0.002}$	$0.030_{\pm 0.003}$	$0.017_{\pm 0.003}$	$0.173_{\pm 0.050}$
Variation 3 - $100 \cdot 10^{-2}$ **[m]**	$1.112_{\pm 0.610}$	$0.863_{\pm 0.087}$	$0.152_{\pm 0.093}$	$0.132_{\pm 0.095}$	$0.198_{\pm 0.085}$
Variation 4 - $1 \cdot 10^{3}$ **[-]**	$1.736_{\pm 1.325}$	$0.958_{\pm 0.005}$	$0.049_{\pm 0.005}$	$0.034_{\pm 0.007}$	$0.134_{\pm 0.045}$
Variation 4 - $4 \cdot 10^{3}$ **[-]**	$1.454_{\pm 1.016}$	$0.973_{\pm 0.002}$	$0.032_{\pm 0.002}$	$0.023_{\pm 0.002}$	$0.167_{\pm 0.015}$
Variation 4 - $8 \cdot 10^{3}$ **[-]**	$1.389_{\pm 0.962}$	$0.974_{\pm 0.001}$	$0.030_{\pm 0.001}$	$0.022_{\pm 0.002}$	$0.168_{\pm 0.011}$
Variation 5 - $[2;\ 1;\ 1]$ **[-]**	$1.600_{\pm 1.106}$	$0.969_{\pm 0.003}$	$0.035_{\pm 0.003}$	$0.025_{\pm 0.003}$	$0.166_{\pm 0.024}$
Variation 5 - $[1;\ 2;\ 1]$ **[-]**	$1.555_{\pm 1.148}$	$0.972_{\pm 0.003}$	$0.033_{\pm 0.003}$	$0.023_{\pm 0.003}$	$0.155_{\pm 0.024}$
Variation 5 - $[1;\ 1;\ 2]$ **[-]**	$1.696_{\pm 1.218}$	$0.964_{\pm 0.004}$	$0.041_{\pm 0.004}$	$0.031_{\pm 0.006}$	$0.162_{\pm 0.025}$
Variation 6 - 0.25 **[-]**	$1.618_{\pm 1.162}$	$0.956_{\pm 0.004}$	$0.050_{\pm 0.004}$	$0.043_{\pm 0.006}$	$0.138_{\pm 0.026}$
Variation 6 - 0.75 **[-]**	$1.543_{\pm 1.075}$	$0.977_{\pm 0.002}$	$0.026_{\pm 0.002}$	$0.018_{\pm 0.003}$	$0.177_{\pm 0.023}$
Variation 6 - 1.0 **[-]**	$1.758_{\pm 1.344}$	$0.981_{\pm 0.002}$	$0.022_{\pm 0.002}$	$0.015_{\pm 0.002}$	$0.198_{\pm 0.031}$

Table B.4: Results of the evaluation metrics for the scenario with objects with changing velocity. Average values and standard deviations are given for all metrics and improvements compared to the baseline are highlighted per variation.

Variation	GOSPA	MS	ME	FPR	FNR
Baseline	$4.477_{\pm 0.601}$	$0.956_{\pm 0.014}$	$0.050_{\pm 0.015}$	$0.030_{\pm 0.015}$	$0.433_{\pm 0.048}$
Variation 1 - 1 **[Hz]**	$4.927_{\pm 1.017}$	$0.940_{\pm 0.020}$	$0.066_{\pm 0.021}$	$0.043_{\pm 0.020}$	$0.617_{\pm 0.178}$
Variation 1 - 2 **[Hz]**	$4.507_{\pm 0.619}$	$0.953_{\pm 0.015}$	$0.053_{\pm 0.016}$	$0.032_{\pm 0.016}$	$0.446_{\pm 0.068}$
Variation 1 - 10 **[Hz]**	$4.402_{\pm 0.551}$	$0.940_{\pm 0.017}$	$0.068_{\pm 0.018}$	$0.049_{\pm 0.018}$	$0.394_{\pm 0.018}$
Variation 2 - 5 **[Hz]**	$4.508_{\pm 0.514}$	$0.946_{\pm 0.014}$	$0.061_{\pm 0.016}$	$0.038_{\pm 0.015}$	$0.534_{\pm 0.064}$
Variation 2 - 20 **[Hz]**	$4.497_{\pm 0.615}$	$0.963_{\pm 0.005}$	$0.042_{\pm 0.006}$	$0.022_{\pm 0.006}$	$0.476_{\pm 0.037}$
Variation 2 - 40 **[Hz]**	$4.446_{\pm 0.541}$	$0.947_{\pm 0.013}$	$0.058_{\pm 0.014}$	$0.036_{\pm 0.014}$	$0.658_{\pm 0.094}$
Variation 3 - $10 \cdot 10^{-2}$ **[m]**	$4.629_{\pm 0.730}$	$0.955_{\pm 0.004}$	$0.051_{\pm 0.004}$	$0.032_{\pm 0.005}$	$0.473_{\pm 0.009}$
Variation 3 - $40 \cdot 10^{-2}$ **[m]**	$4.464_{\pm 0.487}$	$0.957_{\pm 0.010}$	$0.049_{\pm 0.011}$	$0.019_{\pm 0.013}$	$0.495_{\pm 0.071}$
Variation 3 - $100 \cdot 10^{-2}$ **[m]**	$4.354_{\pm 0.319}$	$0.811_{\pm 0.071}$	$0.208_{\pm 0.076}$	$0.162_{\pm 0.084}$	$0.478_{\pm 0.105}$
Variation 4 - $1 \cdot 10^{3}$ **[-]**	$4.576_{\pm 0.711}$	$0.952_{\pm 0.007}$	$0.055_{\pm 0.007}$	$0.031_{\pm 0.009}$	$0.477_{\pm 0.034}$
Variation 4 - $4 \cdot 10^{3}$ **[-]**	$4.442_{\pm 0.528}$	$0.960_{\pm 0.012}$	$0.046_{\pm 0.013}$	$0.028_{\pm 0.013}$	$0.431_{\pm 0.061}$
Variation 4 - $8 \cdot 10^{3}$ **[-]**	$4.440_{\pm 0.544}$	$0.961_{\pm 0.010}$	$0.044_{\pm 0.011}$	$0.027_{\pm 0.011}$	$0.420_{\pm 0.049}$
Variation 5 - $[2;\ 1;\ 1]$ **[-]**	$4.467_{\pm 0.588}$	$0.951_{\pm 0.015}$	$0.056_{\pm 0.017}$	$0.036_{\pm 0.017}$	$0.423_{\pm 0.041}$
Variation 5 - $[1;\ 2;\ 1]$ **[-]**	$4.481_{\pm 0.562}$	$0.952_{\pm 0.016}$	$0.055_{\pm 0.018}$	$0.036_{\pm 0.017}$	$0.435_{\pm 0.043}$
Variation 5 - $[1;\ 1;\ 2]$ **[-]**	$4.547_{\pm 0.603}$	$0.953_{\pm 0.013}$	$0.054_{\pm 0.014}$	$0.034_{\pm 0.017}$	$0.431_{\pm 0.056}$
Variation 6 - 0.25 **[-]**	$4.532_{\pm 0.532}$	$0.943_{\pm 0.016}$	$0.065_{\pm 0.017}$	$0.048_{\pm 0.019}$	$0.438_{\pm 0.049}$
Variation 6 - 0.75 **[-]**	$4.468_{\pm 0.594}$	$0.971_{\pm 0.002}$	$0.033_{\pm 0.002}$	$0.016_{\pm 0.003}$	$0.498_{\pm 0.043}$
Variation 6 - 1.0 **[-]**	$4.535_{\pm 0.604}$	$0.975_{\pm 0.001}$	$0.028_{\pm 0.002}$	$0.012_{\pm 0.002}$	$0.563_{\pm 0.041}$

Table B.5: Results of the evaluation metrics for the scenario with occlusions. Average values and standard deviations are given for all metrics and improvements compared to the baseline are highlighted per variation.

References

[1] S. Singh, "Critical reasons for crashes investigated in the national motor vehicle crash causation survey," Tech. Rep. DOT HS 812 115, U.S. Department of Transportation, Washington, DC, USA, February 2015.

[2] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE Access*, vol. 8, pp. 58443–58469, 2020.

[3] I-CAVE, "I-Cave – Integrated cooperative automated vehicles." https://i-cave.nl/, 2018.

[4] J. Kelly and G. S. Sukhatme, "A general framework for temporal calibration of multiple proprioceptive and exteroceptive sensors," in *Experimental Robotics: The 12th International Symposium on Experimental Robotics* (O. Khatib, V. Kumar, and G. Sukhatme, eds.), (Berlin, Heidelberg), pp. 195–209, Springer, 2014.

[5] P. Zegelaar, T. van der Sande, and M. Alirezaei, "Vehicle control (4at050): Lecture 5 - adas sensors and features." Lecture Slides, 2020.

[6] J. Dickmann, J. Klappstein, M. Hahn, N. Appenrodt, H. Bloecher, K. Werber, and A. Sailer, "Automotive radar the key technology for autonomous driving: From detection and ranging to environmental understanding," in *IEEE Radar Conference, (RadarConf)*, pp. 1–6, 2016.

[7] K. Berns and E. Von Puttkamer, *Autonomous Land Vehicles*. Wiesbaden, Germany: Vieweg+Teubner Verlag, 2009.

[8] D. Joubert, "Adaptive occupancy grid mapping with measurement and pose uncertainty," Master's thesis, Stellenbosch University, 2012.

[9] A. Elfes, "Using Occupancy Grids for Mobile Robot Perception and Navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[10] C. G. D. P. Fernandez, "Grid-Based Multi-Sensor Fusion for On-Road Obstacle Detection: Application to Autonomous Driving," Master's thesis, KTH Royal Institute of Technology, 2015.

[11] J. Moras, V. Cherfaoui, and P. Bonnifait, "Credibilist occupancy grids for vehicle perception in dynamic environments," *IEEE International Conference on Robotics and Automation*, pp. 84–89, 2011.

[12] C. Premebida, J. Sousa, L. Garrote, and U. Nunes, "Polar-Grid Representation and Kriging-Based 2.5D Interpolation for Urban Environment Modelling," *18th IEEE Conference on Intelligent Transportation Systems, (ITSC)*, pp. 1234–1239, 2015.

[13] A. Souza and L. M. Gonçalves, "Occupancy-elevation grid: An alternative approach for robotic mapping and navigation," *Robotica*, vol. 34, no. 11, pp. 2592–2609, 2016.

[14] A. Azim and O. Aycard, "Detection, classification and tracking of moving objects in a 3D environment," *IEEE Intelligent Vehicles Symposium, (IV)*, pp. 802–807, 2012.

[15] J. Degerman, T. Pernstal, and K. Alenljung, "3D occupancy grid mapping using statistical radar models," *IEEE Intelligent Vehicles Symposium, Proceedings, (IV)*, pp. 902–908, 2016.

[16] S. Thrun, B. Wolfram, and D. Fox, *Probabilistic robotics*. Cambridge, MA, USA: MIT Press, 2005.

[17] "Kalman filter." In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Kalman_filter&oldid=1041473714, 2021.

[18] J. Choi, S. Ulbrich, B. Lichte, and M. Maurer, "Multi-Target Tracking using a 3D-Lidar sensor for autonomous vehicles," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 881–886, IEEE, 2013.

[19] S. Steyer, G. Tanzmeister, and D. Wollherr, "Object tracking based on evidential dynamic occupancy grids in urban environments," *IEEE Intelligent Vehicles Symposium (IV)*, pp. 1064–1070, 2017.

[20] C. Coué, C. Pradalier, C. Laugier, T. Fraichard, and P. Bessière, "Bayesian occupancy filtering for multitarget tracking: An automotive application," *International Journal of Robotics Research*, vol. 25, no. 1, pp. 19–30, 2006.

[21] D. Nuss, T. Yuan, G. Krehl, M. Stuebler, S. Reuter, and K. Dietmayer, "Fusion of laser and radar sensor data with a sequential Monte Carlo Bayesian occupancy filter," *IEEE Intelligent Vehicles Symposium (IV)*, pp. 1074–1081, 2015.

[22] A. Vatavu, R. Danescu, and S. Nedevschi, "Stereovision-based multiple object tracking in traffic scenarios using free-form obstacle delimiters and particle filters," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 498–511, 2015.

[23] D. E. Clark, J. Bell, Y. de Saint-Pern, and Y. Petillot, "PHD filter multi-target tracking in 3D sonar," in *Oceans 2005 - Europe*, vol. 1, pp. 265–270, IEEE, 2005.

[24] D. E. Clark, K. Panta, and B.-N. Vo, "The GM-PHD filter multiple target tracker," in *9th International Conference on Information Fusion*, pp. 1–8, IEEE, 2006.

[25] 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," *International Journal of Robotics Research*, vol. 37, no. 8, pp. 841–866, 2018.

[26] T. D. Vu, J. Burlet, and O. Aycard, "Grid-based localization and local mapping with moving object detection and tracking," *Information Fusion*, vol. 12, no. 1, pp. 58–69, 2011.

[27] M. Lindström and J. O. Eklundh, "Detecting and tracking moving objects from a mobile platform using a laser range scanner," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, pp. 1364–1369, 2001.

[28] H. Koyasu, J. Miura, and Y. Shirai, "Realtime omnidirectional stereo for obstacle detection and tracking in dynamic environments," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 31–36, 2001.

[29] E. Prassler, J. Scholz, and A. Elfes, "Tracking multiple moving objects for real-time robot navigation," *Autonomous Robots*, vol. 8, no. 2, pp. 105–116, 2000.

[30] M. Schreier, *Bayesian environment representation, prediction, and criticality assessment for driver assistance systems*. PhD thesis, Technischen Universität Darmstadt, 2017.

[31] B. Schiele and J. L. Crowley, "A comparison of position estimation techniques using occupancy grids," *Robotics and Autonomous Systems*, vol. 12, no. 3-4, pp. 163–171, 1994.

[32] F. Lu and E. Milios, "Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 18, no. 3, pp. 249–275, 1997.

[33] M. Adams, W. S. Wijesoma, and A. Shacklock, "Autonomous Navigation: Achievements in Complex Environments," *IEEE Instrumentation & Measurement Magazine*, vol. 10, no. 3, pp. 15–21, 2007.

[34] M. K. Habib, "Real time mapping and dynamic navigation for mobile robots," *International Journal of Advanced Robotic Systems*, vol. 4, no. 3, pp. 323–338, 2007.

[35] H. Badino, U. Franke, and R. Mester, "Free space computation using stochastic occupancy grids and dynamic programming," in *Workshop on Dynamical Vision, ICCV, Rio de Janeiro, Brazil*, vol. 20, Citeseer, 2007.

[36] M. Li, Z. Feng, M. Stolz, M. Kunert, R. Henze, and F. Küçükay, "High Resolution Radar-based Occupancy Grid Mapping and Free Space Detection," in *VEHITS*, pp. 70–81, 2018.

[37] S. Thrun, "Learning occupancy grids with forward models," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, pp. 1676–1681, 2001.

[38] J. Miura, Y. Negishi, and Y. Shirai, "Mobile robot map generation by integrating omnidirectional stereo and laser range finder," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 250–255, 2002.

[39] R. Garcia, O. Aycard, T.-D. Vu, and M. Ahrholdt, "High level sensor data fusion for automotive applications using occupancy grids," in *10th International Conference on Control, Automation, Robotics and Vision, (ICARCV)*, pp. 530–535, IEEE, 2008.

[40] M. E. Bouzouraa and U. Hofmann, "Fusion of occupancy grid mapping and model based object tracking for driver assistance systems using laser and radar sensors," *IEEE Intelligent Vehicles Symposium, (IV)*, pp. 294–300, 2010.

[41] H. Li and F. Nashashibi, "A new method for occupancy grid maps merging: Application to multi-vehicle cooperative local mapping and moving object detection in outdoor environment," *12th International Conference on Control, Automation, Robotics and Vision, (ICARCV)*, no. December, pp. 632–637, 2012.

[42] F. Homm, N. Kaempchen, J. Ota, and D. Burschka, "Efficient occupancy grid computation on the GPU with lidar and radar for road boundary detection," *IEEE Intelligent Vehicles Symposium, (IV)*, pp. 1006–1013, 2010.

[43] M. A. Richards, *Fundamentals of Radar Signal Processing*. New York, NY, USA: McGraw-Hill Education, 2005.

[44] N. J. Gordon, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," *IEE proceedings F (radar and signal processing)*, vol. 140, no. 2, pp. 107–113, 1993.

[45] R. Danescu, F. Oniga, and S. Nedevschi, "Modeling and tracking the driving environment with a particle-based occupancy grid," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1331–1342, 2011.

[46] K. Li, G. Lin, L. Lee, and J. Juang, "Application of particle filter tracking algorithm in autonomous vehicle navigation," in *2013 CACS International Automatic Control Conference (CACS)*, pp. 250–255, IEEE, 2013.

[47] V. R. Madhavan, "Ultrasonic Sensor-Based Mapping for an Autonomous Vehicle," Master's thesis, Eindhoven University of Technology, 2021.

[48] R. Grewe, M. Komar, A. Hohm, S. Lüke, and H. Winner, "Evaluation method and results for the accuracy of an automotive occupancy grid," *2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012)*, pp. 19–24, 2012.

[49] A. S. Rahmathullah, Á. F. García-Fernández, and L. Svensson, "Generalized optimal sub-pattern assignment metric," *2017 20th International Conference on Information Fusion (Fusion)*, pp. 1–8, 2017.

[50] C. H. Goulden, *Methods of statistical analysis*. New York, NY, USA: Wiley, 1956.